

# Calculating Standardised Indices Using SEI

Sam Allen  
ETH Zürich

Noelia Otero  
University of Bern  
Oeschger Centre for  
Climate Change Research

---

## Abstract

Standardised indices are measurements of variables on 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 vignette discusses the theory underlying well-known standardised indices, outlines the general framework to construct them, and provides implementation details for the **SEI** package. Two case studies are presented whereby standardised indices are applied to climate and energy variables.

*Keywords:* standardised indices, drought monitoring, energy droughts, R.

---

## 1. Introduction

Monitoring the values of a variable over time allows practitioners to not only identify long term patterns in the variables of interest, but also to analyse abnormal situations with potentially adverse socio-economic consequences. However, it can be difficult to compare time series of variables defined on different scales. For example, different spatial regions have different climates, meaning 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. 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 hydrometeorological droughts (Beguería *et al.* 2014). The framework used to construct the SPI and SPEI is simple and interpretable, framing the observed values in terms of their similarity to past values. As such, while other methods of standardisation exist, this framework has also been used to construct many other weather and climate indices.

This includes, for example, a Standardised Runoff Index (Shukla and Wood 2008), a Standardised Streamflow Index (Vicente-Serrano *et al.* 2012), a Standardised Groundwater Index (Bloomfield and Marchant 2013), and a Standardised Temperature Index (Zscheischler *et al.*

2014). Multivariate extensions of these standardised indices have also been proposed (Erhardt and Czado 2018). Hao *et al.* (2019), for example, define a Standardised Compound Event Index, which is used by Li *et al.* (2021) to construct a Standardised Compound Drought and Heat Index. Outside of a weather and climate context, Allen and Otero (2023) introduced analogues of the SPI and SPEI that can be used to monitor energy supply and demand. This general framework for constructing standardised indices can be applied to any variable of interest, not just those previously considered in the literature.

There are several benefits to these standardised indices. The indices are easy to interpret, and are defined on a scale with a probabilistic interpretation, making them useful tools 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 in different conditions, for example in different seasons and locations, these shortages can be defined in a relative sense, allowing their intensity in the different conditions to 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.”

As a result, several software packages have been developed to apply standardised indices in practice. The **SCI** package (Gudmundsson and Stagge 2016) in the programming language R (R Core Team 2017) provides the functionality to calculate several standardised climate indices. The **SPEI** package (Beguiría and Vicente-Serrano 2023), also in R, contains more comprehensive functionality for the SPI and SPEI, and the **standard-precip** package (Nussbaumer 2021) has transferred this functionality to the programming language Python (Python 2017). The **spei** package (Vonk 2017) in Python additionally allows computation of the SPI and SPEI, as well as the Standardised Runoff and Groundwater Indices, while the **climate-indices** package in Python also provides functionality to compute the SPI, SPEI, and other drought indices. The developmental package **standaRdized** (Maetens 2019) can also be used for this purpose, while the **SPIGA** package (Ayala-Bizarro and Zúñiga-Mendoza 2016) in R allows the SPI to be calculated using genetic algorithms.

However, existing software packages are limited in their flexibility, with functionality typically restricted to particular standardised indices (generally the SPI and SPEI), and particular assumptions about the reference data or parametric distributions used to calculate the indices (see next section for details). In this vignette, we introduce the **SEI** R package, which seeks to assimilate the advantages of these various packages to provide a flexible and comprehensive software package with which to calculate standardised indices. In particular, the package is not designed solely for climate indices, and is therefore applicable in much broader generality. The package can handle variables defined on any time scale and with any (possibly non-stationary) underlying distribution, permits user-specified reference data when calculating the indices, and additionally provides plot capabilities to visualise the resulting standardised indices.

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 discusses diagnostic checks that confirm the standardisation was appropriate. Section 4 demonstrates how the **SEI** package can be used to calculate standardised indices in practice. Two applications are then presented in Section 5, whereby the **SEI** package is used to calculate standardised energy indices based on time series of renewable energy production,

as well as standardised wind speed indices. Section 6 concludes the vignette 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 variables, spatial regions, 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 usually assumed to follow a gamma distribution; in the SPEI,  $X$  represents the difference between precipitation and evapotranspiration, which is usually assumed to follow a log-logistic distribution; in the standardised temperature index,  $X$  represents the temperature, which is usually assumed to follow a normal distribution. More complicated variables have also been considered. Hao *et al.* (2019), for example, consider  $X$  to be the probability of a simultaneously hot and dry event, as derived from a copula analysis.

