# Documentation for the "extract" Function, Updated July, 2018

## 1 General

"extract" is an R function which implements the dyad ratios algorithm described in Stimson (2018) for constructing a latent time series from the raw materials of survey research marginals. It inputs data in archival form, as dated survey results, aggregates them into regular periods selected at run time, and extracts the latent dimension(s) which best account for the raw series.

Although its central mathematics is entirely different—based upon ratios of comparable observed series rather than covariance of "variables"—its behavior is strongly analogous to extraction of a "factor score" in standard (iterative) principal components analysis.

## 2 Usage

The routine is called from R by the command:

output<- extract(varname,date,index,ncases,
unit="A",mult=1,begindt=NA,enddt=NA,npass=1,smoothing=TRUE,endmonth=12)

The first three parameters are necessary variable names and others are optional.

**ncases** is the name of a variable which records the number of cases for a particular survey result. If absent, cases are equally weighted for aggregation into periods.

**unit** is the aggregation period, "D" (daily), "M" (monthly), "Q" (quarterly), "A" (annual, default), "O" (other, for multi-year aggregation).

**mult** is the number of years, used only for unit option "O."

**begindt** is the beginning date for analysis. Default (NA) determines beginning date from the earliest date in the data.

Entries for beginning and ending date use the ISOdate function. For example, to start an analysis in January, 1993, enter ISOdate(1993,1,1). (As always, R is case sensitive. So "ISO" must be caps and "date" lower case.)

**enddt** is the ending date for analysis. Default (NA) determines ending date from the latest date in the data.

**Warning**: The routine can not determine the earliest or latest dates of items which actually are used in the analysis. The criterion for usage is that items must appear in more than one period *after* aggregation. So if the beginning or ending dates are determined by an item which is discarded because it does not meet this criterion, the routine will fail.

**smoothing** specifies whether or not exponential smoothing is applied to intermediate estimates during the iterative solution process. Default is TRUE.

**npass** not yet implemented

## 2.1 Data Requirements

Input data must contain four logically necessary elements (variables). These are:

**Varname** Varname is a character name given to input series. The routine assumes that any two observations with the same name are comparable and that any change in name signals noncomparability. Ratios are computed only between comparable observations.

**Date** Date is typically one of the dates of survey field work(e.g., first, last, or median day). It is recorded as an ISO numeric representation of all possible calendar dates. (See below for further documentation.)

**Index** Index is the numeric summary value of the result. It might be a percent or proportion responding in a single category, (e.g., the "approve" response in presidential approval) or some multi-response summary, for example,

$$Index = \frac{(Percent\ Agree)}{(Percent\ Agree + Percent\ Disagree)}$$

Interpretation of the derived latent dimension is eased by having the index coded such that polarity is the same across items—for example, if the concept being measured is liberalism, then high values always stand for the more liberal response—but as in principal component analysis, the routine deals appropriately with polarity switches.

**Ncases** The number of cases indicator, typically sample size, is used during aggregation to produce a weighted average when multiple readings fall in one aggregation period. If this issue doesn't occur or if the user prefers an unweighted average, then ncases=NULL—or omitting the ncases variable—will ignore case weighting. In the case of a mix of available and missing ncases indicators, 0 or NA values are reset to 1000.

## 2.2 Dating Considerations

The routine expects a date variable of R class "date." This is how date will appear if it has been transferred from one of the database, spreadsheet, or statistical package formats recognized by the "foreign" library.

In the case of data entered from text sources, all is not so easy. The key is that a numeric representation of dates is needed, such as separate variables for month, day, and year. From these or fewer, depending on the unit specification, a date variable can be created as follows.

Assume numeric values for 3 variables, named month, day, and year. Then

date<-ISOdate(year,month,day)

will create the desired result. A further step is required to coerce the class of the date variable to "date."
date<-as.Date(date)

Or, for "A" or "O" aggregation:

date<-ISOdate(year,1,1)
date<-as.Date(date)

shows that constants can be used for unneeded date components.)

Warning: ISOdate will not handle fake dates (for example, 1-32-05). It decodes dates that actually existed on past calendars or will exist on future ones (e.g., no Feb 29 unless year is actually a leap year.)

What you don't want is a string representation, e.g., "January 1, 1993" because pulling out the pieces can be hideous. See R documentation for as.Date for conversion methods.

Integer representation of dates from Excel or similar sources translate successfully with the foreign package, but would not be correct if input as integers in text form. Microsoft does not comply with the ISO standard. A Microsoft integer date, multiplied by 86400 (which is 24*60*60 for the number of seconds in a day) will produce an ISO date.

