Initial Indicator Analysis

William Becker

Table of contents

Intro

This is a short document to record the initial indicator analysis of the Municipal Vulnerability Index (MVI) for Guatemala.

The objectives here are:

1. Univariate analysis based on data availability, uniqueness, distribution properties.
2. Multivariate analysis based on correlations.

We recall that the end objective here is to eventually arrive at a "module" which can be used for selecting indicators based on statistical characteristics. Therefore we need some basic criteria to do that by.

Typical criteria for indicator selection and treatment can be as follows:

- Percentage missing data (indicators below a set percentage are flagged - this is often set at e.g. 66%)

- Percentage zeroes (indicators above a set percentage are flagged - this could be e.g. 50%)
- Percentage unique values (again, we can set a threshold here)
- Presence of outliers, flagged by having both |skew| > 2 and kurtosis > 3.5
- Collinearity with other indicators in the same aggregation group
- Strong negative correlation with other indicators in the same group

The idea here would be to simply flag these things to the user, who then has the option to either remove or retain the indicators.

Data import

In a previous version, I began cleaning and formatting the data directly in R. However, since the end objective is to arrive at data modules that can deal with data from other countries, we need to consider a generic input format and it makes sense to do this early on.

So for now I have an intermediate solution, which is to format the data in Excel for COINr. I have also sketched out a possible generic format based on one sheet. However this latter will need the structure of the index hard-coded in the end product, so I'm going to begin by importing the familiar `iData` and `iMeta` data frames.

What I did in Excel:

- Extracted the indicator data, keeping only column headers (codes) and renaming the municipal code to `uCode` and the Municipio to `uName`.
- Created short codes for aggregate levels
- Created an `imeta` table based on the indicator metadata table from the master sheet. This renames columns, converts some things to COINr-compatible (e.g. directions), and also adds the rows for the aggregate levels which effectively defines the structure of the index.
- I noticed one probable typo/mistake and corrected. In the indicator data the code for "Tasa de Pobreza" is "S.E.3" whereas in the metadata it is specified as "S.E.4". I altered the latter to "S.E.3" as I presume this is what it should be.

A suggestion here would be that for any new indicators that become available, we could directly add them to the `iData` table. We could potentially have an Excel sheet with reads the master indicator table and formats it for import to R.

We begin by importing the data set as it stands (the version at the time of writing is 30/01/2023).

```
library(readxl)

# note: if indicators are added the ranges here will need to be adjusted
idata <- read_excel("data_input/data_input_formatted.xlsx", sheet = "iData")
```

```
imeta <- read_excel("data_input/data_input_formatted.xlsx", sheet = "iMeta")

idata
```

```
# A tibble: 340 x 73
   Cód_dep Departa~1 uCode uName A.A.1 A.D.1 A.M.1 A.M.2 A.M.3 A.M.4 A.M.5 A.M.6
     <dbl> <chr>     <dbl> <chr> <lgl> <dbl> <dbl> <dbl> <dbl> <dbl> <lgl> <lgl>
 1       1 Guatemala   101 Guat~ NA      9.1   614 148.    143  852. NA    NA
 2       1 Guatemala   102 Sant~ NA      7.8    57  18.4    NA  741. NA    NA
 3       1 Guatemala   103 San ~ NA      4.2    40  69.7    NA  696. NA    NA
 4       1 Guatemala   104 San ~ NA      2.5     0 166.     NA 2626. NA    NA
 5       1 Guatemala   105 Pale~ NA      7      10 164.     NA 1363. NA    NA
 6       1 Guatemala   106 Chin~ NA      7.2    13  34.9    NA  831. NA    NA
 7       1 Guatemala   107 San ~ NA      7.4     5  49.3    NA  913. NA    NA
 8       1 Guatemala   108 Mixco NA      9.2   114  37.5    NA  861. NA    NA
 9       1 Guatemala   109 San ~ NA      4.2     2  34.1    NA  579. NA    NA
10       1 Guatemala   110 San ~ NA      2.6    19  59.7    NA  707. NA    NA
# ... with 330 more rows, 61 more variables: A.V.1 <dbl>, A.V.2 <dbl>,
#   A.V.3 <dbl>, A.V.5 <lgl>, S.A.1 <dbl>, S.A.2 <dbl>, S.A.3 <dbl>,
#   S.A.4 <dbl>, S.E.1 <dbl>, S.E.2 <dbl>, S.E.3 <dbl>, S.G.1 <dbl>,
#   S.G.2 <dbl>, S.G.3 <dbl>, S.G.4 <dbl>, S.G.5 <dbl>, S.G.6 <dbl>,
#   S.G.7 <dbl>, S.G.8 <dbl>, S.T.1 <dbl>, S.T.2 <dbl>, S.T.3 <lgl>,
#   S.T.4 <dbl>, C.C.1 <dbl>, C.C.2 <lgl>, C.C.3 <lgl>, C.C.4 <lgl>,
#   C.E.1 <lgl>, C.E.2 <lgl>, C.E.3 <lgl>, C.E.4 <dbl>, C.E.5 <dbl>, ...
```

The data includes columns of indicator data, but also municipal code, name of municipality, and department code and name. I have already renamed the relevant columns to the required `uCode` and `uName`. The department name and code are treated as grouping variables.

