Calculating standardised indices using SEI

Sam Allen
University of Bern
Oeschger Centre for
Climate Change Research

Noelia Otero
University of Bern
Oeschger Centre for
Climate Change Research

Abstract

Standardised indices are measurements of variables that have been converted to a standardised scale. The standardised scale facilitates comparisons between different variables, and its probabilistic interpretation means the indices are effective for risk management and decision making. Standardised indices have become popular in weather and climate settings, for example within operational drought monitoring systems, and have also been applied in other contexts, such as to energy variables. To facilitate their implementation in practice, the **SEI** package in R allows the flexible calculation of standardised indices. This paper discusses the theory underlying standardised indices, outlines the general framework to construct them, and provides implementation details for the **SEI** package.

Keywords: energy drought, renewable energy production, residual load, standardised indices, R.

1. Introduction

It is common to monitor the value of a variable over time: climate scientists monitor several weather and climate variables in the hope of identifying trends in the atmosphere; energy suppliers monitor energy demand and production, with particular focus recently on production from renewable sources; financial traders monitor financial indices to assess the behaviour of the economy, and so on. Doing so allows practitioners to not only identify long term trends in the variables of interest, but also to analyse abnormal situations with potentially adverse socio-economic consequences. These impacts can often be mitigated by techniques that permit an efficient monitoring of variables over time.

However, it is often difficult to compare time series of variables defined on different scales. For example, different spatial regions have different climates, and hence a rainfall event that is impactful at one location may not be at another. Similarly, energy demand will depend on the local climate, as well as on installed energy capacities. To account for this, it is convenient to transform variables onto the same, standardised scale. This standardisation can be achieved easily by framing the observed values in terms of their similarity to past values. The resulting standardised variables are often referred to as *standardised indices*.

Well-known examples of standardised indices include the Standardised Precipitation Index (SPI; McKee *et al.* 1993) and Standardised Precipitation Evapotranspiration Index (SPEI; Vicente-Serrano *et al.* 2010), which are ubiquitously used to monitor hydrological droughts at national hydrological centres (Beguería *et al.* 2014). Several other weather and climate indices have similarly been proposed: Zscheischler *et al.* (2014) introduced an analogous Standardised Temperature Index, Li *et al.* (2021) introduced an index to monitor simultaneously hot and dry weather conditions, while Allen and Otero (2023) introduced analogues of the SPI and SPEI that can be used to monitor energy supply and demand.

There are several benefits to standardised indices. The indices are defined on a common scale, and are thus easy to interpret. This standardised scale also has an underlying probabilistic interpretation, making the indices ideal for risk management and decision making. Standardised indices therefore provide an appealing framework with which to define shortages, or droughts, in the variable of interest. Since the standardisation can be performed separately for different seasons and locations, these shortages can be defined in a relative sense, allowing their intensity in different climatic regions to readily be compared. As summarised by Zargar *et al.* (2011), standardised indices provide a "pragmatic way to assimilate large amounts of data into a quantitative information that can be used in applications such as drought forecasting, declaring drought levels, contingency planning and impact assessment."

However, despite the widespread implementation of these standardised indices, no software package exists that provides comprehensive functionality to apply these in practice. The SPI, SPEI, and STI can be calculated using the **SPI**, **SPEI**, and **STI** R packages, respectively, and the **SCI** package was introduced to calculate standardised climate indices. However, the functionality of the **SCI** package is very limited. A developmental package exists at `https://github.com/WillemMaetens/standaRdized`, which is much more comprehensive. Nonetheless, this package itself has several shortcomings that limit its use in practice, particularly when interest is not on climate indices. Firstly, the range of distributions available from which to calculate the indices is still limited to those commonly used for the SPI and SPEI; namely the gamma, three-parameter gamma, Weibull, three-parameter Weibull, generalised extreme value (GEV), and generalised logistic distributions. This package therefore cannot be used to calculate the STI or other indices corresponding to variables with an irregular or multimodal distribution, such as energy demand. Moreover, this package assumes that there is a large archive of data available from which the indices can be calculated, since only measurements used on the same day in previous years are used as reference data. This means it is not possible to define the indices in terms of recently observed values, and the code breaks down if only a few years of data is available.

