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] ⓘ

[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

1. Introduction

Indexes are commonly used to combine and summarize different sources of information into a single number for monitoring, communicating, and decision-making. Examples of those include the Air Quality Index, El Niño-Southern Oscillation Index, Consumer Price Index, QS University Rankings or Human Development Index among many others.

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

To construct an index, experts typically start by defining a concept of interest that requires measurement. This concept often lacks a direct measurable attribute or can only be measure as a composite of various processes, yet it holds social and public significance. To create an index, once the underlying processes involved are identified, relevant and available variables are then defined, collected, and combined using statistical methods into an index that aims at measure the process of interest. The construction process is often not straightforward, and decisions need to be made, such as the selection of variables to be included, which might depend on data availability and the statistical definition of the index to be used, among others. For instance, the 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 components, and variables can be aggregated to different spatial resolutions and temporal scales, leading to various indexes for different monitoring purposes. Hence, all these decisions can result in different index values and have different practical implications.

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 contribution and effect of each component. We call this process of breaking the index construction into different steps the index pipeline. Such decomposition of index components provides the means to standardise index construction via a pipeline and offers benefits for comparing among indexes and calculating index uncertainty. In social indexes for instance, the OECD handbook (OECD, European Union, and Joint Research Centre - European Commission 2008) has provided a set of steps and recommendations for constructing composite socio-economic indexes. However, there is still a need to extend these guidelines and methods to accommodate the inclusion of more methodological complex steps required for indexes in general.

In this work, we aim at providing statistical and computational methods to develop a data pipeline framework for constructing and customize indexes using data. As a companion the R package, tidyindex, is developed to construct and explore different. Furthermore, the tidyindex package can be used to explore the effects of changes on index definition, index construction steps, and construction methods. This work provides researchers actively developing new indexes or aiming at improving indexes with the tools to easily modify and evaluate indexes. It also helps index analysts diagnose indexes, carry out sensitivity analysis to assess small perturbations in the index, and identify potential weaknesses 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 add new steps into the index methodology in the form of new building blocks inside the index construction pipeline. Examples are given in Section 5 to demonstrate the index construction with the pipeline built.

2. Tidy framework

The tidy framework consists of two key components: tidy data and tidy tools. The concept of tidy data (Wickham 2014) prescribes specific rules for organising data in an analysis, with observations as rows, variables as columns, and types of observational units as tables. Tidy tools, on the other hand, are concatenated in a sequence through which the tidy data flows, creating a pipeline for data processing and modelling. These

pipelines are data centric, meaning all the tidy tools or functions take a tidy data object as inputs and return a processed tidy data object, directly ready for the next operations to be applied. Also, the pipeline approach corresponds to the modular programming practice, which breaks down complex problems into smaller and more manageable pieces, as oppose to a monolithic design, where all the steps are predetermined and integrated into a single piece. The flexibility provided by the modularity makes it easier to modify certain steps in the pipeline and to maintain and extend the code base.

Examples of using a pipeline approach for data analysis can be traced back to the interactive graphics literature, including Buja et al. (1988); Sutherland et al. (2000); Xie, Hofmann, and Cheng (2014); Wickham et al. (2009). Wickham et al. (2009) argues that whether made explicit or not, a pipeline has to be presented in every graphics program and making them explicit is beneficial for understanding the implementation and comparing between different graphic systems. While this comment is made in the context of interactive graphics program, it is also applicable generally to any data analysis workflow. More recently, the tidyverse suite (Wickham et al. 2019) takes the pipeline approach for general-purpose data wrangling and has gained popularity within the R community. The pipeline-style code can be directly read as a series of operations applied successively on tidy data object, offering a method to document the data wrangling process with all the computational details for reproducibility.