We still have a couple of things to do: one is to make the `uCode` column into a character vector that cannot be coerced to numeric.

```
# we have to make iCode non-numeric
idata$uCode <- paste0("c", idata$uCode)
```

It is also evident that some indicators here are placeholders and have no available data yet. We have to remove these first.

```
# which indicators have no data?
i_nodata <- names(idata)[colSums(!is.na(idata)) == 0]
i_nodata
```

```
 [1] "A.A.1" "A.M.5" "A.M.6" "A.V.5" "S.T.3" "C.C.2" "C.C.3" "C.C.4" "C.E.1"
[10] "C.E.2" "C.E.3" "C.J.1" "C.J.3" "C.S.1" "C.S.2" "C.S.3"
```

```
idata <- idata[!(names(idata) %in% i_nodata)]
```

We will also do one more thing, which is to convert population columns to thousands - this avoids very small numbers generated by dividing indicators by large numbers.

```
pop_cols <- startsWith(names(idata), "Pop_")

# div by 1000
idata[, pop_cols] <- idata[, pop_cols]/1000
# edit units in imeta
imeta$Unit[startsWith(imeta$iCode, "Pop_")] <- "Thousands"
```

With this, the `idata` data frame is in the correct format. As for the metadata `imeta`, this has already been formatted in Excel. The only thing to note here is that I have removed some metadata columns which are not necessary for COINr. If needed, we could pass through other columns in the future.

The last thing to do is to check consistency between `imeta` and `idata`. Since we removed columns from `idata` we also have to remove these from `imeta`.

```
imeta <- imeta[!(imeta$iCode %in% i_nodata), ]
```

Check whether all `imeta` codes are in `idata`:

```
# exclude aggregate rows
imeta_i <- imeta[imeta$Type != "Aggregate", ]

# any codes missing?
missing_idata <- imeta_i$iCode[!(imeta_i$iCode %in% names(idata))]
missing_idata
```

```
character(0)
```

This is fine. Now do the reverse check:

```
icodes_idata <- names(idata)[!(names(idata) %in% c("uCode", "uName"))]

icodes_idata[!(icodes_idata %in% imeta$iCode)]
```

```
character(0)
```

OK so everything looks consistent. There is one last thing I found which causes a problem. One of the categories “Desplazamento” doesn’t have any indicators underneath it so we need to remove it.

```
imeta <- imeta[imeta$iCode != "Desplaz", ]
```

This means that now we can go ahead and enter the data into COINr, and assemble a “coin”. We will also clean up the environment.

```
library(COINr)

MVI <- new_coin(iData = idata, iMeta = imeta, level_names = c("Indicator", "Category", "Di
```

```
iData checked and OK.
```

```
iMeta checked and OK.
```

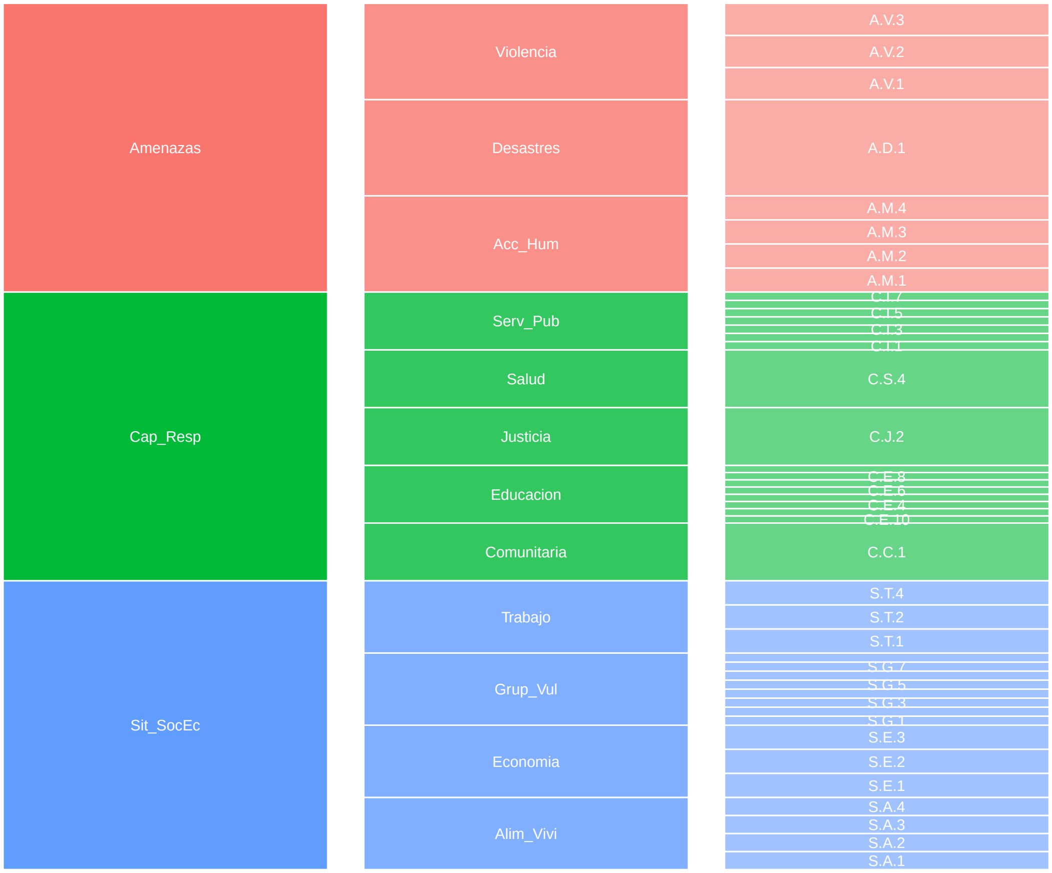
```
Written data set to .$Data$Raw
```

```
# clean all except coin
rm(list = setdiff(ls(), "MVI"))
```

