A Tidy Framework and Infrastructure to Systematically Assemble Spatio-temporal Indexes from Multivariate Data

H. Sherry Zhang[1], Dianne Cook[1], Ursula Laa[2], Nicolas Langrené[3], Patricia Menéndez[1]

ARTICLE HISTORY
Compiled July 29, 2023

[1] Department of Econometrics and Business Statistics, Monash University, Melbourne, Victoria, Australia
[2] Institute of Statistics, University of Natural Resources and Life Sciences, Vienna, Austria
[3] Department of Mathematical Sciences, BNU-HKBU United International College, Zhuhai, Guangdong, China

ABSTRACT

- indexes, useful, quantify severity, early monitoring,
- A huge number of indexes have been proposed by domain experts, however, a large majority of them are not being adopted, reused, and compared in research or in practice.
- One of the reasons for this is the plenty of indexes are quite complex and there is no obvious easy-to-use implementation to apply them to user's data.
- The paper describes a general pipeline framework to construct indexes from spatio-temporal data,
- This allows all the indexes to be constructed through a uniform data pipeline and different indexes to vary on the details of each step in the data pipeline and their orders.
- The pipeline proposed aim to smooth the workflow of index construction through breaking down the complicated steps proposed by various indexes into small building blocks shared by most of the indexes.
- The framework will be demonstrated with drought indexes as examples, but appliable in general to environmental indexes constructed from multivariate spatio-temporal data

KEYWORDS
indexes; data pipeline; software design

1. Introduction

Indexes are commonly used to combine multivariate information into a single number for monitoring, communicating, and decision-making across various domains in society. Their applications can be found in the environment (e.g. Air Quality Index, El Niño-Southern

CONTACT: H. Sherry Zhang. Email: huize.zhang@monash.edu.

Oscillation Index), in the economy (Consumer Price Index, stock market indexes), and in the social sciences (e.g. QS University Rankings, Human Development Index).

To construct an index, experts typically first define an underlying concept of interest to measure. This concept is latent and often without direct measure but of social and public importance. Relevant and available variables are then defined, collected, and combined as an index using statistical methods to approximate the concept. This construction process is often not straight forward and many decisions need to be made, whether explicitly or not. For example, decision on which variables to include may depend on the data availability and the choice in other similar indexes. Indexes constructed from linear combination of variables need to decide on the weight assigned to each variable. Some indexes have a spatial and/or temporal component and variables can be aggregated to different spatial resolutions and temporal scales for different monitoring purpose. These decisions can result in different index values and have different practical implication.

To be able to test different decision choices systematically for an index, the index needs to be broken down into its fundamental building blocks to analyse the effect of each component. Such decomposition of indexes standardise the index construction pipeline and offers benefit for comparing among indexes and calculate index uncertainty. In social indexes, the OECD handbook (OECD, European Union, and Joint Research Centre - European Commission 2008) has provided a set of steps for constructing composite socio-economic indexes, but there is a need to extend these guidelines to accommodate more complex construction steps required for indexes in general.

In this work, we develop a data pipeline framework where indexes from different domains can be built from. Based on the pipeline, an R package, tidyindex, is developed for practitioners to construct indexes and understand how an index responds to changes in different components in the pipeline. This work provides researchers actively developing new indexes with tools to evaluate the proposed indexes on their robustness for wider adoption. It also helps index analysts to diagnose an index to identify its weakness for methodology improvement.

The rest of the paper is structured as follows: Section 2 reviews the concept of data pipeline in R. The pipeline framework for index construction is presented in Section 3. Section 4 explains how to include a new building block in each pipeline module. Examples are given in Section 5 to demonstrate the index construction with the pipeline built.

2. Data pipeline

Think about if there is another word for data pipeline

Why you should care about pipeline

Data pipeline is not a new concept to computing. It refers to a set of data processing elements connected in series, where the output of one element is the input of the next one. Wickham et al. (2009) argues that whether made explicit or not, the pipeline has to be presented in every graphics program. The paper also argues that breaking down graphic rendering into steps is beneficial for understanding the implementation and comparing between different graphic systems. The discussion on pipeline construction is well documented in early interactive graphics software: Buja et al. (1988), Sutherland et al. (2000), and Xie, Hofmann, and Cheng (2014) and their pipeline steps include non-linear transformation, variable standardization, randomization and dimension reduction.

What is pipeline, its underlying software design philosophy, and how these are reflected in R

One of the most commonly known pipeline examples is perhaps the Unix pipeline where programs can be concatenated with `|` to flow the output from the last program into the next program, i.e.

```
command 1 | command 2 | command 3 | ...
```

To solve a complex problem, the Unix system builds simple programs that do one thing well and work well together. This design is also reflected in the tidyverse ecosystem in R. To solve a complicated data problem using tidyverse, analysts typically build the solution using a collection of tools from the tidyverse toolbox. The data object can flow smoothly from one command to the next, safeguarded by the tidy data format (Wickham 2014), which prescribes three rules on how to lay out tabular data. The tidyverse tools also embrace a strong human-centered design where function names are intuitive and easy to reference through autocomplete. With the tidyverse design principle in mind, the tidymodel suite enables analysts to build machine learning models through the data pipeline. It includes typical tasks required in machine learning like data resampling, feature engineering, model fitting, model tuning, and model evaluation. An advantage of tidymodel pipeline over separate software for individual models is that analysts no longer need to write model-specific syntax to work with each model, but pipeline-specific syntax that is applicable to all the models implemented in tidymodel. This allows users to easily experiment with a collection of machine learning models.

Constructing indexes would also benefit from pipeline and embracing the aforementioned design philosophy.