In this vignette, we introduce the **SEI** R package, which seeks to address these deficiencies and provide a flexible software package with which to calculate standardised indices. The primary advantage of this package is that it is not designed solely for climate indices, as the examples listed above are, and therefore contains a much broader functionality, making it applicable in a wider range of applications. Nonetheless, some of the code used in this package has been influenced by this developmental **standaRdized** package.

The remainder of this vignette is organised as follows. Section 2 provides some theoretical background on standardised indices and describes the general formula to construct them. Section 3 demonstrates how the **SEI** package can be used to calculate standardised indices in practice. An application of the package is then presented in Section 4, whereby the **SEI** package is used to calculate standardised energy indices based on time series of renewable energy production. Section 5 concludes the paper and discusses possible extensions to the package.

2. How to construct standardised indices

There is no unique way to define standardised indices. However, several widely-adopted indices are constructed using the same, simple procedure. Using this procedure, standardised indices are straightforward to calculate, can be defined on any timescale, and behave in a relative sense, meaning they can readily be compared for different regions, variables, or points in time.

The general approach is to choose a univariate variable X that measures the quantity of interest, and then to estimate X's distribution: in the SPI, X represents the precipitation accumulation aggregated over the time scale of interest, which is assumed to follow a gamma distribution; in the SPEI, X is the difference between precipitation and evapotranspiration, which is assumed to follow a log-logistic distribution; in the standardised temperature index, X is the temperature (at a given point in time and space), which is assumed to follow a normal distribution. More complicated variables have also been considered. Li *et al.* (2021), for example, consider X to be the probability of a simultaneously hot and dry event, as derived from a copula analysis, while Allen and Otero (2023) derive indices corresponding to energy production and demand.

The distribution of X is typically estimated from an archive of realisations of X, labelled $X_1, \dots, X_n$. These could represent precipitation accumulations on different days, for example. It is common to make parametric assumptions about the distribution, as is done when constructing the SPI, SPEI, and STI. However, the choice of distribution has led to much debate about what assumptions are most appropriate, particularly in the context of more extreme events. In reality, the optimal distribution will likely depend on several factors, and we therefore recommend using more flexible density estimation methods, such as the empirical distribution or kernel density estimation, particularly when larger archives of data are available. In any case, the distributional assumptions must be verified and checked for all locations and time periods at which the index is calculated.

Let F denote the distribution function of X. A well-known result in statistics says that the *probability integral transform* (PIT), $F(X)$, will be standard uniformly distributed. This assumes that F is continuous, though suitable randomisation can be applied when F is discrete or discrete-continuous. Having estimated F from the archive $X_1, \dots, X_n$, this transformation therefore provides a convenient approach with which to standardised a new observation X^*.

This *PIT variable*, $F(X^*)$, itself provides a standardised index of sorts, and can be rescaled using $2F(X^*) - 1$ to get an index that is centred at zero, bounded between -1 and 1, and with values intrinsically linked to probabilities. However, it is more common to use the Gaussian quantile function to transform the PIT variable, resulting in an index that follows a standard normal distribution, i.e. $\Phi^{-1}(F(X^*))$, where Φ^{-1} is the quantile function of the standard normal distribution.

That is, for any variable of interest, the probability integral transform can be used to define three different types of standardised indices:

- *Normal indices*: $\Phi^{-1}(F(X^*))$
- *Probability indices*: $F(X^*)$
- *Bounded indices*: $2F(X^*) - 1$

	Exceedance			Fall below		
Index Type	Moderate	Severe	Extreme	Moderate	Severe	Extreme
Normal	1.28	1.64	1.96	-1.28	-1.64	-1.96
Probability	0.9	0.95	0.975	0.1	0.05	0.025
Bounded	1.8	1.9	1.95	-1.8	-1.9	-1.95

Table 1: Possible thresholds used to define droughts when using each of the three types of standardised indices.

All three types of indices can be calculated from observations of the variable X and an estimate of its distribution F. In practice, it is most common to use normal indices, though one could argue that probability and bounded indices have a more direct probabilistic interpretation. For example, if we observe the 90^{th} percentile of X, then (assuming F is correctly estimated) the corresponding normal index will be 1.28, the 90^{th} percentile of the standard normal distribution, whereas the probability index would be 0.9, and the bounded index would be 0.8. Similarly, for the 10^{th} percentile of X, the normal index will be -1.28, the probability index would be 0.1, and the bounded index would be -0.8. For the median of X, the normal and bounded indices will be zero, whereas the probability index would be 0.5.