Since the success of tidyverse, more packages have been developed to analyze data using the tidy framework for domains specific applications, a noticeable example of which is `tidymodels` for building machine learning models (Kuhn and Wickham 2020). To create a tidy workflow tailored to a specific domain, developers first need to identify the fundamental building blocks to create a workflow. These components are then implemented as modules, which can be combined to form the pipeline. For example, in supervised machine learning models, steps such as data splitting, model training and model evaluation are commonly used in most workflow. In the `tidymodels`, these steps are correspondingly implemented as package `rsample`, `parsnip`, and `yardstick`, agnostic to the specific model chosen. The uniform interface in tidymodels frees analysts from recalling model-specific syntax for performing the same operation across different models, increasing the efficiency to work with different models simultaneously.

For constructing indexes, the pipeline approach adopts explicit and standalone modules that can be assembled in different ways. Index developers can choose the appropriate modules and arrange them accordingly to generate the data pipeline that is needed for their purpose. The pipeline approach provides many advantages:

- makes the computation more transparent, and thus more easily debugged.
- allows for rapidly processing new data to check how different features, like outliers, might affect the index value.
- provides the capacity to measure uncertainty by computing confidence intervals from multiple samples as generated by bootstrapping to original data.
- enables systematic comparison of surrogate indexes designed to measure the same phenomenon.
- it may even be possible to automate diagramatic explanations and documentation of the index.

Adoption of this pipeline approach would provide uniformity to the field of index development, research and application.

3. Details of the index pipeline

In constructing various indexes, the primary aim is to transform the data, often multivariate, into a univariate index. Spatial and temporal considerations are also factored into the process when observational units and time periods are not independent. However, despite the variations in contextual information for indexes in different fields, the underlying statistical methodology remains consistent across diverse domains. Each index can be represented as a series of modular statistical operations on the data. This allows us to decompose the index construction process into a unified pipeline workflow with a standardized set of data processing steps to be applied across different indexes.

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.

Now, we introduce the notation used for describing pipeline modules. Consider multivariate spatio-temporal data,

$$
\begin{aligned}
\mathbf{x}(s;t) &= [x^1(s;t), x^2(s;t), \cdots, x^P(s;t)] \\
&\quad p = 1, 2, \cdots P, s \in D_s \subseteq \mathbb{R}^d, t \in D_t \subseteq \mathbb{R}^m
\end{aligned}
\tag{1}
$$

where x^l is the pth variable, s is the spatial location, and t is the temporal order. The spatial space, $D_s \subseteq \mathbb{R}^d$, is generic to include different spatial representations, including a 1D set of n locations $\{D_s : s_1, s_2, \cdots, s_n\}$, a 2D set of longitude and latitude pairs, and d-D coordinate set. The higher dimension in the temporal space, $D_t \subseteq \mathbb{R}^m$, can be used to represent for a m-D deconstruction of time. For example, a year-month combination, $(Y, M)'$, can be used to record monthly precipitation series from January 1990 to December 2020, where $Y = 1990, \cdots, 2020$ and $M = 1, \cdots, 12$. To simplify the notation, a subscript letter s or t is used to denote the data have a fixed spatial locations or temporal order. When the spatial locations are fixed, the data become multivariate time series, denoted as $x_s^l(t)$. Similarly, $x_t^l(s)$ represents multivariate spatial data when the temporal dimension is fixed.

3.1. *Temporal processing*

`Input`: $x_s(t)$,

`Output`: $x_s^l(t')$, where $t' \in D_{t'}$.

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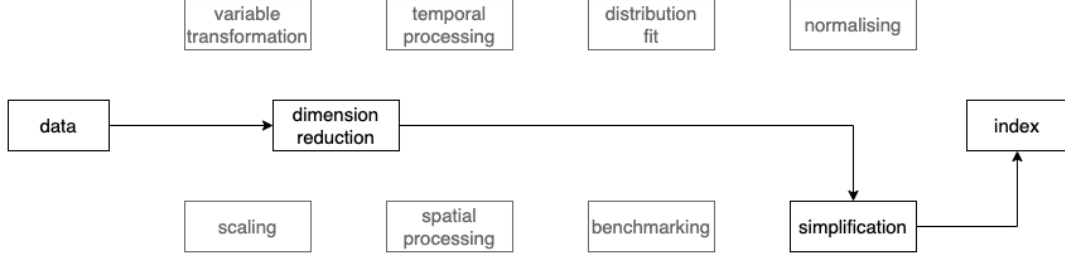
Figure 1. Diagram of pipeline steps for index construction. will need to be updated with better design and the distribution fitting step.