In index construction, data pipeline is often presented in a workflow diagram in the research paper to illustrate how the raw data is transformed into the final indexes. This agrees with Wickham's argument on the presence of the data pipeline, however, more often than not, the pipeline is not made explicit in the software. Often the time, all the steps are lumped into a single wrapper function, rather than being split into smaller, modulated functions. This increases the cost of maintaining and understanding the code base, gives analysts little freedom to customise the indexes for specific needs, and hinders reusing existing code for building new indexes. A pipeline approach unites a range of indexes under a single data pipeline and analysts can compose indexes from pipeline steps like building Legos from individual bricks. In this workflow, analysts are not limited by indexes that have been already proposed and can easily combine pipeline steps to compose novel indexes. Analysis of the indexes (i.e. calculation of uncertainty) is also feasible by adding external code into the pipeline.

3. A pipeline for building statistical indexes

3.1. *How does the pipeline constructin of an index look like?*

Consider a commonly used drought index: Standardized Precipitation-Evapotranspiration Index (SPEI) (Vicente-Serrano, Beguería, and López-Moreno 2010). Its construction involves:

1) transform the average temperature (`TMED`) into potential evapotranspiration (`pet`)

2) combine precipitation (`prcp`) and potential evapotranspiration (`pet`) into a single variable `diff`
3) aggregate the difference series with a sliding window (`.scale`)
4) fit a distribution to the aggregated series, and
5) derive the index value from the normal density values.

Conventionally approach may combine all these steps into a single function, with some level of modularity. However, these modules may only work for the selected index offered by the package.

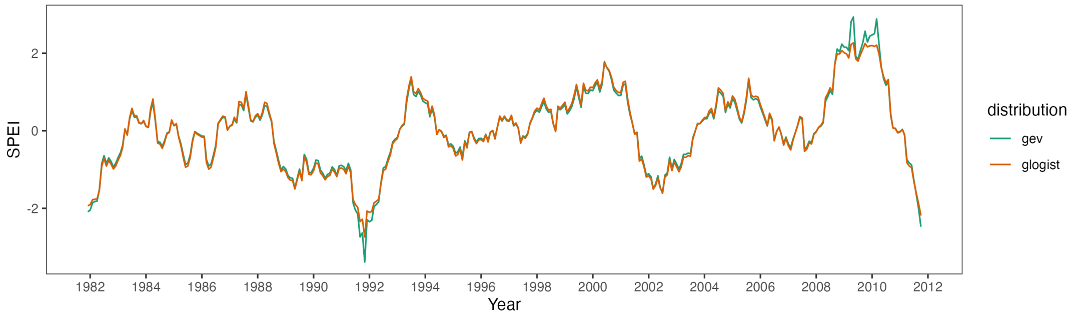
Under the pipeline approach, analysts first need to identify which module each step belongs to.

Below shows the pseudocode for constructing SPEI with the pipeline:

```
DATA %>%
  var_trans(.method = thornthwaite, Tave = TMED, ..., .new_name = "pet") %>%
  dim_red(diff = prcp - pet) %>%
  aggregate(.var = diff, .scale = 12, .new_name = "agg") %>%
  dist_fit(.method = "lmoms", .var = agg, .dist = DIST) %>%
  augment(.var = agg)
```

The pipeline construct allows for multiple `.scales` and `.dist` to be evaluated in `aggregate()` and `dist_fit()` to compare index under different parameterisations. The result can then be passed into the `ggplot2` to crease visualisation. **?@fig-toy-example** compares the SPEI calculated with two distributions (log-logistic and pearson III).

Apart from evaluating multiples parameters, the pipeline approach allows the the steps written for one index can be directly extrapolate to another index building within the pipeline. The flexibility of the pipeline also integrate well with other existing packages, for examples, fitting distributions using L-moment is commonly used when constructing drought indexes. The package `lmomco` provides general L-moment fits to a wide range of distributions and users can easily access to all the distributions within the pipeline.



3.2. *Pipeline steps for constructing indexes*

any index can be broken down into multiple steps and then we can do things with it: swap, change parameter, etc

variables == indicators

An overview of the pipeline is given in Figure 1 to illustrate the construction from raw data to the final indexes. The pipeline includes eight modules for operations in the spatial, temporal, and multivariate aspects of the data as well as modules for comparing and communicating indexes. Analysts can choose a subset of modules and reorder them as needed to construct an index.

Data pre-processing may happen before the start of the index pipeline to prepare multivariate data from different locations/ countries ready for constructing an index. For remote sensing data, it includes aligning the spatial resolution and temporal frequency of data collected from different satellite products, or merging in-situ stations with the satellite data.

The notation used for introducing the pipeline modules is as follows. Let $\mathbf{x}(\mathbf{s};\mathbf{t})$ denote the raw data with spatial, temporal, and multivariate aspects: the spatial dimension $\mathbf{s} = (s_1, s_2, \cdots, s_n)'$ is defined in the 2D space: $\mathbf{s} \in \mathcal{D}_s \subseteq \mathbb{R}^2$, the temporal dimension $\mathbf{t} = (t_1, t_2, \cdots, t_J)'$ is defined in the 1D space: $\mathbf{t} \in \mathcal{D}_t \subseteq \mathbb{R}$. When more than one variable is involved, the multivariate data can also be written as: $\mathbf{x}(\mathbf{s};\mathbf{t}) = (x_1(\mathbf{s};\mathbf{t}), x_2(\mathbf{s};\mathbf{t}), \cdots, x_P(\mathbf{s};\mathbf{t}))'$.

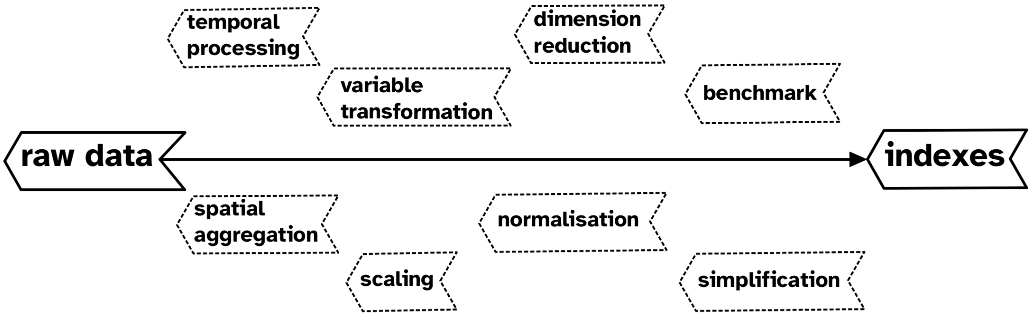


Figure 1. Diagram of pipeline steps for index construction. will need to be updated with better design and the distribution fitting step.