The probabilistic interpretation of standardised indices has made them a useful tool when analysing shortages and excesses of the variable X. This was first proposed by McKee *et al.* (1993) in the context of hydrometeorological droughts, and Allen and Otero (2023) demonstrate that a similar framework can be employed to define droughts in energy systems. In this case, droughts are defined in terms of thresholds of the standardised indices. Different classes of droughts can be defined using different thresholds, with a more severe drought corresponding to a more extreme threshold.

Following McKee *et al.* (1993), it is most common to employ three categories, typically labelled "Moderate", "Severe", and "Extreme" droughts. These often correspond to the thresholds 1, 1.5, and 2 of the standardised indices, respectively, though the thresholds 1.28, 1.64, and 1.96 could also be used, which correspond to quantiles of the standard normal distribution. This assumes that an exceedance of the thresholds constitutes a drought, though negatives of these values could analogously be used when interest is on values that fall below a threshold. Table 1 gives corresponding threshold values for the other two types of indices.

The **SEI** package allows the computation of all three types of droughts, and additionally allows drought occurrences and characteristics to be computed from a time series of standardised indices.

3. Package functionality

3.1. Standardised indices

The **SEI** package is designed to facilitate the computation of standardised indices. The principal function in the package is `std_index()`, which takes a time series or vector `x_new` as an input, and returns a time series or vector of corresponding standardised index values.

```
std_index(
  x_new,
  x_ref = x_new,
  timescale = NULL,
  dist = "empirical",
  return_fit = FALSE,
  moving_window = NULL,
  window_scale = NULL,
  agg_period = NULL,
  agg_scale = NULL,
  agg_fun = "sum",
  rescale = NULL,
  rescale_fun = "sum",
  index_type = "normal",
  ignore_na = FALSE
)
```

The input `x_new` must be either a vector or an **xts** time series object. The argument `x_ref` is similarly a vector or time series, containing the archive $X_1, \dots, X_n$ of past observations that are used to estimate the distribution F. This distribution is then applied to each value in `x_new`, which therefore correspond to values of X^* in the previous section. The default is that `x_ref = x_new`, in which case the standardised indices are calculated *in-sample*.

The `dist` argument specifies how the distribution F will be estimated from `x_ref`. Several parametric distributions are available, as well as non- and semi-parametric methods that are more flexible but require more data. A list of the distributions available in `std_index()` is presented in Table 2. The default is to use the empirical distribution of `x_ref`.

Table 2 additionally contains the support of the available distributions. The support of the empirical distribution is determined by the archive of observations $X_1, \dots, X_n$. To accurately estimate F, the empirical distribution requires a sufficiently large archive. A warning is therefore returned if the length of `x_ref` is smaller than 100, in which case the resolution of the index will be larger than 0.01.

Since we want to obtain a good estimate of F, the choice of distribution should be chosen to correspond with the variable of interest. For example, if interest is on precipitation, then we should not choose a distribution that assigns positive probability density to negative values, such as the normal distribution. Conversely, if interest is on temperature (in celcius), then we should not choose a distribution that has support $[0, \infty)$, such as the exponential distribution. An error message is returned if this is the case. The argument `return_fit` is a logical that specifies whether or not to return information about the model fit, including estimated parameters and goodness-of-fit statistics; the default is `return_fit = FALSE`.

The distribution fitting is performed using `fit_dist()`, which is essentially a wrapper for `fitdist()` in the **fitdistrplus** package, along with extensions that are relevant for the construction of standardised indices. The `fit_dist()` function is also exported by **SEI**, and further details can therefore be found on the associated help pages.

The argument `index_type` to `std_index()` specifies which type of standardised index should be used. This must be one of `"normal"` (the default), `"probability"`, or `"bounded"`, corresponding to the three definitions in the previous section.

Distribution	Argument code	Support
Empirical	`"empirical"`	range$(X_1, \dots, X_n)$
Kernel density estimation	`"kde"`	$\mathbb{R}$
Normal	`"norm"`	$\mathbb{R}$
Log-normal	`"lnorm"`	$[0, \infty)$
Logistic	`"logis"`	$\mathbb{R}$
Log-logistic	`"llogis"`	$[0, \infty)$
Exponential	`"exp"`	$[0, \infty)$
Gamma	`"gamma"`	$[0, \infty)$
Weibull	`"weibull"`	$[0, \infty)$

Table 2: Distributions available in **SEI**, and their supports.

