# Simple Indexes

*Joseph Nathan Cohen*

*Fall 2019*

## Contents

**Introduction** **1**

**The Model** **1**

**Implementation** **2**

Data . . . . . 2

Step One: Standardize Variables . . . . . 2

Step Two: Average the Standardized Variables . . . . . 3

**Pitfalls** **3**

## Introduction

We are studying different ways to consolidate variables in a data set. The simplest method for consolidating variables is to average standardized variable scores. However, this technique for variable consolidate employs a range of assumptions that may not hold in reality.

This method assumes:

- All variables are continuous
- All variables are capturing the same latent construct
- All variables reflect the underlying construct equally

## The Model

$$s_{ij} = \frac{\sum_{=1}^{N} (\frac{x_{ij} - \bar{x_i}}{\sigma_i})}{N}$$

$$I = \frac{\sum_{i=1}^{M} s_{ij}}{M}$$

Where:

- $I$ is the index score
- $x_ij$ is variable $i$ for subject $j$
- $s_ij$ is a standardized transformation of $x_ij$ with a mean of zero and a standard deviation of one.
- $\bar{x_i}$ is the mean of variable $x_i$
- $\sigma_i$ is the standard deviation of variable $x_i$
- $N$ is the total number of subjects with reported values of $x_i$
- $M$ is the total number of variables being used in the index

# Implementation

## Data

We are going to use the OECD *Better Life Index*[1], a database that seeks to measure 41 countries' living standards. You can download a copy of the data here

```
library(readxl)
DATA <- read_xlsx("OECD Better Life Data.xlsx", sheet = 1)
DATA <- data.frame(DATA)

#The set has 24 variables, which are explained in the dataset Excel workbook
#Here are the variable names:
names(DATA)
```

```
##  [1] "country"      "nofacilities" "houseexp"     "rooms"
##  [5] "dispinc"      "finwealth"    "labinsec"     "employment"
##  [9] "ltunemp"      "earnings"     "support"      "edattain"
## [13] "skills"       "schoolyears"  "airpollution" "waterqual"
## [17] "stakeeng"     "voters"       "lifeexp"      "selfhealth"
## [21] "satisfaction" "feelsafe"     "murder"       "longhours"
## [25] "leisuretime"
```

```
#A quick look at the top of the first few variables:
head(DATA[1:7], n=5)
```

```
##     country nofacilities houseexp rooms dispinc finwealth labinsec
## 1 Australia           NA       20    NA   32759    427064      5.4
## 2   Austria          0.9       21   1.6   33541    308325      3.5
## 3   Belgium          1.9       21   2.2   30364    386006      3.7
## 4    Canada          0.2       22   2.6   30854    423849      6.0
## 5     Chile          9.4       18   1.2      NA    100967      8.7
```

## Step One: Standardize Variables

You begin by standardizing the variables that will be used in the index. Doing so puts all of the variables on the same scale, so that the index isn't unduly influenced by variables that are denominated in large numbers.

Below, we standardize the variables of our data set using the **standardize()** function in the ***psycho*** package.[2]:

```
library(psycho)
SDATA <- standardize(DATA)

#Rounding the data's numeric variables to two decimal places, just
#to make it easier to read:
SDATA[2:25] <- round(SDATA[2:25], 2)

#A look at the resulting set:
head(SDATA[1:7], n = 5)
```

```
##     country nofacilities houseexp rooms dispinc finwealth labinsec
## 1 Australia           NA    -0.26    NA    0.70      0.83    -0.24
## 2   Austria        -0.49     0.14 -0.04    0.81      0.11    -0.56
## 3   Belgium        -0.38     0.14  1.31    0.36      0.58    -0.53
```

[1] OECD (2019) "OECD Better Life Index" Online database, available at http://www.oecdbetterlifeindex.org/

[2] We use this command in lieu of the base packages **scale()** command because **standardize()** ignores character variables instead of returning an error. This feature ends up being convenient when you are working with a set that has numeric and text variables interspersed throughout the data frame

```
## 4    Canada        -0.58     0.53  2.21    0.43      0.81    -0.14
## 5     Chile         0.51    -1.05 -0.94      NA     -1.14     0.32
```

### Step Two: Average the Standardized Variables

The next step is to average the standardized variables. You may use mean or median. The former is perhaps more popular and arugably better incorporates the effect of each variable that comprises the index, but is also more likely to be influenced by outliers.

To average these variables, we use the **rowMeans()** command in the base package. We are asking the command to deliver the mean of scores in rows 2 to 25 in the SDATA data frame. The command's default is to render an "NA" if an observation has any missing values (i.e., it employs listwise deletion). We set the subcommand "na.rm = T" in order to get the average of all non-missing observations.

```
SDATA$wellbeing.1 <- rowMeans(SDATA[2:25], na.rm = T)

#Top 5 wellbeing using this consolidation strategy:
library(dplyr)
arrange(SDATA,-wellbeing.1)[1:5,c(1,26)]
```

```
##         country wellbeing.1
## 1       Iceland   0.4809524
## 2   New Zealand   0.4740909
## 3     Australia   0.4618182
## 4   Switzerland   0.4542857
## 5 United States   0.4133333
```

```
#Bottom 5:
arrange(SDATA,wellbeing.1)[1:5,c(1,26)]
```

```
##    country wellbeing.1
## 1    Chile  -0.6300000
## 2  Hungary  -0.5472727
## 3   Russia  -0.5294444
## 4 Colombia  -0.4783333
## 5   Brazil  -0.4740000
```