3.2.1. Temporal processing

The construction of an index sometimes needs to consider information from neighbouring time periods. The temporal processing is a general operator on the time dimension of the data in the form of

$$f_\psi(x(\mathbf{s};\mathbf{t})), \tag{1}$$

where $\psi \in \Psi \subseteq \mathbb{R}^{d_\psi}$ is the parameters associated with the temporal operation and d_ψ is the number of parameter of ψ. A typical example of temporal processing is aggregation, which is used in the drought index SPI to measure the lack of precipitation for meteorological drought. In SPI, monthly precipitation is aggregated by a time scale parameter k: $x(s_i; t_{j'}) = \sum_{j=j'-k+1}^{j'} x(s_i; t_j)$, where j' is the new time index after the aggregation. In this notation, each spatial location is separately aggregated and precipitation is summed from k month back, $j' - k + 1$, to the current period, j', to create the aggregated series, indexed by j'.

more explicit on k will influence 1) long term vs. short term, 2) uncertainty

The choice of time scales parameter k can result in variation in the calculated index values: a small k of 3 or 6 months produces the index more sensitive to individual months, while a large k of 24 or 36, an equivalent to a 2- or 3-year aggregation, gives dryness information relative to the long term condition. As will be shown in section [SECTION EXAMPLE], this variation may even lead to conflicting conclusions on the dry/wet condition of the area, highlighting the importance to account for index uncertainty when interpreting index values for decision-making.

Effective drought index

3.2.2. Spatial processing

Spatial processing may be needed when indexes are not calculated independently on each collected location or when variables collected from multiple sources need to be fused before further processing. The process can be written as a general operation in the form of

$$x(\mathbf{s}'; \mathbf{t}) = g_\theta(x(\mathbf{s}; \mathbf{t})), \tag{2}$$

where $\theta \in \Theta \subseteq \mathbb{R}^{d_\theta}$ is the associated parameters in the process and d_θ is the number of parameter of θ. An example of spatial processing is to align variables collected in different resolutions. When variables are collected at different resolutions, analysts may choose to down-sample those in a finer resolution, i, to match those in a coarser resolution, i'. This is a spatial aggregation and if aggregate using the mean, it can be written as

$$g(x) = \frac{\sum_{i \in i'} x}{n_{i'}}, \tag{3}$$

where $i \in i'$ includes all the cells from the finer resolution in the coarser grid and $n_{i'}$ is the number of observations falls into the coarser grid. Other examples of spatial processing include 1) borrowing information from neighbouring spatial locations to interpolate unobserved locations and 2) fusing variables from ground measures with satellite imageries.

3.2.3. Variable transformation

The purpose of variable transformation is to create variables that fits assumptions for further computing. These assumptions include a stable variance, normal distribution, or a certain scale required by some algorithms down the pipeline. Variable transformation is a general notion of a functional transformation on the variable:

$$h_\tau(x(\mathbf{s}; \mathbf{t})), \tag{4}$$

where $\tau \in T \subseteq \mathbb{R}^{d_\tau}$ is the parameter in the transformation if any, and d_τ is the number of parameter of τ. Transformation is needed for data that are highly skewed and some common transformations include log, quadratic, and square root transformation.

3.2.4. Scaling

While scaling can be seen as a specific type of variable transformation, it is separated into its own step to make the step explicit in the pipeline. The key difference between the two steps is that variable transformation typically changes the shape of the data while scaling only changes the data scale and can usually be written in the form of

$$[x(s_i;t_j) - \alpha]/\gamma. \tag{5}$$

For example, a z-score standardisation can be written in the above form with $\alpha = \bar{x}(s;t)$ and $\gamma = \sigma(s;t)$, a min-max standardisation uses $\alpha = \min[x(s_i,t_j)]$ and $\gamma = \max[x(s_i,t_j)] - \min[x(s_i,t_j)]$. Figure 2 shows a collection of variable pre-processing operations and uses color to differentiate whether the operation is a variable transformation or a scaling step. While both variable transformation and scaling are pre-processing steps, the scaling operations in green show the same distribution as the original data.