The remaining arguments are only applicable when `x_new` and `x_ref` are time series, rather than vectors, since they all involve manipulations based on the date of the data. For example, the argument `moving_window` can be used to calculate standardised indices using a moving window. In this case, `x_ref` is updated for each observation in `x_new` so that it contains the previous `moving_window` (an integer) observations in the time series; the indices are therefore determined relative to recently observed values. The argument `window_scale` specifies the timescale that `moving_window` refers to. For example, setting `moving_window = 10` and `window_scale = "days"` would calculate standardised indices using a moving window of length 10 days.

If `moving_window` is specified but `window_scale = NULL`, then it is assumed that the units of `moving_window` correspond to the timescale of `x_new`. If the time difference between consecutive observations is consistent throughout the time series, then this can be determined automatically within `std_index()`. Alternatively, and more robustly, it can be specified by the user using the `timescale` argument. This (and `window_scale`) must be one of `"hours"`, `"days"`, `"weeks"`, `"months"`, `"quarters"`, or `"years"`.

If the original time series contains, say, hourly data, but the time series of standardised indices is desired on a different timescale, then the `rescale` argument can be used to rescale the data. This must also be one of the timescales listed above. By default, `std_index()` assumes the sum of the values should be returned when rescaling. For example, if interest is on precipitation accumulations, then converting hourly data to daily data will return the daily aggregations. However, alternative rescaling could also be performed using the `rescale_fun` argument, which is a function specifying what operation to perform over the aggregated data. If `rescale_fun = "mean"` or `rescale_fun = "max"`, for example, then the daily (or weekly, monthly, etc) mean or maximum would be provided, respectively, in place of the hourly data.

Similarly, the `agg_period` argument can be used to aggregate data across multiple time steps. Like `moving_window` and `window_scale`, `agg_period` is an integer representing the number of previous time steps over which the data should be aggregated, while `agg_scale` specifies the timescale of the aggregation period, which is by default assumed to be the timescale of the data. The argument agg_fun represents the function used to aggregate the data over the aggregation period.

This differs from rescale in the following way. If we want to convert a time series of daily observations to a time series of weekly observations, then we can use rescale = "weeks" (and timescale = "days"). On the other hand, if we set agg_period = 1 and agg_scale = "weeks", then we get a daily time series of weekly aggregated data.

To perform the rescaling, std_index() calls one of the **xts** functions apply.daily, apply.weekly, apply.monthly, apply.quarterly, and apply.yearly, alongside the argument rescale_fun. To perform the aggregation, std_index() uses the function aggregate_xts(), which is additionally exported by **SEI**.

```
aggregate_xts(
  x,
  len,
  scale = c("days", "hours", "weeks", "quarters", "years"),
  fun = "sum",
  timescale = c("days", "hours", "weeks", "quarters", "years"),
  na_thres = 10
)
```

The interpretation of the arguments is equivalent to in the discussion above. Further information can be found on the corresponding help pages.

3.2. Plotting indices

To visualise the indices, **SEI** includes the plot_sei() function.

```
plot_sei(
  x,
  type = c("ts", "hist"),
  title = NULL,
  lab = "Std. Index",
  ylims = NULL,
  n_bins = 30
)
```

The argument x is either a vector or an **xts** time series object that contains the index values to be displayed. This function can either be used to plot a time series of the values (type = "ts"), or a histogram (type = "hist"). In the latter case, the function is essentially a wrapper of geom_histogram() in **ggplot2**. Here, n_bins can be used to specify the number of bins in the histogram.

Additional aspects of the plot can be specified using title, lab, and ylims. For the time series, lab refers to the label of the y-axis, whereas it corresponds to the x-axis label of the histogram.