## Pitfalls

Before enumerating the pitfalls associated with the simple index method of consolidating variables, readers might have noted that missing data is a problem in this analysis. Of the 40 countries in this data set, only 15 have data on all 24 variables. Another 20 are missing 1 - 3 values, and five more are missing five or more values. The result is that c ountreis' "wellbeing" scores are generated by an inconsistent mix of variables. This is not a weakness of simple indexes, but a generic missing data problem. Were someone to approach me with this data and concerns about missing data, I would recommend using multiple imputaton with randomness, which we bracket here but discuss in another module.

Now, let's turn to problems that are more specific to simple indexation as a data consolidation method. I can think of a few pitfalls associated with this indexation method offhand, which are related to this indexation method's underlying assumptions. The fragility of these assumptions are easier to see when we consider the indexation operation above.

In my view, the biggest pitfall with this method is that it can tempt anaylsts to build indexes mechanistically without thinking about whether their data consolidation strategy makes sense. Consider the "wellbeing" index that we build using OECD data in the above example. We built a "wellbeing" index using the mean of 24 standardized variables. In so doing, we assumed that:

- All 24 variables were continuous (which they were!)
- All 24 variables were measuring the same thing
- Each of our 24 variables captured precisely 1/24th of countries' "wellbeing" levels

In other words, the method assumes that people's subjective feelings of safety are just as important as their children's performance on standardized aptitude tests, each of which are precisely as important as water quality. Moreover, the data has three variables related to housing, but only two related to physical safety. If we simply calculate the mean across all our variables, we are implicitly assuming that housing is 50% more important than physical safety in shaping a country's overall wellbeing.

There are easy statistical fixes for dealing with this kind of situation. For example, the first situation – concerns that these individual wellbeing metrics aren't equally important can be address by weighting. For example, I could manually specify weights associated with each of this set's 24 variables. In the example below, I double the importance of the public safety-related metrics, and multiply the health metrics' importance by 10. We then calculate an alternative wellbeing index using the **rowWeightedMeans()** command from the ***matrixStats*** package:

```
#We are going to use variables 2 to 25 - 24 total variables
#We specify weights for each of these 24, as they appear in the set.
#See the results of the names(DATA) operation above.

weights <- c(1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,10,10,1,3,3,1,1)

#To double check the weights you are assigning to each variable:
#cbind(names(SDATA[2:25]), weights)

#Recalculate the Wellbeing index:
library(matrixStats)
```

```
## Warning: package 'matrixStats' was built under R version 3.5.3
```

```
##
## Attaching package: 'matrixStats'
```

```
## The following object is masked from 'package:dplyr':
##
##     count
```

```
SDATA$wellbeing.2 <- rowWeightedMeans(as.matrix(SDATA[2:25]), weights, na.rm = T)

#Does it affect our rankings? Top 5: (I swear this is a coincidence)
arrange(SDATA,-wellbeing.2)[1:5,c(1,26, 27)]
```

```
##       country wellbeing.1 wellbeing.2
## 1      Canada   0.4037500   0.6263043
## 2 New Zealand   0.4740909   0.6081818
## 3 Switzerland   0.4542857   0.6051163
## 4   Australia   0.4618182   0.5852273
## 5     Iceland   0.4809524   0.5279070
```

```
#Bottom 5:
arrange(SDATA,wellbeing.2)[1:5,c(1,26,27)]
```

```
##        country wellbeing.1 wellbeing.2
## 1 South Africa  -0.2620000  -1.7053571
## 2       Russia  -0.5294444  -0.9940000
## 3    Lithuania  -0.3590476  -0.7793023
## 4       Latvia  -0.4166667  -0.7015217
```

```
## 5      Hungary  -0.5472727  -0.5750000
```

Insofar as concerns that different domains of wellbeing (e.g., health, safety, economic life) have different numbers of variables in this set (see above and the set's codebook). There are a lot of economic variables, which is part of the reason that the US performs so wellin the first measure but less well in the second (US living standards are strongly buoyed by their high levels of material wealth). Many well-known indexes of this type deal with this concern by consolidating variables into first-order indexes that group empirical variables according to some common theme, and then calculate a higher-order, overall index by averaging these first order indexes.

For example, we might create a Health index using this OECD data as follows. And then we might repeat the same procedure with the other domains of wellbeing described in the OECD set.

```
SDATA$health <- rowMeans(SDATA[c(20,21)])
```

But ultimately we are always left with the questions surrounding our assumption that these variables are actually related to the same underlying construct "wellbeing". We are just assuming that there is a thing called "wellbeing", and that it is somehow made up of things like work-life balance, safety, housing, and the other major categories described in the set. How do we know this is true?

A more empirically-minded analyst might want to check the data for evidence that these variables are in fact measure what we think they are measuring. These are the subjects of this module's other three lessons.