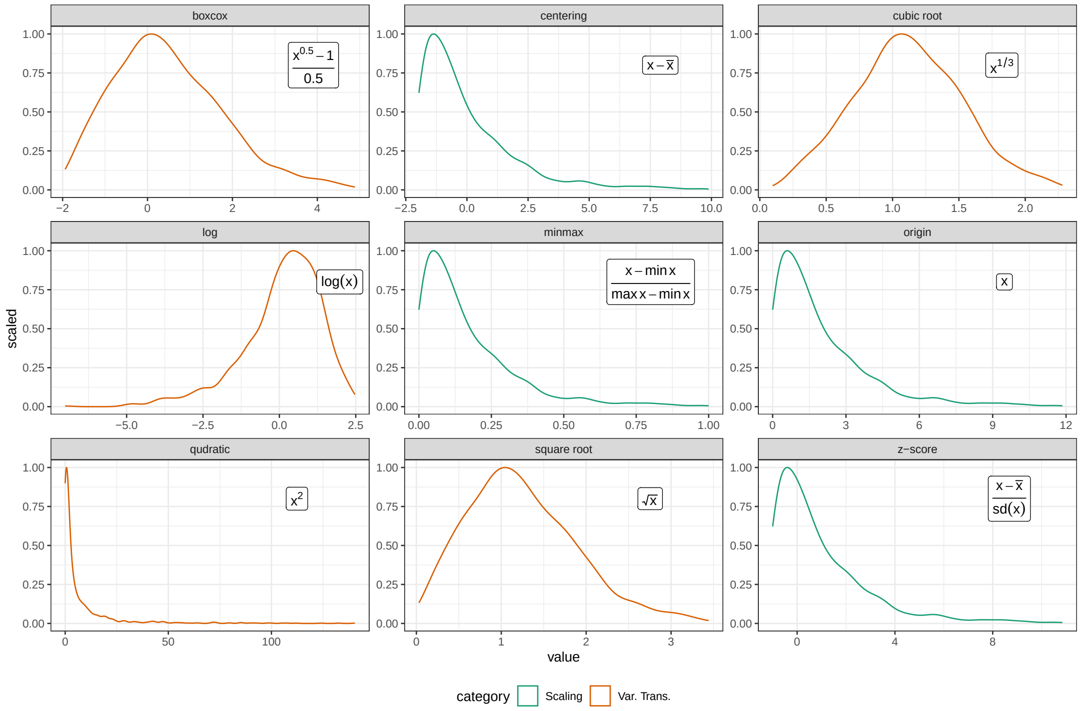


Figure 2. Comparison of operations in scaling (green) and variable transformation (orange) steps in free scale. Variables after the scaling operations have the same distribution as the origin, while the distribution changes after variable transformation.

3.2.5. Dimension reduction

Dimension reduction summarises the multivariate information into univariate, which can be denoted as:

$$x_p(\mathbf{s};\mathbf{t}) \rightarrow x(\mathbf{s};\mathbf{t}), \tag{6}$$

where $p = 1, 2, \cdots, P$. The combination can be based on domain-specific knowledge, originated from theories describing the underlying physical process. For example, the SPEI uses a water balance model ($D = P - \text{PET}$) to calculate the difference series (D) from precipitation (P) and potential evapotranspiration (PET).

Another widely used approach is linear combination, which aggregates a collection of variables in a linear additive structure, expressed as:

$$x(\mathbf{s};\mathbf{t}) = \sum_{p=1}^{P} \lambda_p x_p(\mathbf{s};\mathbf{t}),$$

where λ_p denotes the weight assigned to variable x_p. Most indexes uses an equal weight that sum to 1, which allows each variable to contribute equally to the index. A linear combination can also be viewed as a linear projection of multivariate information by the weight vector. Projecting data from higher to lower dimension inevitably leads to information loss. For example, two countries can receive similar index, but their components could significantly differ – one with average scores across all components and another with extreme scores the opposite ends. The selected set of weights need to be examined to understand its effect on the index and its implications for decision-making. To do this, analysts can vary the coefficients in the linear projection to observe how the index value and countries' ranking change. These changes can also be visualized using a method called tour by generating an animation of the projections among different sets of weights.

3.2.6. Distribution fit

model fit?

Distribution fit can be seen as the model fitting in its simplest term. It can be represented by

$$F_\eta(x(\mathbf{s};\mathbf{t})), \tag{7}$$

where $\eta \in H \subseteq \mathbb{R}^{d_\eta}$ is the distribution parameter and d_η is the number of parameter of η. A distribution fit typically aims at finding the distribution that best fits the data. Analysts may start from a pool of candidate distributions with a chosen fitting method and goodness of fit measure. While it is useful to find the ultimate best distribution to fits the data, from a probabilistic perspective, the fitting procedure itself has an uncertainty associated with the data fed and the parameter chosen. A reasonable alternative is to understand how much the index values can vary given different distributions, fitting methods, and goodness of fit tests, and whether these variations are negligible in a given application.

3.2.7. Normalising

This step maps the univariate series into a different scale, typically for ease of comparison across regions. For example, a normal scale, [0, 1], or [0, 100] may be favored for reporting certain indexes. In drought indexes, i.e. SPI or SPEI, the quantiles from the fitted distribution are converted into the normal scale via the normal reverse CDF function: $\Phi^{-1}(.)$. Normalising is usually used at the end of the pipeline and its main difference from the scaling step is that here the change of scale also changes the distribution of the variable. While being commonly used, this step can get criticism from analysts for forcing the data into the decided scale, which can be either unnecessary or inaccurately exaggerate or downplay the outliers. Also, the use of a normal scale needs to be interpreted with caution. Figure 3 illustrates the normal density not being directly proportional to its probability of occurrence. This is concerning, especially at the extreme values, since a small difference in the tail density can have magnitudes of difference in its probability of occurrence.

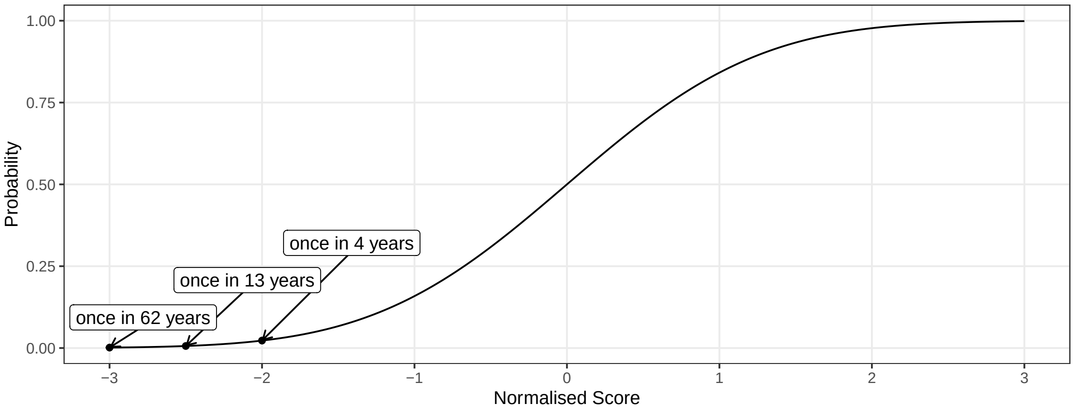


Figure 3. Scatterplot of normal quantiles against their density values. THree tail density values are highlighted with its probability of occurence labelled. Probability is calculated assuming monthly data: with a density of -2, the probability of occurrence is 1/pnorm(-2)/12 = 4 years. The non-linear relationship between the two quantities suggests normalised indexes need to be interpreted with caution since a slight change in the tail distribution can result in magnitudes of difference in its probability of occurrence.