4. Application

4.1. Renewable energy production

Consider an application of standardised indices to renewable energy production in Europe. Hourly time series of wind and solar power generation is available for 27 European countries. This data is publicly available at `https://researchdata.reading.ac.uk/275/` (see Bloomfield *et al.* 2020, for details), and a subset of this data set corresponding to 2019 production can also be accessed from **SEI** using

```
data("data_supply", package = "SEI")
head(data_supply)

##                    date country    PWS
## 1 2019-01-01 00:00:00 Austria 0.0782
## 2 2019-01-01 01:00:00 Austria 0.0533
## 3 2019-01-01 02:00:00 Austria 0.0677
## 4 2019-01-01 03:00:00 Austria 0.0997
## 5 2019-01-01 04:00:00 Austria 0.1868
## 6 2019-01-01 05:00:00 Austria 0.2900
```

The data frame `data_supply` contains a `POSIXct` time series of dates, along with the corresponding country and wind and solar power production (`PWS`). The units of the production are Gigawatt hours (GWh).

The resources available for a country to generate wind and solar power can be quantified using its corresponding installed wind and solar capacities. The installed wind and solar capacities for the 27 countries (in 2017) are also available from **SEI**.

```
data("data_ic2017", package = "SEI")
head(data_ic2017)

## # A tibble: 6 x 3
## # Groups:   country [6]
##   country        IC17_wind IC17_solar
##   <chr>              <dbl>      <dbl>
## 1 Austria              2.4      1.03
## 2 Belgium              2.2      2.95
## 3 Bulgaria             0.6      1.05
## 4 Croatia              0.3      0.051
## 5 Czech_Republic       0.3      2.03
## 6 Denmark              4.8      0.601
```

We restrict attention here to renewable energy production in Germany.

```
de_supply_h <- subset(data_supply, country == "Germany")
de_supply_h <- xts::xts(de_supply_h$PWS, de_supply_h$date) # convert to xts
```

This can be rescaled from hourly to daily or weekly time series using **xts** functionality.

```
lab <- "Renewable Energy Production (GWh)"
plot_h <- plot_sei(de_supply_h, lab = lab, title = "Hourly")
plot_d <- plot_sei(de_supply_d, lab = lab, title = "Daily")
plot_w <- plot_sei(de_supply_w, lab = lab, title = "Weekly")
grid.arrange(plot_h, plot_d, plot_w, nrow = 1)
```

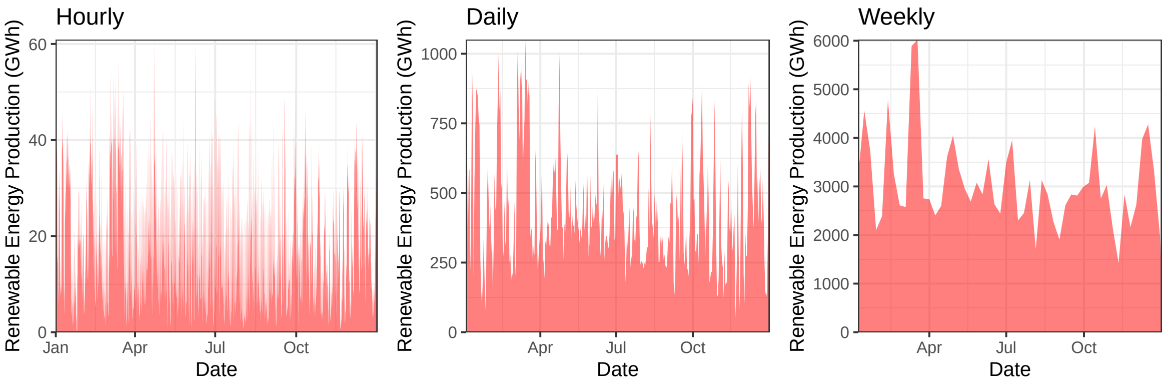

Figure 1: Time series of 2019 renewable energy production in Germany at hourly, daily, and weekly time scales.

```
de_supply_d <- xts::apply.daily(de_supply_h, "sum")     # daily data
de_supply_w <- xts::apply.weekly(de_supply_h, "sum")    # weekly data
```

These time series of renewable energy production in Germany can be visualised using the `plot_sei()` function.

All three time series display the same patterns, though the weekly time series removes the hourly and daily fluctuations.