Analysis

With the data in COINr we can begin the analysis. The first thing is to look at the structure of the index:

```
plot_framework(MVI, type = "stack", text_size = 4)
```



Let us also check the contents of the coin:

```
MVI
```

```
--------------
A coin with...
--------------
Input:
  Units: 340 (c1001, c1002, c1003, ...)
  Indicators: 44 (A.M.1, A.M.2, A.M.3, ...)
  Denominators: 9 (Pop_Total, Pop_Men, Pop_Women, ...)
  Groups: 2 (Departamento, Cód_dep)
```

```
Structure:
  Level 1 Indicator: 44 indicators (A.M.1, A.M.2, A.M.3, ...)
  Level 2 Category: 12 groups (Acc_Hum, Desastres, Violencia, ...)
  Level 3 Dimension: 3 groups (Amenazas, Cap_Resp, Sit_SocEc)
  Level 4 Index: 1 groups (MVI)

Data sets:
  Raw (340 units)
```

This looks fine. Now, the next thing is to begin with the indicator analysis. Here, the question is whether to perform the analysis on the data before or after indicators have been denominated (divided by scaling variables such as population). On reflection the sensible option seems to be to do analysis on the data *after* denomination because this is how we would expect the data to arrive in the generalised version. So, let's go ahead and denominate.

```r
MVI <- Denominate(MVI, dset = "Raw")
```

```
Written data set to .$Data$Denominated
```

We don't need to specify anything here in particular because the denominator variables (populations by group) are already input with `iData`, and the mapping of which denominators to use with which indicators is specified in `iMeta`. The output is a new data set with denominated data.

We can quickly generate a table of indicator statistics in COINr using the `get_stats()` function, and this will help us with several of the analysis tasks.

```r
df_stats <- get_stats(MVI, dset = "Denominated",
                      t_skew = 2, t_kurt = 3.5,
                      t_avail = 0.66,
                      t_zero = 0.66,
                      t_unq = 0.5,
                      out2 = "df")
```

Data availability

Let's begin with data availability. Let's assume a rule "data availability should be greater than or equal to 66%". Under this criterion let's see which indicators are flagged:

```r
df_stats[df_stats$Flag.Avail == "LOW", c("iCode", "N.Avail", "Frc.Avail")]
```

```
  iCode N.Avail Frc.Avail
4 A.M.3      40     0.118
```

This shows that only one indicator fails: “A.M.3” which is “# de NNA no acompañados deportados”. Here, the recommendation would be to exclude this indicator since its data availability is very low, at 12%. Only if the indicator is conceptually vital should we include it.

Zeros and duplicates

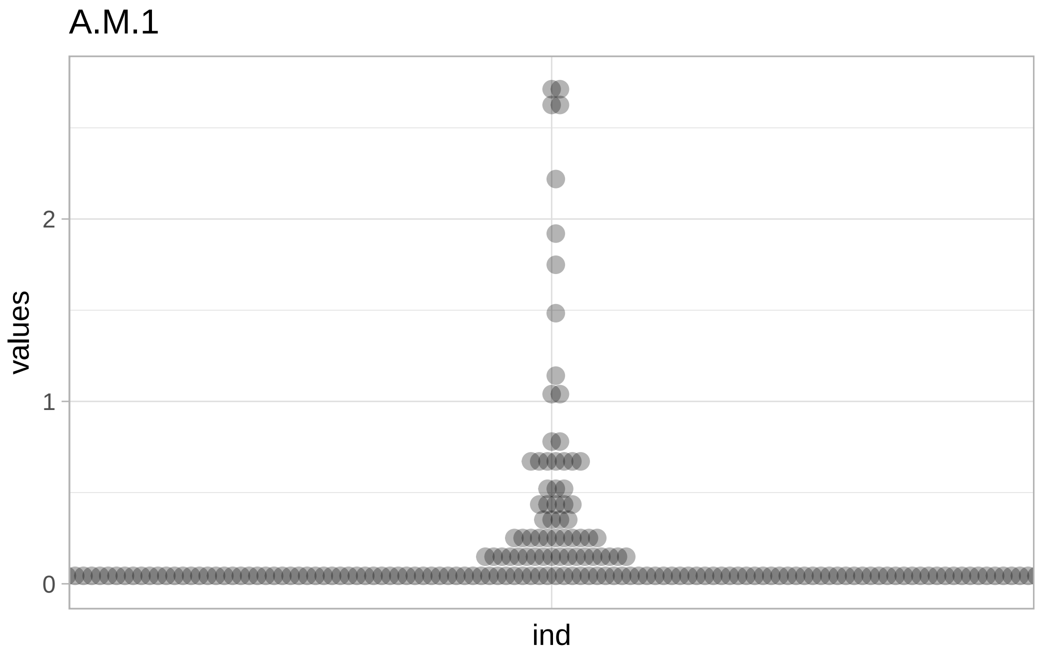
Zeros are not a problem per se, but quite often we find indicators with a very large proportion of zeros and just a handful of non-zero values. Or more generally we find that a large proportion of indicator values share the same value. This means that the power of the indicator to differentiate between units/regions is quite low. We can check this using the stats table.

```
df_stats[(df_stats$Frc.NonZero < 0.5) | (df_stats$Frc.Same > 0.5), c("iCode", "Frc.NonZero
```

```
  iCode Frc.NonZero Frc.Same
2 A.M.1        0.25     0.75
```