3.2.8. Benchmarking

Benchmarking sets a constant value to allow the constructed index to be compared across time. Here we denote it with $u[x(s_i, t_j)]$ where u is a scalar of interest in the index constructed. A benchmark value could be a constant or a function of the data, i.e. mean.

3.2.9. Simplification

In public communication, the index values are usually accompanied by a categorical grade. The categorised grades are an ordered set of descriptive words or colors to communicate the severity or guide the comprehension of the indexes. The mapping from continuous index values to the discrete grades is called simplification in the pipeline and it can be written as a piece-wise function:

$$\begin{cases} C_0 & c_1 \leq (s_i;t_j) < c_0 \\ C_1 & c_2 \leq x(s_i;t_j) < c_1 \\ C_2 & c_3 \leq x(s_i;t_j) < c_2 \\ \cdots & \\ C_z & c_z \leq x(s_i;t_j) \end{cases} \tag{8}$$

where $C_0, C_1, \cdots, C_z$ are the categories and $c_0, c_1, \cdots, c_z$ are the thresholds for each category. In SPI, droughts are sorted into four categories: mild drought: $[-0.99, 0]$; moderate drought: $[-1.49, -1]$; severe drought: $[-1.99, -1.5]$, and extreme drought: $[-\infty, -2]$. In this case, C_0, C_1, C_2, C_3 are the drought categories: mild, moderate, severe, and extreme drought ($z = 3$) and $c_0 = 0, c_1 = -1, c_2 = -1.5, c_3 = -2$ are the cutoff value for each class.

4. Incorporating alternative methods into the pipeline components

5. Examples

This section uses the example of drought and social indexes to show the analysis made possible with the index pipeline. The drought index example computes two indexes with various time scales and distributions simultaneously using the pipeline framework to understand the flood and drought events in Queensland. The social index example focuses on the dimension reduction in Global Gender Gap Index to explore the impact of weight changes in linear combination on index value and country ranking.

5.1. *Every distribution, every scale, every index all at once*

A common task for drought researchers is to compute indexes at different parameter combinations. This can be used to identify the spatial and temporal extent of drought events, recommend the best parameter choice, or compare the effectiveness of indexes for monitoring drought. The example below computes two indexes: SPI and SPEI, at various time scales and fitted distributions, for stations in the state of Queensland in Australia. The purpose of the example is to demonstrate the interfaces the tidyindex package built to allow easy computing at different parameter combinations.

The state of Queensland in Australia is frequently affected by natural disaster events such as flood and drought, which can have significant impacts on its agricultural industry. This study uses daily data from Global Historical Climatology Network Daily (GHCND), accessed via the package `rnoaa` to examine drought/flood condition in Queensland. Daily data is average into monthly and stations are excluded if monthly data contains missings, which is required for calculating both SPI and SPEI. This gives 29 stations with complete records from 1990 January to 2022 April.

The function `compute_indexes()` can be used to collectively compute multiple indexes. The tidyindex offers wrapper functions, with the prefix idx_, that simplify the calculation of commonly used indexes by combining a set of pipeline steps into a single function. For example, the function `idx_spei()` includes the five steps previously described in Section 3.1 (variable transformation, dimension reduction, temporal aggregation, distribution fit, and normalise). Each `idx_xxx()` function specifies the relevant parameters relevant to the index: the thornthwaite method is used to calculate PET in SPEI, with the average temperature (`tavg`) and latitude (`lat`) used as inputs. The SPEI is computed at four time scales (6, 12, 24, and 36 months) and fitted with two distributions (Log-logistic and General Extreme Value (GEV)). The SPI is also computed at the same four time scales and uses the default gamma distribution to fit the aggregated series.

```
.scale <- c(6, 12, 24, 36)
(idx <- queensland %>%
  init(id = id, time = ym) %>%
  compute_indexes(
    spei = idx_spei(
      .pet_method = "thornthwaite", .tavg = tavg, .lat = lat,
      .scale = .scale, .dist = c(gev(), loglogistic())),
    spi = idx_spi(.scale = .scale)
  ))
```

```
[1] "Checking for missing values (`NA`): all the data must be complete. Input type is vec
```

```
# A tibble: 128,586 x 19
   .idx  .period id               ym  prcp  tmax  tmin  tavg  long   lat name    pet   dif
   <chr>   <dbl> <chr>         <mth> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr> <dbl>  <dbl
 1 spei        6 ASN00029~ 1990 Jun   170  29.7  16.2  23.0  142. -15.5 KOWA~  85.9   84.
 2 spei        6 ASN00029~ 1990 Jun   170  29.7  16.2  23.0  142. -15.5 KOWA~  85.9   84.
 3 spei        6 ASN00029~ 1990 Jun     0  23.0  11.8  17.4  139. -20.7 MOUN~  47.6  -47.
 4 spei        6 ASN00029~ 1990 Jun     0  23.0  11.8  17.4  139. -20.7 MOUN~  47.6  -47.
 5 spei        6 ASN00031~ 1990 Jun   794  25.8  18.1  21.9  146. -16.9 CAIR~  80.3  714.
 6 spei        6 ASN00031~ 1990 Jun   794  25.8  18.1  21.9  146. -16.9 CAIR~  80.3  714.
 7 spei        6 ASN00031~ 1990 Jun   504  23.0  13.8  18.4  145. -17.1 WALK~  48.0  456.
 8 spei        6 ASN00031~ 1990 Jun   504  23.0  13.8  18.4  145. -17.1 WALK~  48.0  456.
 9 spei        6 ASN00032~ 1990 Jun  1970  23.9  16.4  20.2  146. -17.6 SOUT~  70.2 1900.
10 spei        6 ASN00032~ 1990 Jun  1970  23.9  16.4  20.2  146. -17.6 SOUT~  70.2 1900.
# i 128,576 more rows
# i 6 more variables: .scale <dbl>, .agg <dbl>, .method <chr>, .fitted <dbl>,
#   .dist <chr>, .index <dbl>
```