$$x(\mathbf{s}; \mathbf{t}') = f_\psi(x(\mathbf{s}; \mathbf{t})), \qquad (2)$$

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 and in the drought community, these indexes are often called multi-scalar indexes. For example, Standardized Precipitation Index (SPI) aggregates monthly precipitation using a time scale parameter k in the form of

$$x(s_i; t_{j'}) = \sum_{j=j'}^{j'+k-1} x(s_i; t_j),$$

where j' is the new time index after the aggregation. Multi-scalar drought indexes are widely used to study different types of drought events. A smaller value of the time scales parameter k is used for short term drought events, while a large k of 18 or 24, an equivalent to a 1.5- or 2-year aggregation, is often used for long term assessment.

3.2. *Spatial processing*

`Input:` $x_t^l(s)$

`Output:` $x_t^l(s')$ where $s' \in D_{s'}$

`Computation:` 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

$$x(\mathbf{s}'; \mathbf{t}) = g_\theta(x(\mathbf{s}; \mathbf{t})), \qquad (3)$$

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

$$g(x) = \frac{\sum_{i \in i'} x}{n_{i'}}, \tag{4}$$

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.

3.3. *Variable transformation*

`Input`: $x^l(s;t)$

`Output`: $y^l(s;t)$

`Computation`: 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:

$$h_\tau(x(\mathbf{s};\mathbf{t})), \tag{5}$$

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.

3.4. *Scaling*

`Input`: $x^l(s;t)$

`Output`: $x^l(s;t)$

`Computation`: 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

$$[x(s_i;t_j) - \alpha]/\gamma. \tag{6}$$

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.
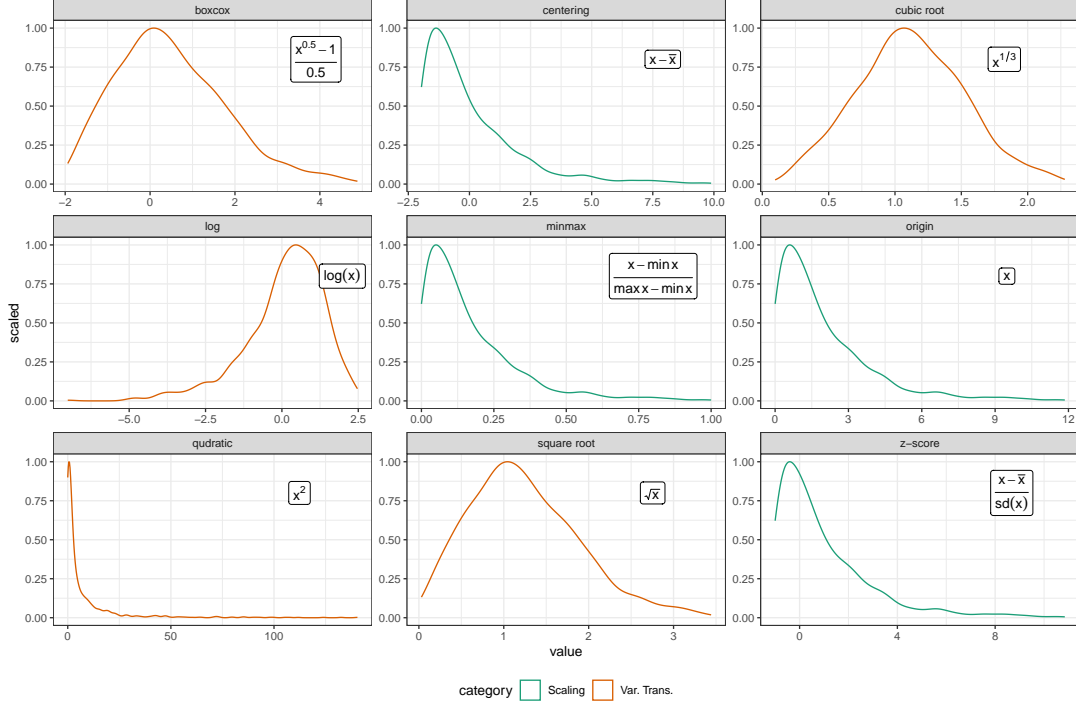