Here the indicator “A.M.1” is flagged (“# de personas refugiadas y solicitantes de asilo”). This is because 75% percent of its indicator values are zero and this can be confirmed visually:

```
plot_dist(MVI, dset = "Denominated", iCodes = "A.M.1", type = "Dot")
```



Whereas a very low data availability is a strong reason to exclude an indicator, here the flag is "weaker": this indicator has a low power to differentiate between municipalities, so is not hugely useful but if it is important to the framework it could be kept without causing any trouble.

Outliers

In composite indicator construction, a rule of thumb to identify distributions with outliers is any distributions that have absolute skew greater than 2 and kurtosis greater than 3.5. This is very much subjective but gives a rough idea and usually helps flag distributions that warrant a closer look. As for why we should be concerned about outliers at all, see my opinions here.

In any case let's see which indicators are flagged on this basis:

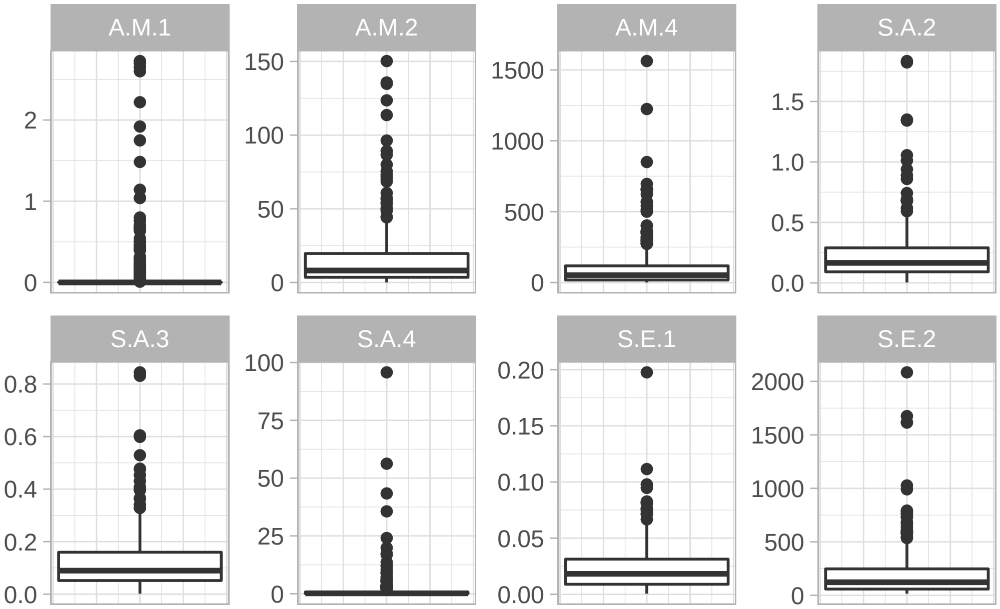
```r
df_outliers <- df_stats[df_stats$Flag.SkewKurt == "OUT", c("iCode", "Skew", "Kurt")]
knitr::kable(df_outliers, row.names = F)
```

iCode	Skew	Kurt
A.M.1	5.04	27.90
A.M.2	3.24	13.20
A.M.4	4.10	24.30

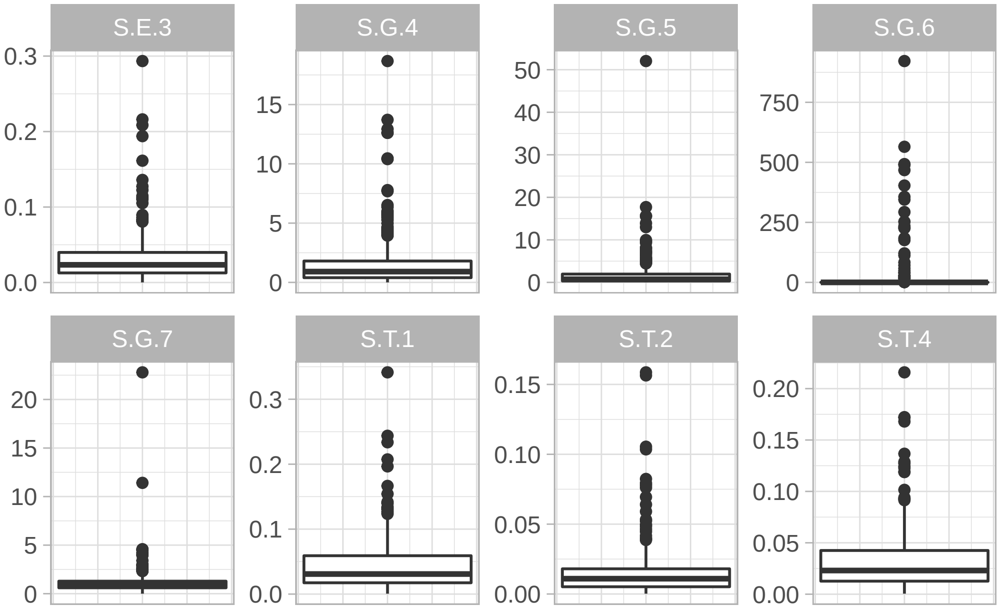
iCode	Skew	Kurt
S.A.2	3.32	16.50
S.A.3	2.67	10.80
S.A.4	9.12	101.00
S.E.1	2.74	14.70
S.E.2	3.73	20.00
S.E.3	3.43	17.90
S.G.4	3.71	18.00
S.G.5	8.70	108.00
S.G.6	6.25	46.50
S.G.7	10.50	142.00
S.T.1	2.70	11.90
S.T.2	4.06	23.30
S.T.4	2.42	8.25
C.E.4	9.10	95.10
C.E.6	2.62	8.09
C.E.8	3.14	14.30
C.E.9	5.75	55.70
C.E.10	2.99	14.20
C.E.11	4.17	22.40
C.I.5	3.47	19.20
C.I.6	4.30	29.00
C.I.7	5.23	44.50
C.J.2	10.90	157.00
C.S.4	3.19	13.10