The output from `compute_indexes()` contains index values and associated parameter in a long tibble. It includes the original variables (`id`, `ym`, `prcp`, `tmax`, `tmin`, `tavg`, `long`, `lat`, and `name`), index parameters (`.idx`, `.scale`, `.method`, and `.dist`), intermediate variables (`pet`, `.agg`, and `.fitted`), and the final index (`.index`). This data can be visualised across space or time, or simultaneously, to explore the wet/dry condition in Queensland. Figure 4 visualises the spatial distribution of SPI at two periods (2010 October - 2011 March and 2019 October - 2020 March) with significant natural disaster events: 2010/11 Queensland flood and 2019 Australia drought, which contributes to the notorious 2019/20 bushfire. Figure 5 displays the sensitivity of the SPEI series for one particular station, Texas post office, at different time scales and fitted distributions. These two plots demonstrate some possibilities to explore the indexes after they are computed from `compute_indexes()`.

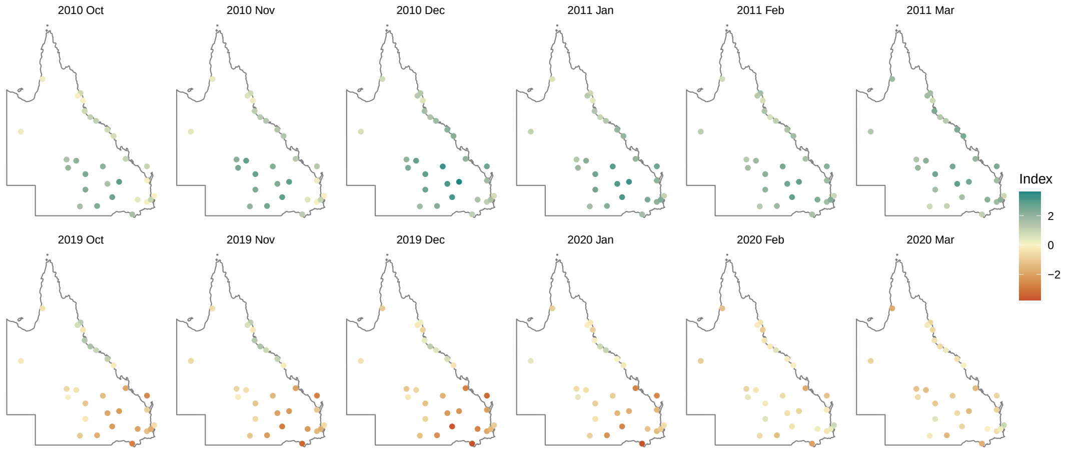


Figure 4. Spatial distribution of Standardized Precipitation Index (SPI-12) in Queensland, Australia during two major flood and drought events: 2010/11 and 2019/20. The map shows a continous wet period during the 2010/11 flood period and a mitigated drought situation, after its worst in 2019 December and 2020 Janurary, likely due to the increased rainfall in February from the meteorological record.

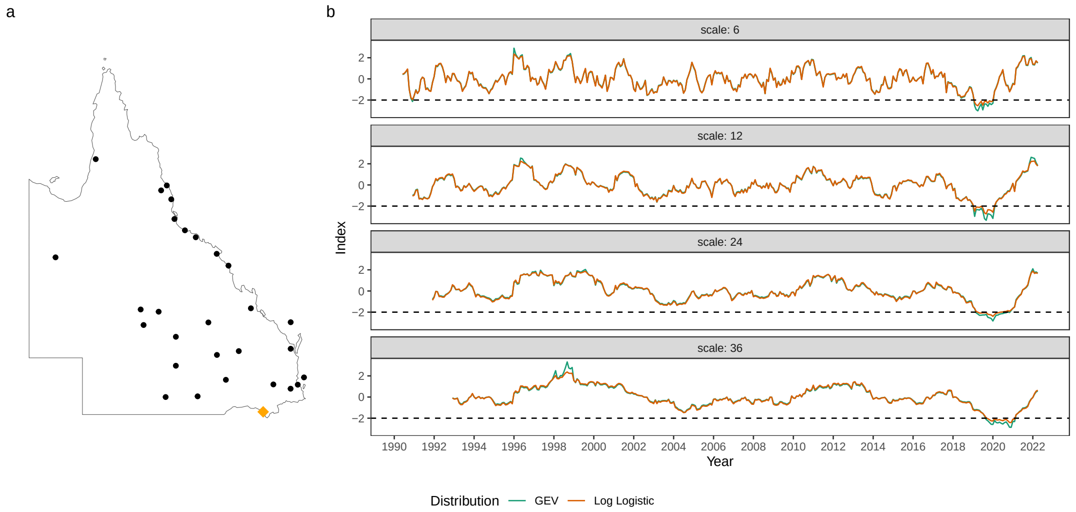


Figure 5. Time series plot of Standardized Precipitation-Evapotranspiration Index (SPEI) at the Texas post office station (highlighted by a diamond shape in panel a). The SPEI is calculated at four time scales (6, 12, 24, and 36 months) and fitted with two distributions (Log Logistic and GEV). The dashed line at -2 represents the class “extreme drought” by the SPEI. A larger time scale gives a smoother index series, while also takes longer to recover from an extreme situation as seen in the 2019/20 drought period. The SPEI values from two distribution fits mostly agree, while GEV can results in more extreme values, i.e. in 1998 and 2020.

5.2. *Does a minor change in variable weights cause a tornado?*

The Global Gender Gap Index (GGGI), published annually by the World Economic Forum, measures gender parity by assessing relative gaps between men and women in four key areas: Economic Participation and Opportunity, Educational Attainment, Health and Survival, and Political Empowerment (World Economic Forum 2023). The index is composed of 14 variables, expressed as female-to-male ratios, which are first aggregated in a linear combination into the four dimensions using the weight from the `V-weight` column in Table 1. The weight is calculated as the inverse of the standard deviation of each variable and scaling to sum to 1 within each dimension to allow a one percentage point change in the standard deviation of each variable to contribute equally to the index. The four dimensions are then aggregated in another linear combination with equal weight to obtain the index. The 2023 GGGI data is available from the Global Gender Gap Report 2023 in the country's economy profile and can be accessed in R via the `tidyindex` package as `gggi`, along with the corresponding weights `gggi_weights`.