Figure 2. Comparison of operations in scaling (green) and variable transformation (orange) steps in free scale. Variables after the scaling operations have the same distribution as the origin, while the distribution changes after variable transformation.

3.5. *Dimension reduction*

`Input`: $x^l(s; t)$

`Output`: $x^{l'}(s; t)$, where $l' = 1, 2, \cdots, L'$

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

$$x_p(\mathbf{s}; \mathbf{t}) \rightarrow x(\mathbf{s}; \mathbf{t}), \tag{7}$$

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

7

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.

3.6. *Distribution fit*

`Input`: $x^l(s;t)$

`Output`:

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

$$F_\eta(x(\mathbf{s};\mathbf{t})), \tag{8}$$

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.

3.7. *Normalising*

`Input`: $x^l(s;t)$

`Output`:

`Computation`: 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.
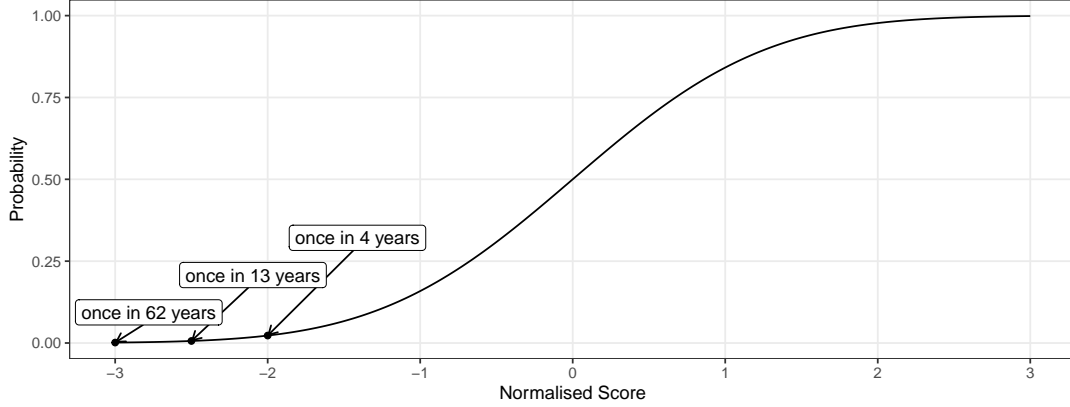

Figure 3. Scatterplot of normal quantiles against their density values. THree tail density values are highlighted with its probability of occurence labelled. Probability is calculated assuming monthly data: with a density of -2, the probability of occurrence is $1/\text{pnorm}(-2)/12 = 4$ years. The 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.

3.8. *Benchmarking*

`Input`: $x^l(s;t)$

`Output`:

$D(x^l(s,t), a)$ where $D(.,.)$ is a distance function between

`Computation`: 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.