There are a total of 27 indicators flagged, which is quite a lot. This deserves a closer look. I'll plot in groups of eight

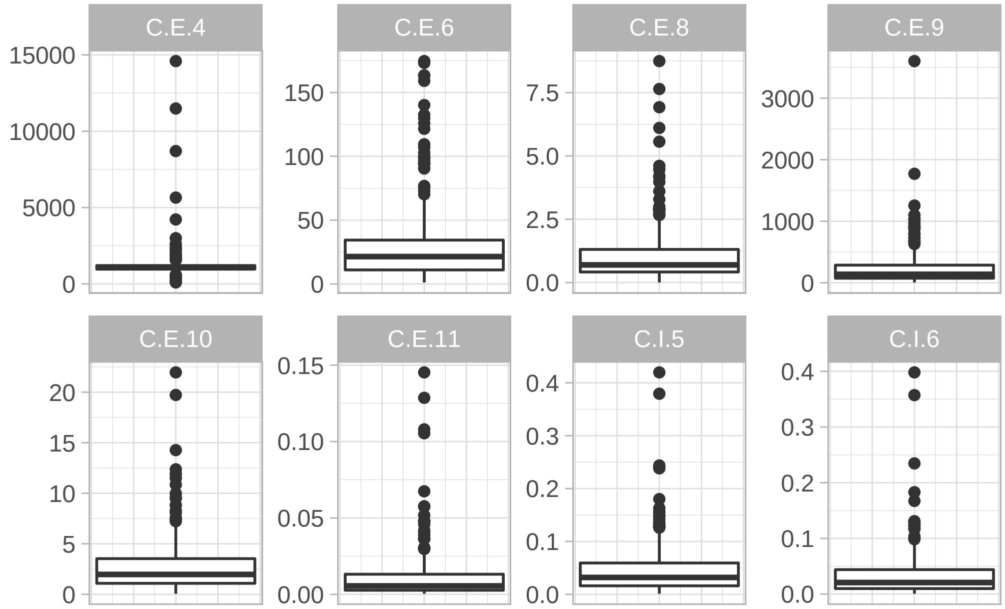
```
plot_dist(MVI, dset = "Denominated", iCodes = df_outliers$iCode[1:8])
```



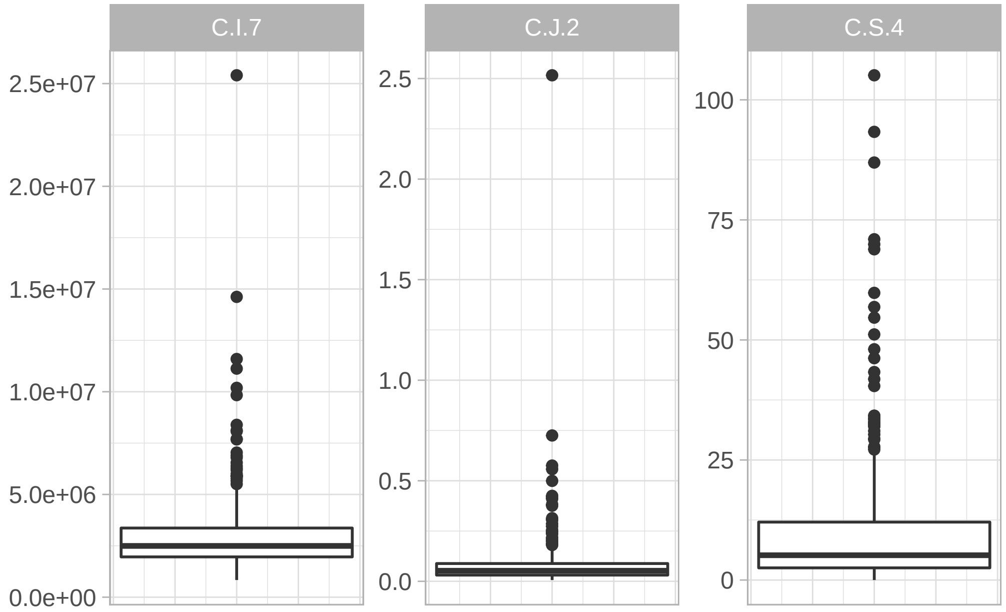
```r
plot_dist(MVI, dset = "Denominated", iCodes = df_outliers$iCode[9:16])
```



```r
plot_dist(MVI, dset = "Denominated", iCodes = df_outliers$iCode[17:24])
```



```r
plot_dist(MVI, dset = "Denominated", iCodes = df_outliers$iCode[25:27])
```



The plots here are all fairly similar and reflect log-normal distributions. We already know that most indicators have a reasonable proportion of unique values, so the high skew/kurtosis values are not caused by duplications. These are typical distributions that are associated with socioeconomic variables such as GDP, wealth, and so on (heavily skewed).