Table 1. Weights of the fourteen variables in Global Gender Gap Index

Variable	Dimension	V-weight	D-weight	Weight
Labour force participation	Economy	0.199	0.25	0.050
Wage equality for similar work	Economy	0.310	0.25	0.078
Estimated earned income	Economy	0.221	0.25	0.055
Legislators senior officials and managers	Economy	0.149	0.25	0.037
Professional and technical workers	Economy	0.121	0.25	0.030
Literacy rate	Education	0.191	0.25	0.048
Enrolment in primary education	Education	0.459	0.25	0.115
Enrolment in secondary education	Education	0.230	0.25	0.058
Enrolment in tertiary education	Education	0.121	0.25	0.030
Sex ratio at birth	Health	0.693	0.25	0.173
Healthy life expectancy	Health	0.307	0.25	0.077
Women in parliament	Politics	0.310	0.25	0.078
Women in ministerial positions	Politics	0.247	0.25	0.062
Years with female head of state	Politics	0.443	0.25	0.111

A natural thing to do when provided with the index data is to reproduce the index. This helps index analysts to verify the index calculation and become familiar with the methodology. For GGGI, the construction can be simplified as a single linear aggregation step in the dimension reduction module, with the `Weight` column in Table 1, which is the product of the variable weight (`V-weight`) and dimension weight (`D-weight`).

```
gggi %>%
  init(id = country) %>%
  add_meta(gggi_weights, var_col = variable) %>%
  dimension_reduction(
    index_new = aggregate_linear(
      ~labour_force_participation:years_with_female_head_of_state,
      weight = weight))
```

The result can be compared with the GGGI values available in the report as shown in Figure 6, validating the reproducibility of the index for country with no missing variables.

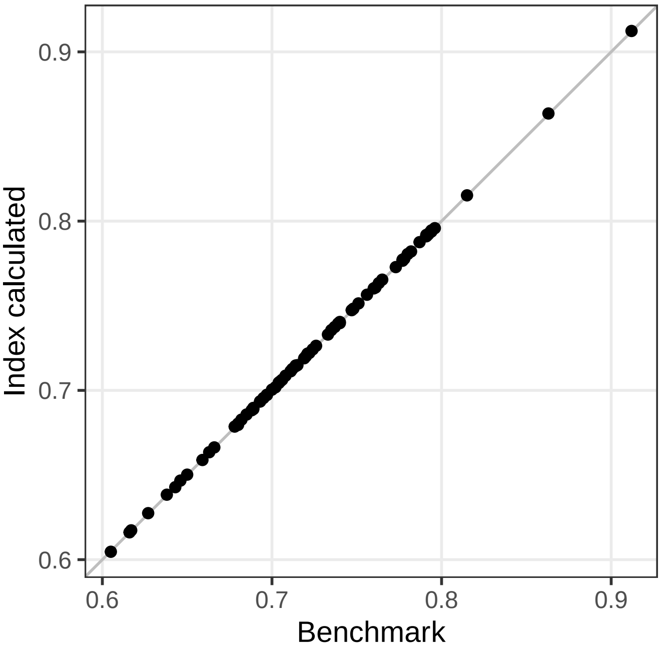


Figure 6. Verifying the calculation of the Global Gender Gap Index (GGGI). The index can be reproduced from the methodology described in the Global Gender Gap Report 2023 after removing the countries with missing variables, the treatment of which is unclear.

To understand the uncertainty of this dimension reduction step while avoiding the missingness issue on the variable level, we can run a local tour to slightly vary the weight of each dimension to see how index value and country ranking changes. We select countries in the South Asia and Sub-Saharan Africa region and gradually increase the weight of one variable and reduce back to equal weight all four dimensions, one at a time, to produce an animation of how GGGI changes in each country. Five frames (equal weights and one for each dimension with a relatively higher weight) selected from the tour animation are shown in Figure 7 and you can find the link to the full animation in figure caption.

Many insights can be derived from the animation and these selected frames. These can be helpful in understanding the sensitivity of the index value and country ranking to each variable and identifying the strengths and weaknesses of each country relative to others. When economy and education are given a greater weight, the variation in index values and ranking highlights countries that performs relatively better (moving right) or worse (moving left) in these two single dimensions. For example, the index moves to the left for Ethiopia and Senegal when economy is given a higher weight. This reveals the economy as a weakness for these two countries compared to their similarly-ranked peers. When health receives a higher weight, the impact on the index value is minimal. Notably, increasing the weight in politics leads to a pronounced decrease of index value for almost all the countries. Given the index value has a direct interpretation on how much of the gender gap has been closed, such a weight variation can cast a negative light on the progression to gender parity. Analysts need to further investigate the construction of the politics dimension and consider whether this is desirable for the index.