3.9. *Simplification*

`Input`: $x^l(s;t)$

`Output`: $C^l(s;t)$ where $\{C : c_0, c_1, \cdots, c_z\}$ is a discrete set

`Computation`: 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:

$$
\begin{cases}
c_0 & C_1 \leq x^l(s;t) < C_0 \\
c_1 & C_2 \leq x^l(s;t) < C_1 \\
c_2 & C_3 \leq x^l(s;t) < C_2 \\
\cdots \\
c_z & C_z \leq x^l(s;t)
\end{cases}
\tag{9}
$$

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.   Software design

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.
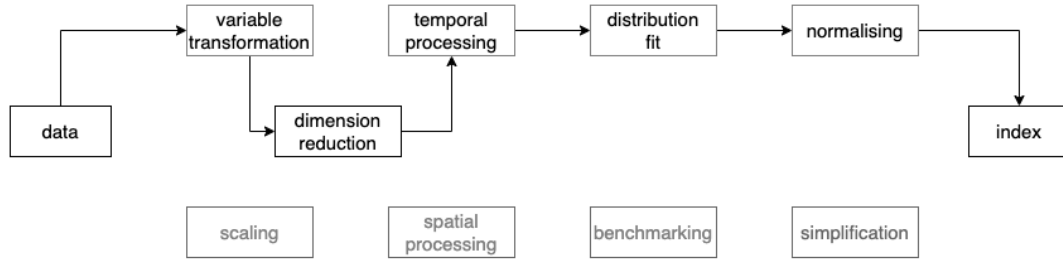
Figure 4. Diagram of pipeline steps for index construction. will need to be updated with better design and the distribution fitting step.