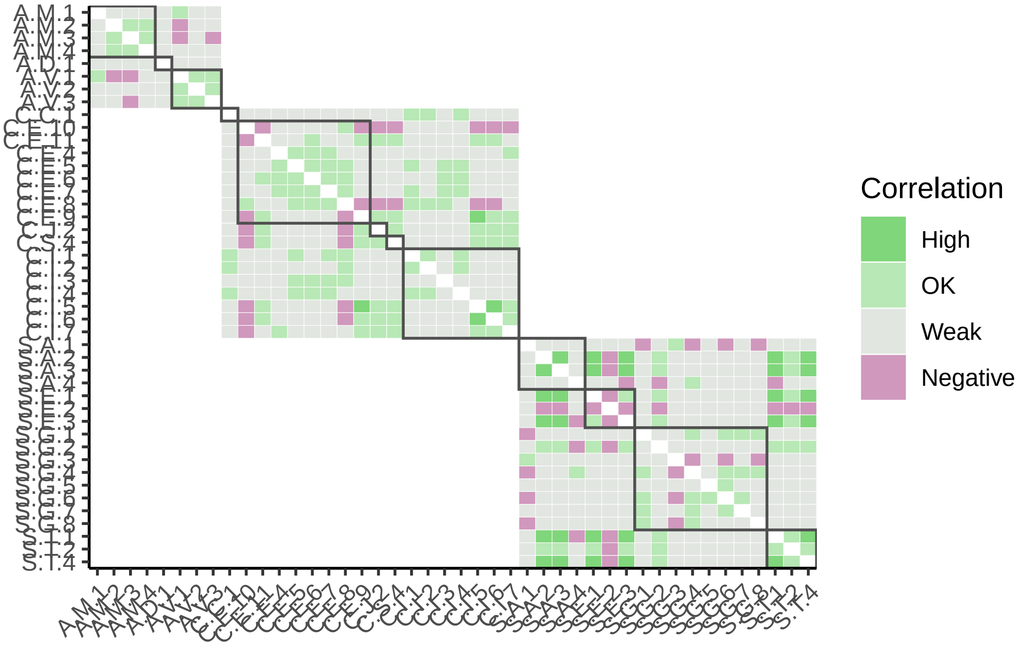
These indicators are not problematic but may need a data transformation, depending on the objectives of the index (to be discussed). The default here would be to apply a log transformation to each indicator before aggregation, which would make the distributions roughly normal.

Correlations

Here we look at the associations between indicators, using correlations. Although correlations are a linear measure of dependence and relationships could be nonlinear, linear dependence still does a good job of flagging strong associations which we can later check more closely if required.

Let us begin with a general plot of all correlations between indicators to get an idea for the correlation structure. We exclude correlations between indicators that are not in the same dimension, and we draw boxes around each categoria. We also account for the *directions* of indicators, i.e. indicators with negative directionality have their directions reversed before calculating correlations.

```r
plot_corr(MVI, dset = "Denominated", showvals = FALSE, box_level = 2,
          grouplev = 3, flagcolours = TRUE, cortype = "spearman", pval = 0, use_directions
```



This shows that most correlations can be considered "OK" (correlation 0.3-0.9) or "weak" (correlation between -0.4 and 0.3). Obviously these thresholds are rough. In any case the main things to look out for are strongly negative correlations (which almost cancel out in aggregation) or collinearities (which can imply double counting). COINr has a dedicated function for picking these up:

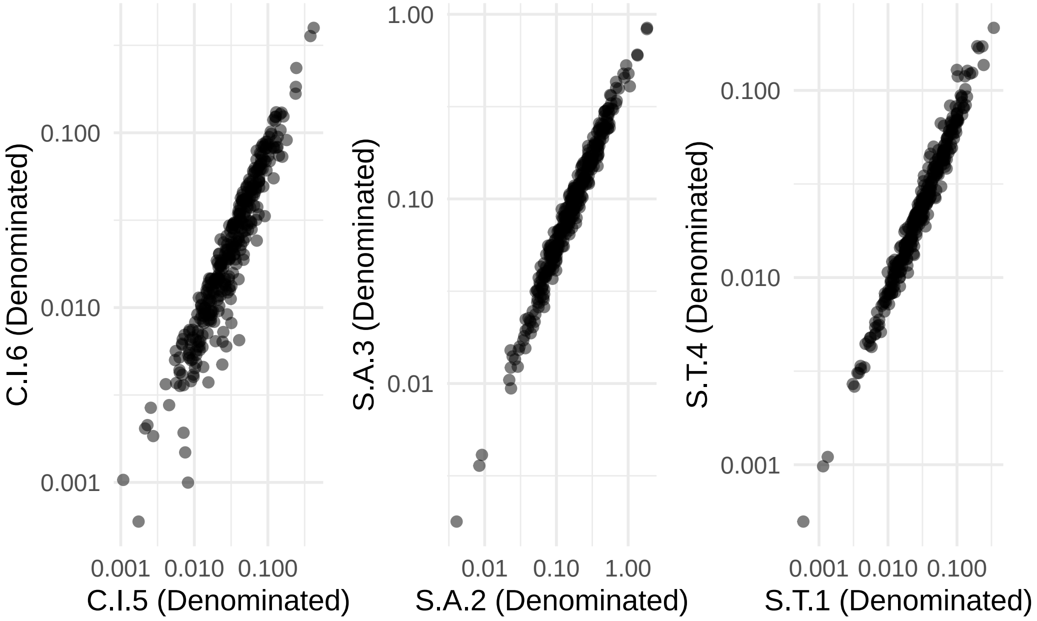
```r
# any correlations above 0.9 within the same categoria
get_corr_flags(MVI, dset = "Denominated", cor_thresh = 0.9, thresh_type = "high",
               grouplev = 2, cortype = "spearman", use_directions = TRUE) |>
  knitr::kable(row.names = F)
```