# 3 Output

extract produces as output 7 categories of information:

**formula** reproduces the user call

**setup** supplies basic information about options and the iterative solution

**period** is a list of the aggregation periods, for example, 2005.02 for February, 2005

**varname** is a list in order of the variables actually used in the analysis, a subset of all those in the data.

**loadings** are the item-scale correlations from the final solution. Their square is the validity estimate used in weighting.

**means and std.deviations** document the item descriptive information, and

**latent** is the estimated time series, the purpose of everything.

## 3.1 Output Functions

The raw output file created at run time contains everything of interest, but in an inconvenient format. Three output functions display the same information in a more logically coherent manner. Each is based on the output file and each has a class designation of "extract" to discriminate it from other R commands.

**plot** plot(outputobject) displays a time series plot of the number of dimensions estimated on y axes against time units on the x axis.

**display** display(outputobject) displays the latent series estimates by time unit, the format depending upon aggregation interval chosen. For monthly, it is "yyyy mm latent-estimate" for example.

display(outputobject,"myfilename") will write the latent estimates to a disk file on the current working directory.

**summary** summary(outputobject) displays information about the raw series of the analysis. Under the heading: "Variable Name Cases Loading Mean Std Dev" it lists as many series as are used in the solution, giving variable name, number of cases (after aggregation), dimension loadings, and means and standard deviations of the raw series.

summary(outputobject,"myfilename") will write the variable information to a disk file.

## 3.2 Negative Correlations?

Correlations, in the case of time series, measure whether two series vary in or out of phase. Thus the cross-sectional interpretation of the negative correlation—that two items are inversely related—does not hold. It is not unusual to observe negative "loadings" in extract analyses. They mean only that items move out of phase, not that they are opposites.

## 3.3 Model

Assume N survey results, each coded into a meaningful single indicator. These results consist of n subsets of comparable items measured at different times, 1–T. Designate each result $x_{it}$, where the i subscript indicates item and t indicates aggregation period, 1–T.

The starting assumption is that ratios, $r_{it+k} = x_{it+k}/x_{it}$ of the comparable item i at different times will be meaningful indicators of the latent concept to be extracted. Metric information is thus lost, which is desirable because absent a science of question wording, we have no ability to compare different items. If there were no missing observations, then for each item i, we could let $r_{i1} = 1.0$ and observe the complete set of ratios, $r_{i2}, r_{i3}, \ldots r_{iT}$. Then an average across the n items forms an excellent estimate of the latent concept $\theta_t$:

$$\hat{\theta}_t = \frac{\sum_{i=1}^{n} r_{it}}{n} \tag{1}$$

But we do have missing x's—and in the typical case it is most of them. We would still be in good shape if we had a common period, say period 1, which was available for all series. We could then form the sum from the k available items, $k \leq n$, and divide by k. But we also lack such a common period. That motivates a recursive approach.

**Forward Recursion:** Begin by selecting that subset of items which are available at time 1. For them we can form $\hat{\theta}_t$ for t=1,T setting $\hat{\theta}_1 = 1.0$ and calculating $\hat{\theta}_2 \ldots \hat{\theta}_T$ from whatever subsets of items are available. Now proceed to period 2, setting $\hat{\theta}_2$ to that value estimated from period 1 and now, using the subset of items which include period 2, estimating $\hat{\theta}_3 \ldots \hat{\theta}_T$ from the assumption that $\theta_2 = \hat{\theta}_2$. By projecting $\hat{\theta}_2$ forward in this manner, the estimates for periods 3 through T become comparable to what they would have been had period 1 information been available. This procedure is repeated one period at a time through T-1, giving from 1 to T-1 different estimates of each of the $\theta_t$. An average of all of them becomes $\hat{\theta}_t$.

**Backward Recursion:** It will be seen that forward recursion very heavily weights early information relative to later information. Period 1 contributes to all subsequent estimates,

whereas period T-1 contributes only to T, and period T only to itself. Thus the direction of recursion matters. Employing the same procedure backward puts a different weight on the items and gives a comparable, but not identical, set of estimates. Thus a more efficient set of estimates, one weighting all information equally, can be gained by averaging the two recursive estimates. (And the correlation between forward and backward series becomes a reliability estimate.)

### 3.3.1 Iterative Validity Estimation

As in iterated principal components we both make assumptions about item validity and then, post hoc, have the ability to observe empirical estimates of validities (the square of item/scale correlations). At the beginning validities are assumed to be 1.0 for all items. Then the empirically estimated validities become assumed validities for the next iteration. This procedure is repeated until the difference between assumed and estimated validities is effectively zero for all items, the maximum item discrepancy less than .001.

# References

Stimson, James A. 2018. "The Dyad Ratios Algorithm for Estimating Latent Public Opinion: Estimation, Testing, and Comparison to Other Approaches." *Bulletin of Methodological Sociology* 137-138:201–218.