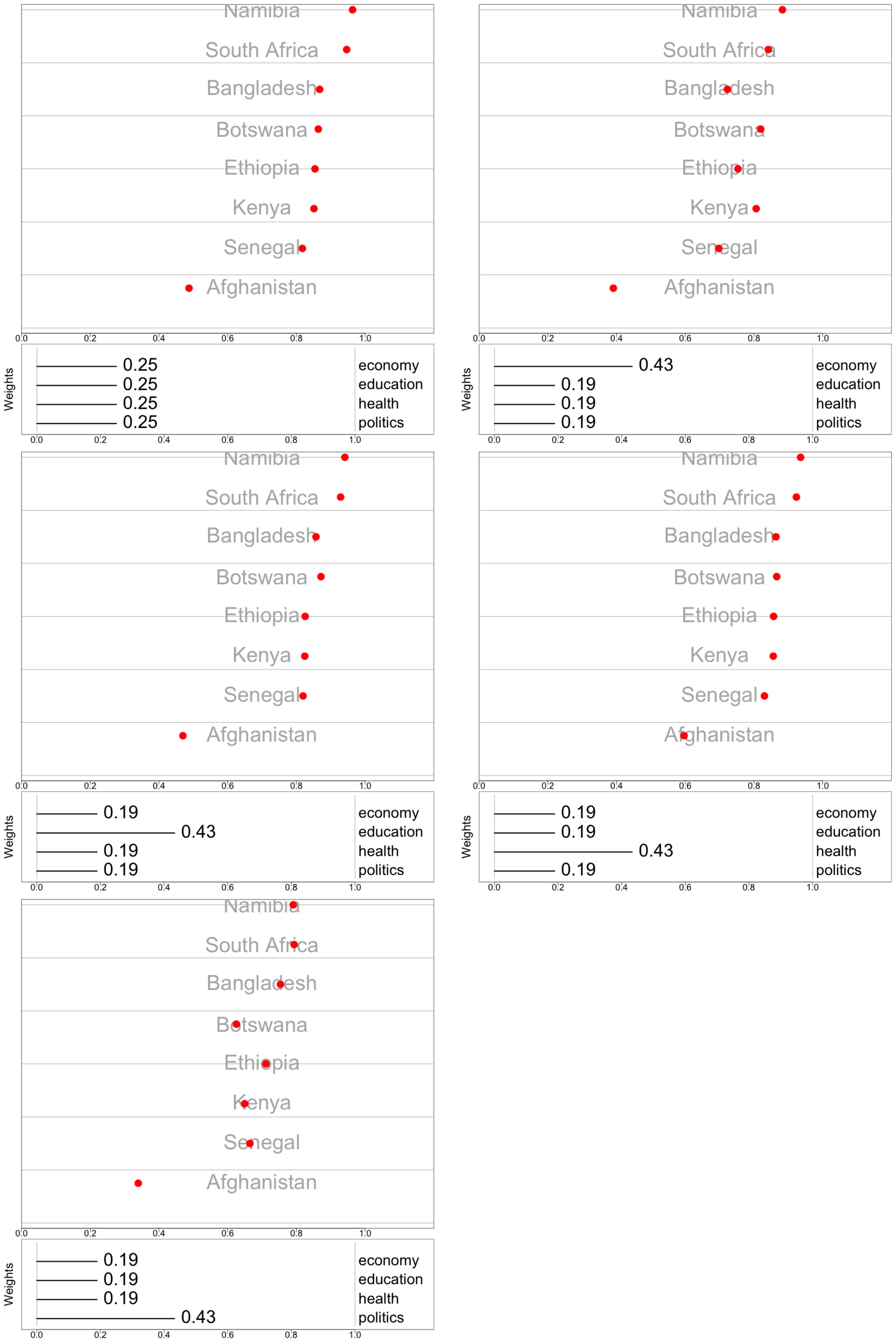


Figure 7. Five frames selected from varying the linear weights of four dimensions in Global Gender Gap Index. The weights vary slightly from the official simple average weights (0.25, 0.25, 0.25, 0.25) to observe how the index and ranking reponse. Full animation is available at https://vimeo.com/847874016?share=copy.

6. Conclusion

The paper presents a data pipeline with nine modules for constructing and analysing indexes. The pipeline increases transparency in the practice for index analysts to experiment with different index design and parameter choices to better design and apply their indexes. The significance of this work is its ability to provide a universal framework for index construction, which can be applied across different domains.

Examples have been given in the drought indexes and human development index to demonstrate computing of indexes with different parameters combinations and how alternative index design can provide insights to understand distinctive country characteristics that could sometimes be overlooked. The accompanied package, tidyindex, is not meant to provide comprehensive implementation for all indexes across all domains. Instead, it demonstrates implementing individual pipeline steps that are versatile to multiple indexes and composing new indexes from existing steps. Domain experts are welcomed to adopt the pipeline approach to develop specialised packages for specific-domains indexes.

Future work: - integrate more complex dimension reduction methods to calculate weights - strengthen the spatial processing module

Reference

Buja, A, D Asimov, C Hurley, and JA McDonald. 1988. "Elements of a Viewing Pipeline for Data Analysis." In *Dynamic Graphics for Statistics*, 277–308. Wadsworth, Belmont.

OECD, European Union, and Joint Research Centre - European Commission. 2008. *Handbook on Constructing Composite Indicators: Methodology and User Guide*. OECD. https://doi.org/10.1787/9789264043466-en.

Sutherland, Peter, Anthony Rossini, Thomas Lumley, Nicholas Lewin-Koh, Julie Dickerson, Zach Cox, and Dianne Cook. 2000. "Orca: A Visualization Toolkit for High-Dimensional Data." *Journal of Computational and Graphical Statistics* 9 (3): 509–29. https://www.jstor.org/stable/1390943.

Vicente-Serrano, Sergio M., Santiago Beguería, and Juan I. López-Moreno. 2010. "A Multiscalar Drought Index Sensitive to Global Warming: The Standardized Precipitation Evapotranspiration Index." *Journal of Climate* 23 (7): 1696–1718. https://journals.ametsoc.org/view/journals/clim/23/7/2009jcli2909.1.xml.

Wickham, Hadley. 2014. "Tidy Data." *Journal of Statistical Software* 59 (September): 1–23. https://doi.org/10.18637/jss.v059.i10.

Wickham, Hadley, Michael Lawrence, Dianne Cook, Andreas Buja, Heike Hofmann, and Deborah F. Swayne. 2009. "The Plumbing of Interactive Graphics." *Computational Statistics* 24 (2): 207–15. https://doi.org/10.1007/s00180-008-0116-x.

World Economic Forum. 2023. "The Global Gender Gap Report 2023." https://www3.weforum.org/docs/WEF_GGGR_2023.pdf.

Xie, Yihui, Heike Hofmann, and Xiaoyue Cheng. 2014. "Reactive Programming for Interactive Graphics." *Statistical Science* 29 (2): 201–13. https://www.jstor.org/stable/43288470?seq=1.