The raw renewable energy production values can be transformed to a standardised index using `std_index()`. Following, Allen and Otero (2023), we refer to this as the standardised renewable energy production index (SREPI).

```
srepi_h <- std_index(de_supply_h)
srepi_d <- std_index(de_supply_d)
srepi_w <- std_index(de_supply_w)

## Warning in check_distribution(inputs):  using the empirical distribution is
only recommended when at least
##                  100 values are available when fitting the distribution
```

In this case, the reference data `x_ref` is assumed to be the input time series itself, meaning the indices are calculated relative to this input data. By default, `std_index()` returns normal indices (rather than probability or bounded indices), and estimates the distribution of the renewable energy production using the empirical distribution. This returns a warning for the weekly data, since there are fewer than 100 weekly production values in the time series,

```
lab <- "SREPI"
plot_h <- plot_sei(srepi_h, lab = lab, title = "Hourly", )
plot_d <- plot_sei(srepi_d, lab = lab, title = "Daily")
plot_w <- plot_sei(srepi_w, lab = lab, title = "Weekly")
grid.arrange(plot_h, plot_d, plot_w, nrow = 1)
```

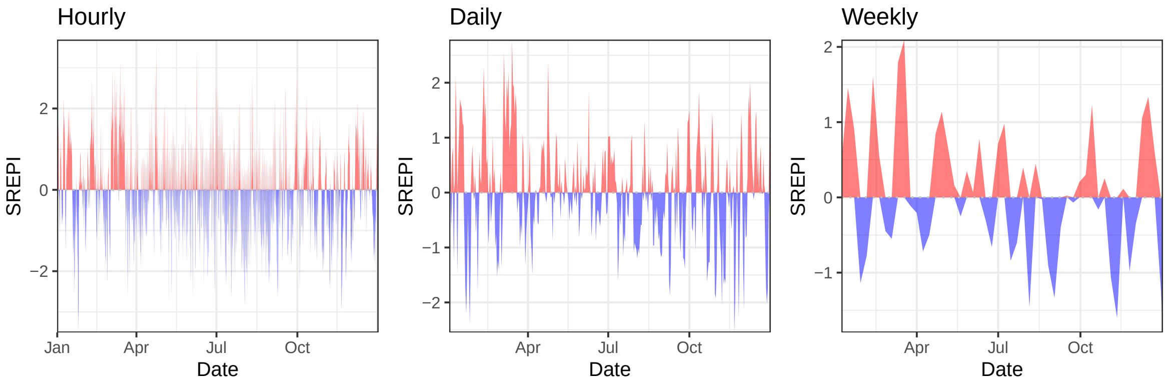

Figure 2: Time series of 2019 SREPI values in Germany at hourly, daily, and weekly time scales.

whereas the empirical distribution function is only recommended if at least 100 reference values are available. Instead, the weekly indices could be calculated using kernel density estimation, by specifying `dist = "kde"` in `std_index()`.

While these indices are created by standardising the respective hourly, daily, and weekly time series of raw production values, they could all be obtained from the original hourly time series by specifying the `rescale` argument in `std_index()`.

```
z <- std_index(de_supply_h, rescale = "days")
all.equal(srepi_d, z)

## [1] TRUE
```

By default, the `plot_sei()` function displays the time series of input values. However, by specifying `type = "hist"` (rather than the default `type = "ts"`), this function can additionally be used to plot a histogram of the input values. An example of this for the daily data production and corresponding SREPI values is presented in Figure 3.

While the raw renewable energy production follows a heavily skewed distribution, the SREPI values, by construction, resemble a sample from a standard normal distribution. Note that this is only the case for the "normal indices", if probability or bounded indices are used, then the resulting histograms should be flat. This is illustrated in Figure 4.

Finally, the **SEI** package additionally allows shortages or drought characteristics of the time series to be assessed. In this case, an energy supply drought could occur if the renewable energy production is low compared to previously observed values (Allen and Otero 2023). This corresponds to a low SREPI.

As in Table 1, we define three categories of droughts, each defined using increasing thresholds

```
plot_raw <- plot_sei(de_supply_d, type = "hist",
                     lab = "Renewable Energy Production (GWh)")
plot_ind <- plot_sei(srepi_d, type = "hist", lab = "SREPI")
grid.arrange(plot_raw, plot_ind, nrow = 1)
```