Group	Ind1	Ind2	Corr
Serv_Pub	C.I.5	C.I.6	0.956
Alim_Vivi	S.A.2	S.A.3	0.987
Trabajo	S.T.1	S.T.4	0.987

Here we see that there are three instances of near-collinear relationships between indicators within the same categoria. In this case, these indicators would be flagged, such that they could be considered for removal. It is anyway a good idea to plot these indicators

```
p1 <- plot_scatter(MVI, dsets = "Denominated", iCodes = c("C.I.5", "C.I.6"), log_scale = c
p2 <- plot_scatter(MVI, dsets = "Denominated", iCodes = c("S.A.2", "S.A.3"), log_scale = c
p3 <- plot_scatter(MVI, dsets = "Denominated", iCodes = c("S.T.1", "S.T.4"), log_scale = c

library(patchwork)
p1 + p2 + p3
```



We can see that the relationships are very strongly linear, although there is also a degree of scatter.

We can do the same for negative correlations:

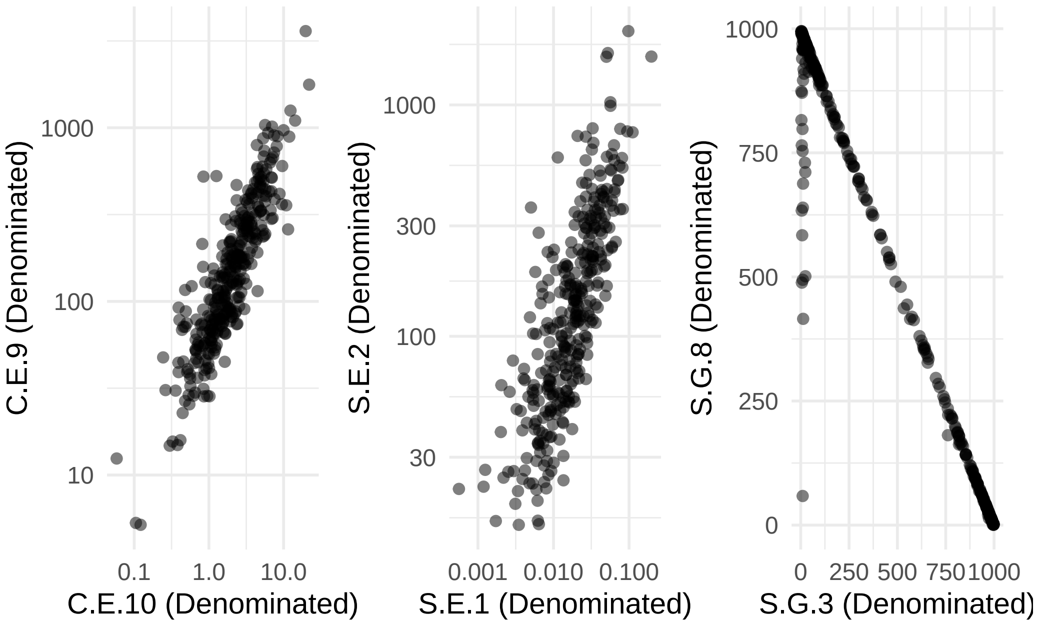
```
# any negative correlations below -0.4 within the same categoria
get_corr_flags(MVI, dset = "Denominated", cor_thresh = -0.4, thresh_type = "low",
               grouplev = 2, cortype = "spearman", use_directions = TRUE) |>
  knitr::kable(row.names = F)
```

Group	Ind1	Ind2	Corr
Educacion	C.E.10	C.E.11	-0.574
Educacion	C.E.10	C.E.9	-0.885
Educacion	C.E.8	C.E.9	-0.738
Economia	S.E.1	S.E.2	-0.808
Economia	S.E.2	S.E.3	-0.688
Grup_Vul	S.G.3	S.G.4	-0.550
Grup_Vul	S.G.3	S.G.6	-0.424
Grup_Vul	S.G.3	S.G.8	-0.939

Here we see some fairly strong negative correlations. We will visualise some of these:

```
p1 <- plot_scatter(MVI, dsets = "Denominated", iCodes = c("C.E.10", "C.E.9"), log_scale =
p2 <- plot_scatter(MVI, dsets = "Denominated", iCodes = c("S.E.1", "S.E.2"), log_scale = c
p3 <- plot_scatter(MVI, dsets = "Denominated", iCodes = c("S.G.3", "S.G.8"))

p1 + p2 + p3
```



Recall that the plots here are of the raw (denominated) indicator data which does not account for the directionality of indicators (unlike the correlation plots): here C.E.9 has negative directionality, as does S.E.2. The plot of S.G.3 against S.G.8 is of particular interest: it seems that where there are a lot of Maya people there are very few ladino/mestizo people. I wonder however if this is an artifact of the way one of these indicators is calculated - perhaps one is calculated based on the other?