The distribution of  $X$  is typically estimated from an archive of realisations,  $x_1, \dots, x_n$ . These could represent precipitation accumulations in different months, for example, or energy demand on different days. It is common to make parametric assumptions about the distribution, as in the examples above, though non- and semi-parametric density estimation methods could also be employed. The choice of distribution is discussed in detail in the following section. In any case, the distributional assumptions must be verified when the index is calculated.

Let  $F$  denote the distribution function of  $X$ , and let  $\hat{F}$  denote an estimate of  $F$  obtained from  $x_1, \dots, x_n$ . If  $F$  is continuous, then the *probability integral transform* (PIT)  $F(X)$  will be standard uniformly distributed. Hence, this transformation provides a convenient approach with which to standardise a (possibly new) random observation  $X^*$  (with realisation denoted  $x^*$ ): the PIT value  $\hat{F}(x^*)$  denotes the relative position of the observation  $x^*$  within the distribution  $\hat{F}$ . If  $x^*$  is large relative to  $\hat{F}$ , then the PIT value will be close to one; if  $x^*$  is small relative to  $\hat{F}$ , then the PIT value will be close to zero.

This *PIT variable*  $\hat{F}(X^*)$  is bounded between 0 and 1, with values intrinsically linked to probabilities. This therefore itself provides a standardised index of sorts. This PIT variable can also be rescaled using  $2\hat{F}(X^*) - 1$  to get an index that is centred at zero and bounded between -1 and 1, though it is more common to use the Gaussian quantile function  $\Phi^{-1}$  to transform  $\hat{F}(X^*)$ . In this case, if  $X^*$  is identically distributed to  $X$ , and  $\hat{F}$  is a good approximation of  $F$ , then the standardised index will follow a 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:*  $\Phi^{-1}(\hat{F}(X^*))$
- *Probability indices:*  $\hat{F}(X^*)$
- *Bounded indices:*  $2\hat{F}(X^*) - 1$

All three types of indices can be calculated from observations of the variable  $X$  and an estimate of its distribution  $F$ . Since they are defined as increasing functions of PIT values, these standardised indices all quantify the relative position of an observation  $x^*$  within the previously observed sample  $x_1, \dots, x_n$ , making them easy to interpret.

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<sup>th</sup> percentile of  $X$ , then (assuming  $F$  is correctly estimated) the corresponding normal index will be 1.28, the 90<sup>th</sup> percentile of the standard normal distribution, whereas the probability index will be 0.9, and the bounded index will be 0.8. Similarly, for the 10<sup>th</sup> percentile of  $X$ , the normal index will be -1.28, the probability index will be 0.1, and the bounded index will be -0.8. For the median of  $X$ , the normal and bounded indices will be zero, whereas the probability index will be 0.5.

The general framework can therefore be summarised as follows.

1. Estimate  $F$  from the observations  $x_1, \dots, x_n$ .
2. Check that the estimate  $\hat{F}$  correctly fits the data.
3. Apply  $\hat{F}$  to an observation  $x^*$  (or several observations).
4. Transform  $\hat{F}(x^*)$  so that the index is on the desired scale.

The probabilistic interpretation of these 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 the same framework can be employed to define droughts in energy systems. In this case, droughts are defined when the standardised index exceeds or falls below a relevant threshold. 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 normal standardised indices, respectively, though the thresholds 1.28, 1.64, and 1.96 are also often used, which correspond to quantiles of the standard normal distribution. This assumes that a drought is defined as an exceedance of a threshold, though the negative 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 indices, and additionally allows drought occurrences and characteristics to be computed from a time series of standardised indices.