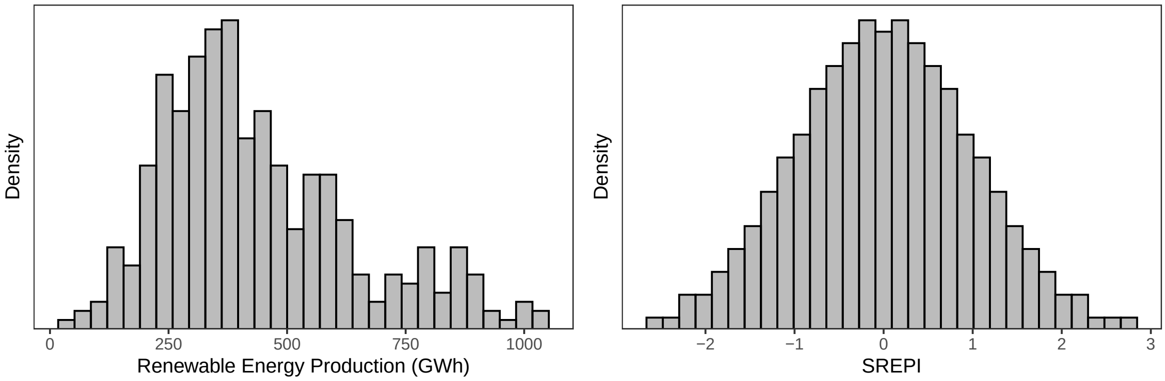

Figure 3: Histogram of 2019 daily renewable energy production and SREPI values in Germany.

```
srepi_d_prob <- std_index(de_supply_d, index_type = "probability")
plot_prob <- plot_sei(srepi_d_prob, type = "hist", lab = "SREPI",
                      title = "Probability")
srepi_d_bnd <- std_index(de_supply_d, index_type = "bounded")
plot_bnd <- plot_sei(srepi_d_prob, type = "hist", lab = "SREPI",
                     title = "Bounded")
grid.arrange(plot_prob, plot_bnd, nrow = 1)
```

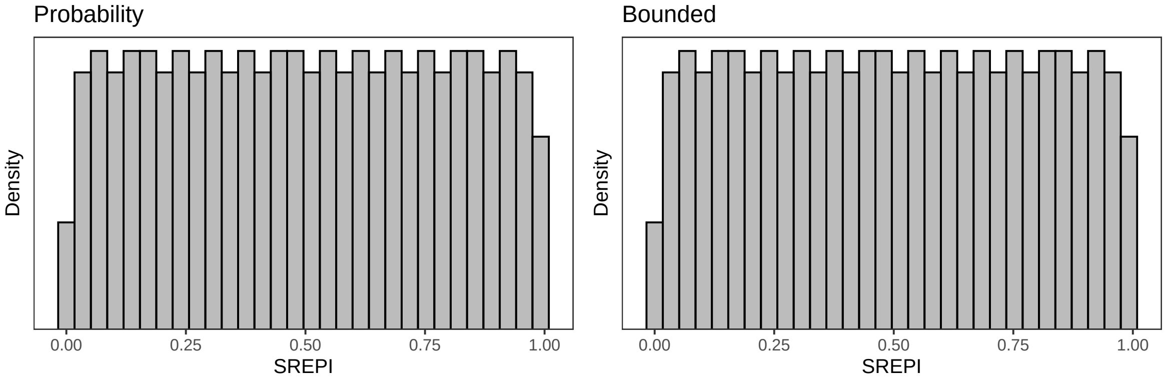

Figure 4: Histogram of 2019 daily renewable energy production and SREPI values in Germany, where the indices are probability and bounded indices.

of the SREPI. The function `get_drought()` can then be used to obtain a time series of drought occurrences, intensities, durations, and magnitudes.

The output `drought_df` is a dataframe containing the original time series, as well as four additional columns: the intensity (`ins`), which corresponds to the category of drought, a higher value referring to a more severe drought category; the occurrence (`occ`), which corresponds to the intensity being higher than zero; the duration (`dur`), which provides the number of consecutive time steps that are in a drought state; and the drought magnitude (`mag`), which is the sum of all indices within the drought event. Note that if only one threshold were used to define a drought, rather than the three used here, then the intensity is equivalent to the occurrence, and is therefore not returned.

From `drought_df`, it is straightforward to analyse the properties of these drought events. For example, we can list the frequency that a moderate, severe, or extreme drought event occurs,

```r
num_ev <- sapply(0:3, function(i) sum(drought_df$ins == i))
names(num_ev) <- c("None", "Moderate", "Severe", "Extreme")
print(num_ev)

##     None Moderate   Severe  Extreme
##      330       18        9        8
```

we could display the relative frequency of drought durations,

```r
table(drought_df$dur[drought_df$dur > 0])

##
##  1  2  3
## 11  3  6
```

or we could calculate the average drought magnitude

```r
mean(drought_df$mag[drought_df$mag != 0])

## [1] 2.99
```

These are just simple examples of analyses that could be performed using the output of `get_drought()`. For a more thorough analysis, readers are referred to Allen and Otero (2023).