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 **?@sec-toy-example** (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)
  )
```

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 5 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 6 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()`.
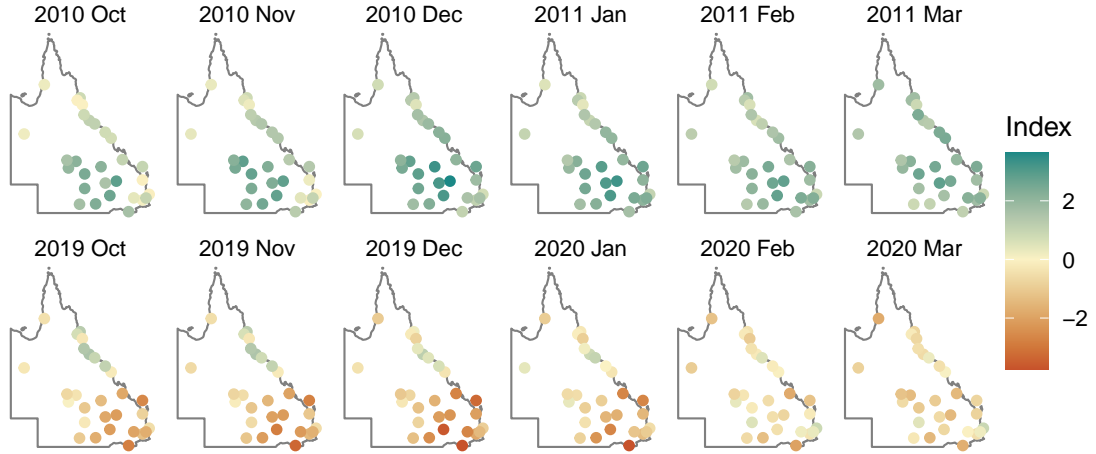
Figure 5. Spatial distribution of Standardized Precipitation Index (SPI-12) in Queensland, Australia during two major flood and drought events: 2010/11 and 2019/20. The map shows a continous wet period during the 2010/11 flood period and a mitigated drought situation, after its worst in 2019 December and 2020 Janurary, likely due to the increased rainfall in February from the meteorological record.

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

Variable	Dimension	V-weight	D-weight	Weight
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, validating the reproducibility of the index for country with no missing variables.

To understand the uncertainty of this dimension reduction step from combining four dimensions (Economy, Education, Health, and Politics), we run a local tour [TODO: reference to local tour] that varies the weight of each dimension slightly to observe how index value and country ranking changes. The local tour produces an animation that gradually increases the weight of one variable and reduces it back to equal weight, one at a time. A set of countries are selected for this exercise, which includes 1) the top four with GGGI > 0.85, 2) a set of countries with GGGI between 0.72 and 0.73 (Brazil, Panama, Poland, Bangladesh, Kazakhstan, Armenia, and Slovakia), and 3) the bottom five countries with GGGI < 0.6. Five frames (equal weights and one for each dimension with a relatively higher weight) are selected from the local tour animation to show in Figure 8 and the full animation is available at https://vimeo.com/847874016.