### 3. The choice of distribution

#### 3.1. Stationary distribution estimation

The construction of standardised indices relies on an estimate of the distribution function  $F$ . This has led to much debate about what distributional assumptions are most appropri-

Index Type	Fall below			Exceed		
	Extreme	Severe	Moderate	Moderate	Severe	Extreme
Normal	-1.96	-1.64	-1.28	1.28	1.64	1.96
Probability	0.025	0.05	0.1	0.9	0.95	0.975
Bounded	-0.95	-0.9	-0.8	0.8	0.9	0.95

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

ate. Consider the SPI, for example. [McKee \*et al.\* \(1993\)](#) initially introduced the SPI using the gamma distribution to model  $F$ . However, [Guttman \(1999\)](#) argued that the Pearson III distribution is more appropriate, due to the additional flexibility afforded by an extra parameter, while results in [Stagge \*et al.\* \(2015\)](#) suggest the Weibull distribution fits the monthly precipitation accumulations in their study better than alternative parametric choices. [Russo \*et al.\* \(2013\)](#) also consider using non-stationary distributions that include parameters that depend on time, while [Li \*et al.\* \(2015\)](#) incorporate additional covariates based on other climate variables. In reality, there is generally no true parametric distribution underlying the data, and the optimal assumptions will change depending on the data under consideration.

As an alternative, it has been argued (e.g. [Erhardt and Czado 2018](#); [Hao \*et al.\* 2019](#); [Allen and Otero 2023](#)) that the choice of a parametric distribution can be circumvented by using the empirical distribution function of the observations,

$$\hat{F}_n(x) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{x_i \leq x\}, \quad (1)$$

where  $\mathbb{1}\{\cdot\}$  is equal to one if the statement inside the brackets is true, and zero otherwise. Since  $\Phi^{-1}(0) = -\infty$  and  $\Phi^{-1}(1) = \infty$ , it is common to rescale  $\hat{F}_n(x)$  so that it is never equal to zero or one, thereby ensuring that the (normal) standardised indices are real-valued; for example, using  $(n\hat{F}_n(x) + 1)/(n + 2)$ . The empirical distribution function returns a standardised index that can only take on  $n + 1$  possible values. This is not an issue in practice if  $n$  is large, but can become problematic with smaller sample sizes, particularly when interest is on extreme events.

[Allen and Otero \(2023\)](#) propose the use of semi-parametric density estimation methods, which provide a compromise between parsimonious parametric distributions and the empirical distribution function. For example, kernel density estimators estimate the density function  $f$  corresponding to  $F$  using

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - x_i}{h}\right),$$

where  $K$  is a kernel function, and  $h > 0$  is a smoothing parameter called the bandwidth. The **SEI** package provides functionality to perform kernel density estimation using a Gaussian kernel. Non- and semi-parametric density estimation methods are considerably more flexible than parametric methods, though they require more data to accurately model  $F$ . Further details about density estimation methods can be found, for example, in [Silverman \(2018\)](#).

Distribution	Argument code	Support
Empirical	"empirical"	$\text{range}(x_1, \dots, x_n)$
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.

Table 2 lists the distributions available in the **SEI** package for estimating  $F$ . The choice of distribution should align 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 Celsius), then we should not choose a distribution that has support  $[0, \infty)$ , such as the exponential distribution. Table 2 additionally contains the support of the available distributions; the support of the empirical distribution is determined by the archive of observations  $x_1, \dots, x_n$ .

### 3.2. Non-stationary distribution estimation

In many practical applications, the variable of interest may be non-stationary. For example, several weather variables have exhibited strong temporal trends in recent decades as a result of changes in the climate. If these trends are ignored when estimating the distribution  $F$ , then the resulting index values may be misleading: if the variable is increasing in time, then larger index values will generally occur for later observations, meaning extreme events, as defined using the standardised indices, will be clustered in time.

This could be circumvented by estimating  $F$  using an adaptive subset of the observations  $x_1, \dots, x_n$  that accounts for these trends, rather than using all available data. For example,  $\hat{F}$  could be constructed from a moving window that only considers (or assigns higher weight to) the most recent observations, so that the standardised indices, as well as the corresponding drought and extreme events, are defined in terms of the recent climate. This would require estimating a separate distribution for each new observation  $x^*$ , but provides an example of how the data used to construct standardised indices affects their interpretation.

Alternatively, the trends could be accounted for by estimating  $F$  as a function of time, so that time is included as a predictor in the distribution (Russo *et al.* 2013). The distribution could similarly be made to depend on other predictor variables (Li *et al.* 2015). For example, precipitation accumulations are strongly influenced by changes in the temperature. By incor-

porating temperature as a predictor within the distribution of precipitation, the PIT values would quantify how extreme a precipitation observation is, given the corresponding temperature. The problem of estimating the distribution of the observations  $x_1, \dots, x_n$  then becomes estimating the conditional distribution of the observations given corresponding realisations of a set of predictor variables.