5. Summary

This vignette documents the **SEI** package in R, which provides comprehensive functionality to calculate standardised indices. Standardised indices are projections of variables onto an interpretable and probabilistically meaningful scale, and they have become popular tools when monitoring variables of interest over time. The **SEI** package converts a time series (or vector) of observations to a time series (or vector) of standardised indices. The package allows the time series to be aggregated and rescaled to different time scales, and plot capabilities are also available to visualise the indices over time.

The package additionally allows users to define and analyse shortages, or droughts, of the variable under consideration. An example is presented in Section 4 whereby the package is employed to calculate standardised indices to monitor renewable energy production droughts. However, the package is applicable in much larger generality. We outline the general framework for constructing standardised indices, and reiterate that this approach can be applied to any variable of interest, not just those previously considered in the literature. With this in mind, the **SEI** offers an extensive collection of distributions with which to calculate the indices, including flexible non-parametric methods, making it applicable in a variety of settings. This is in contrast to existing software packages for standardised indices, which restrict attention to the SPI and SPEI.

Acknowledgements

This package was created under funding by the Oeschger Centre for Climate Change Research and the Swiss Federal Office of Meteorology and Climatology (MeteoSwiss).

References

Allen S, Otero N (2023). "Standardised Indices to Monitor Energy Droughts." *Renewable Energy*, **217**, 119206. doi:10.1016/j.renene.2023.119206.

Beguería S, Vicente-Serrano SM, Reig F, Latorre B (2014). "Standardized Precipitation Evapotranspiration Index (SPEI) Revisited: Parameter Fitting, Evapotranspiration Models, Tools, Datasets and Drought Monitoring." *International Journal of Climatology*, **34**, 3001–3023. doi:10.1002/joc.3887.

Bloomfield H, Brayshaw D, Charlton-Perez A (2020). "ERA5 Derived Time Series of European Country-Aggregate Electricity Demand, Wind Power Generation and Solar Power Generation: Hourly Data From 1979-2019." doi:10.17864/1947.272.

Li J, Wang Z, Wu X, Zscheischler J, Guo S, Chen X (2021). "A Standardized Index for Assessing Sub-Monthly Compound Dry and Hot Conditions With Application in China." *Hydrology and Earth System Sciences*, **25**, 1587–1601. doi:10.5194/hess-25-1587-2021.

McKee TB, Doesken NJ, Kleist J (1993). "The Relationship of Drought Frequency and Duration to Time Scales." In *Proceedings of the 8th Conference on Applied Climatology*, volume 17, pp. 179–183. Boston, MA, USA.

Vicente-Serrano SM, Beguería S, López-Moreno JI (2010). "A Multiscalar Drought Index Sensitive to Global Warming: The Standardized Precipitation Evapotranspiration Index." *Journal of Climate*, **23**, 1696–1718. doi:10.1175/2009JCLI2909.1.

Zargar A, Sadiq R, Naser B, Khan FI (2011). "A Review of Drought Indices." *Environmental Reviews*, **19**, 333–349. doi:10.1139/a11-013.

Zscheischler J, Michalak AM, Schwalm C, Mahecha MD, Huntzinger DN, Reichstein M, Berthier G, Ciais P, Cook RB, El-Masri B, *et al.* (2014). "Impact of Large-Scale Climate Extremes on Biospheric Carbon Fluxes: An Intercomparison Based on MsTMIP Data." *Global Biogeochemical Cycles*, **28**, 585–600. doi:10.1002/2014GB004826.

Affiliation:

Sam Allen
University of Bern
Institute of Mathematical Statistics and Actuarial Science
Alpeneggstrasse 22
3012 Bern, Switzerland
E-Mail: sam.allen@unibe.ch
and
Oeschger Centre for Climate Change Research

Noelia Otero
University of Bern
Institute of Geography
Hallerstrasse 12
3012 Bern, Switzerland
E-Mail: noelia.otero@unibe.ch
and
Oeschger Centre for Climate Change Research