Conclusions

This analysis has checked:

- Data availability
- Zeroes and duplicates
- Outliers
- Presence of collinearity and/or strong negative correlations within groups

The objective being to see if indicators are *statistically* suitable for inclusion into the index. This makes no consideration of *conceptual* matters such as whether each indicator is relevant to the concept of vulnerability at the municipal level.

The main outcomes could be divided into "Red flags" and "Orange flags" (any indicators not mentioned are green-flagged). I would not normally use this kind of terminology but looking ahead to some kind of function that automatically flags indicators we could consider this approach. Or just have one flag category.

Red flags

Issue: A.M.3 has a very low data availability at 12%

Recommendation Remove A.M.3 unless there is a very strong conceptual reason to keep it, or more data is expected to arrive soon.

Orange flags

- **Issue:** A.M.1 has 75% of municipalities sharing the same value (zero). This means it is not very useful in sorting between municipalities.
  - **Recommendation:** This indicator could be removed on the above basis but could equally be kept if conceptually important.
- **Issue:** In total 27 indicators are flagged as having "outliers" based on skew and kurtosis thresholds. This can cause issues in aggregation if not treated.
  - **Recommendation:** Consider using outlier treatment methods for these indicators: in particular use log transformations to transform from log-normal to normal distributions. An alternative could be to use rank transformations. Both can be performed in COINr.

- **Issue:** Three indicator pairs were flagged as being close to collinear (C.I.5/C.I.6, S.A.2/S.A.3 and S.T.1/S.T.4).
  - **Recommendation:** Inspect each indicator pair to see whether there are any conceptual redundancies and whether any indicators could be removed.
- **Issue:** Eight pairs of indicators were flagged as having strong or moderately-strong negative correlations (after accounting for directionality). Indicators: "C.E.10/C.E.11", "C.E.10/C.E.9", "C.E.8/C.E.9", "S.E.1/S.E.2", "S.E.2/S.E.3", "S.G.3/S.G.4", "S.G.3/S.G.6", "S.G.3/S.G.8".
  - **Recommendation:** Check the directionality of these indicators to see whether there are any mistakes. Check whether conceptually each indicator belongs to the correct aggregation group. In particular inspect the negative-collinear relationship of S.G.3 and S.G.8.

Summary table

```r
source("R/selection.R")

get_indicator_flags(MVI, "Denominated") |>
  knitr::kable(row.names = F)
```

iCode	Frc.Avail	Frc.Same	Skew	Kurt	Collinear	NegCorr
A.M.1	1.000	0.75000	5.040	27.90	NA	NA
A.M.2	1.000	0.00294	3.240	13.20	NA	NA
A.M.3	0.118	0.02500	1.190	2.26	NA	NA
A.M.4	1.000	0.00294	4.100	24.30	NA	NA
S.A.2	1.000	0.00294	3.320	16.50	S.A.3	NA
S.A.3	1.000	0.00294	2.670	10.80	S.A.2	NA
S.A.4	1.000	0.35000	9.120	101.00	NA	NA
S.E.1	1.000	0.00294	2.740	14.70	NA	S.E.2
S.E.2	1.000	0.00294	3.730	20.00	NA	S.E.1, S.E.3
S.E.3	1.000	0.00294	3.430	17.90	NA	S.E.2
S.G.3	1.000	0.00294	0.206	-1.76	NA	S.G.4, S.G.6, S.G.8
S.G.4	1.000	0.03530	3.710	18.00	NA	S.G.3
S.G.5	1.000	0.03820	8.700	108.00	NA	NA
S.G.6	1.000	0.21800	6.250	46.50	NA	S.G.3
S.G.7	1.000	0.00588	10.500	142.00	NA	NA
S.G.8	1.000	0.00294	-0.153	-1.74	NA	S.G.3
S.T.1	1.000	0.00294	2.700	11.90	S.T.4	NA
S.T.2	1.000	0.00294	4.060	23.30	NA	NA

iCode	Frc.Avail	Frc.Same	Skew	Kurt	Collinear	NegCorr
S.T.4	1.000	0.00294	2.420	8.25	S.T.1	NA
C.E.4	1.000	0.00294	9.100	95.10	NA	NA
C.E.6	1.000	0.00294	2.620	8.09	NA	NA
C.E.8	1.000	0.00294	3.140	14.30	NA	C.E.9
C.E.9	1.000	0.00294	5.750	55.70	NA	C.E.10, C.E.8
C.E.10	1.000	0.00294	2.990	14.20	NA	C.E.11, C.E.9
C.E.11	0.771	0.00382	4.170	22.40	NA	C.E.10
C.I.5	1.000	0.00294	3.470	19.20	C.I.6	NA
C.I.6	1.000	0.00294	4.300	29.00	C.I.5	NA
C.I.7	1.000	0.00294	5.230	44.50	NA	NA
C.J.2	1.000	0.00294	10.900	157.00	NA	NA
C.S.4	1.000	0.00294	3.190	13.10	NA	NA