There are many ways to estimate this conditional distribution (see e.g. Kneib *et al.* 2023), but one flexible and general approach is the Generalized Additive Models for Location, Scale and Shape (GAMLSS) framework (Rigby and Stasinopoulos 2005). Several recent studies have proposed to construct standardised indices from non-stationary distributions estimated using GAMLSS (e.g. Wang *et al.* 2015; Rashid and Beecham 2019; Shao *et al.* 2022). The general approach is to choose a parametric distribution and then assume that its location, scale and shape parameters depend additively (but possibly non-linearly) on the set of predictor variables. GAMLSS is a well-known framework for which many resources are available, to which readers are referred for further details. The **SEI** package allows non-stationary distributions to be implemented to construct standardised indices by calling functions from the **gamlss** R package (Stasinopoulos and Rigby 2008; Stasinopoulos *et al.* 2017).

Incorporating predictors into standardised indices implicitly assumes that the impacts associated with the variable of interest are somehow mitigated by the predictors. For example, a society’s capability to deal with hydrometeorological droughts typically becomes better as these events become more common, so that an unusually low precipitation event today leads to a lower impact than it would have done 50 years ago, say; instead, a yet more extreme precipitation event is required to generate the same impact. Including time, or other relevant predictors, within the distribution  $F$  allows this information to be incorporated into the standardised indices. Conversely, if we cannot assume that the predictors help to mitigate the impacts associated with the variable being monitored, then there is little reason to include it as a predictor when constructing standardised indices.

### 3.3. Checking distribution fit

Regardless of the method used to estimate  $F$ , it is necessary to check that the estimated distribution fits the data. Several goodness-of-fit tests exist for this purpose. The **SEI** package implements the Kolmogorov-Smirnov test (Massey Jr 1951) applied to the PIT values  $\hat{F}(x_1), \dots, \hat{F}(x_n)$ . If the distribution fits the data, these PIT values should resemble a sample from a standard uniform distribution. The Kolmogorov-Smirnov test statistic then calculates the maximum difference between the empirical distribution function of the PIT values and the distribution function of the standard uniform distribution. If this statistic is close to zero, then it suggests that the hypothesised distribution fits the data well.

The fit of the distribution could also be visualised graphically by plotting the distribution of the PIT values and checking whether they resemble a sample from a standard uniform distribution; a histogram of these values should therefore be flat, up to sampling variation, for example. Similarly, the corresponding normal standardised indices should resemble a sample from a standard normal distribution. An example of how this can be checked using the **SEI** package is presented in Section 5.

## 4. Package functionality



The **SEI** package is designed to facilitate the computation of standardised indices. The package is available on CRAN ([CRAN.R-project.org/package=SEI](https://CRAN.R-project.org/package=SEI)) and a developmental version is available on GitHub ([github.com/noeliaof/SEI](https://github.com/noeliaof/SEI)). In this section, we demonstrate the functionality of the **SEI** package when implementing standardised indices in practice.

#### 4.1. 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,
  dist = "empirical",
  preds_new = NULL,
  preds_ref = preds_new,
  method = "mle",
  return_fit = FALSE,
  index_type = "normal",
  gr_new = NULL,
  gr_ref = gr_new,
  timescale = NULL,
  moving_window = NULL,
  window_scale = NULL,
  agg_period = NULL,
  agg_scale = NULL,
  agg_fun = "sum",
  rescale = NULL,
  rescale_fun = "sum",
  ignore_na = FALSE,
  n_thres = 10,
  na_thres = 10,
  lower = -Inf,
  upper = Inf,
  cens = index_type,
  ...
)
```

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 realisations of  $X^*$  in the previous section. The default is that `x_ref = x_new`, in which case the standardised indices are calculated *in-sample*.

As a very simple example, consider a vector of observations drawn from a gamma distribution. These can be converted to (normal) standardised indices as follows.