Figure 8 helps to understand the how different dimension reduction alternatives affect the whole index distribution and individual countries.

- Bangladesh
- when decreasing the weight on politics, difference between countries narrower.
- increase politics -> all the country has lower index value

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
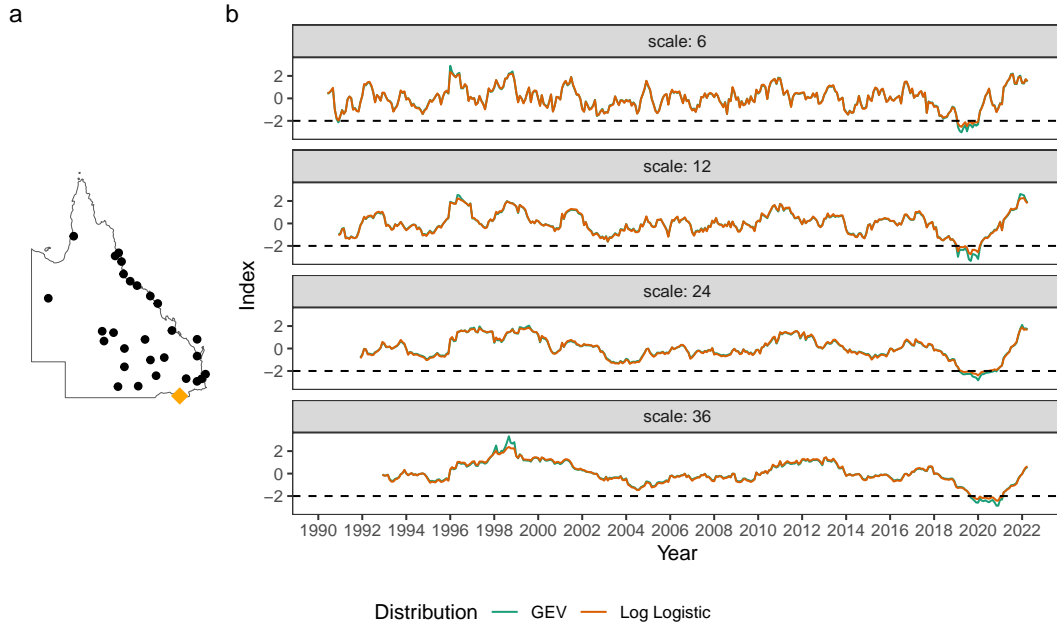
Figure 6. Time series plot of Standardized Precipitation-Evapotranspiration Index (SPEI) at the Texas post office station (highlighted by a diamond shape in panel a). The SPEI is calculated at four time scales (6, 12, 24, and 36 months) and fitted with two distributions (Log Logistic and GEV). The dashed 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.
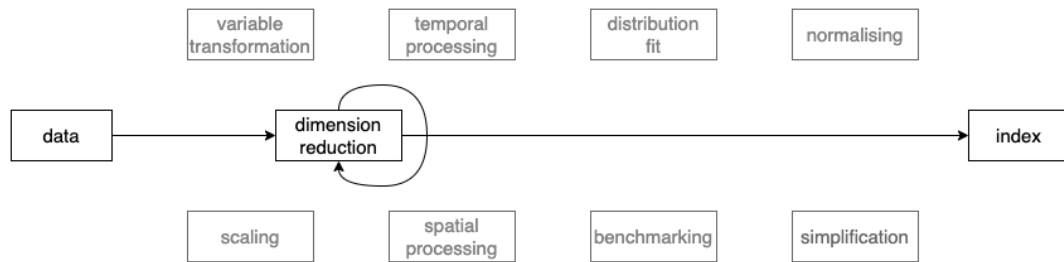

Figure 7. Diagram of pipeline steps for index construction. will need to be updated with better design and the distribution fitting step.
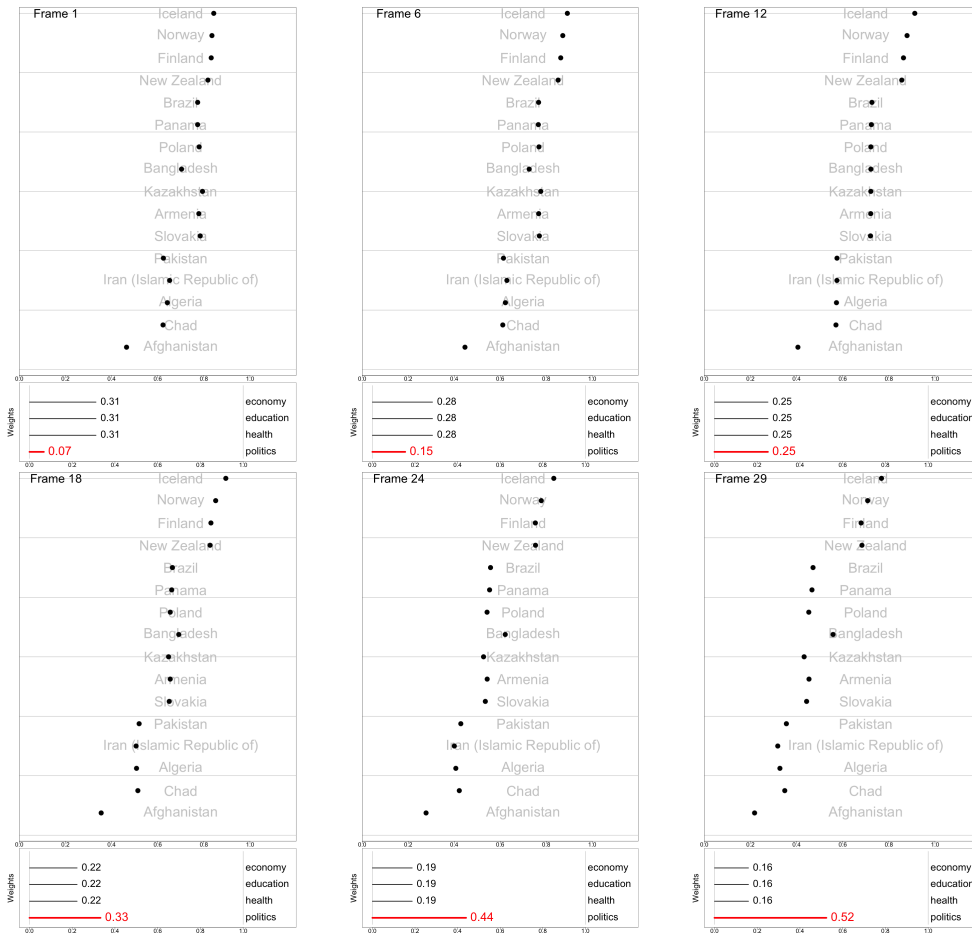
Figure 8. Five frames selected from varying the linear weights of four dimensions in Global Gender Gap Index. The weights vary slightly from the official simple average weights (0.25, 0.25, 0.25, 0.25) to observe how the index and ranking reponse. Full animation is available at https://vimeo.com/847874016.

15

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.

Kuhn, Max, and Hadley Wickham. 2020. *Tidymodels: A Collection of Packages for Modeling and Machine Learning Using Tidyverse Principles.* https://www.tidymodels.org.

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.

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

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the Tidyverse." *Journal of Open Source Software* 4 (43): 1686. https://doi.org/10.21105/joss.01686.

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.