```
plot_raw <- plot_sei(x, type = "hist", title = "Raw")
plot_std <- plot_sei(x_std, type = "hist", title = "Standardised")
grid.arrange(plot_raw, plot_std, nrow = 1)
```

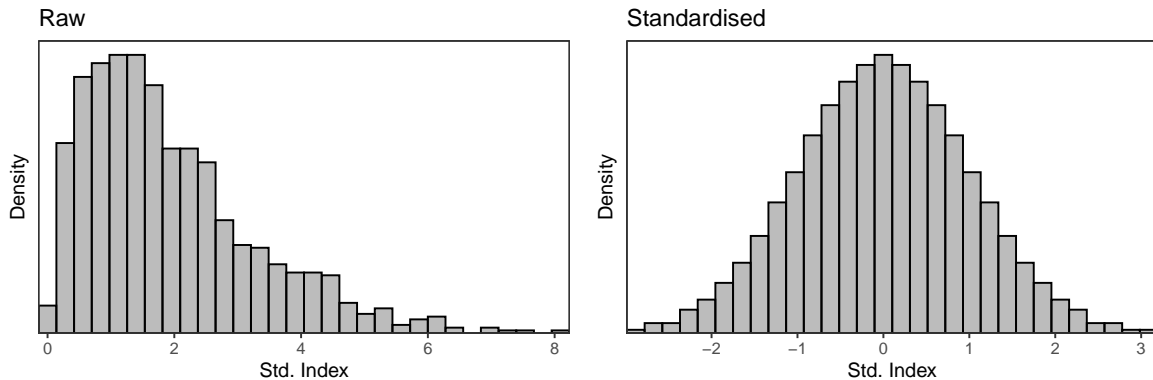


Figure 1: Histogram of values sampled from a gamma distribution, and corresponding standardised values.

```
x <- rgamma(1000, shape = 2, scale = 1)
x_std <- std_index(x)
```

Histograms of the raw and standardised values are displayed in Figure 1. While the raw values are heavily skewed, the standardised values closely resemble a standard normal distribution. The argument `index_type` in `std_index()` specifies which type of standardised index should be constructed from the estimated distribution. This must be one of "normal" (the default), "prob01", or "prob11", corresponding respectively to the definitions of normal, probability, and bounded standardised indices defined in the previous section.

The `dist` argument in `std_index()` specifies how the distribution  $F$  will be estimated from `x_ref`. The default is to use the empirical distribution of `x_ref` (Equation 1). To accurately estimate  $F$ , the empirical distribution requires a sufficiently large archive, and a warning is therefore returned if the length of `x_ref` is smaller than 100. An error message is returned if the distribution support does not align with the data; for example, if a distribution with support  $[0, \infty)$  is chosen while the data contains negative values.

Information about the distribution fit, including estimated parameters and goodness-of-fit statistics, can be returned by setting `return_fit = TRUE`; the default is `return_fit = FALSE`. If `return_fit = TRUE`, the output of `std_index()` is a list containing the time series of standardised indices (`si`), the estimated distribution parameters (`params`), and a named vector containing properties of the model fit (`fit`); this includes the number of observations used to estimate the distribution, the number and percentage of NA values in the data, the Akaike Information Criterion (AIC) value, and the p-value of the Kolmogorov-Smirnov test that the distribution correctly fits the data.

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 be found on the associated help page. Parameters of the distributions

(if any) are estimated by default using maximum likelihood estimation, though many other estimation techniques, including method of moments and method of L-moments, can be employed by changing the `method` argument. Other arguments to `fitdist()` can be specified as variable arguments `....`

## 4.2. Non-stationary distributions

The default is to estimate a non-stationary distribution from the data in `x_ref`. However, non-stationary distributions can be estimated via the GAMLSS framework by inputting a data frame of predictor variables in `preds_ref`. This data frame should have the same number of rows as the length of `x_ref`, with each column representing a separate predictor variable. Similarly, `preds_new` should be a data frame containing the same predictor variables as `preds_ref` (with the same column names), but with rows corresponding to the entries of `x_new`. The `std_index()` function then estimates the non-stationary distribution by calling `gamlss()` from the `gamlss` package. The model assumes that the location parameter of the distribution depends linearly on all predictors in `preds_ref`. Relationships between the predictors and other distribution parameters, as well as other optional arguments to `gamlss()`, can again be applied using `....`

For example, consider observations drawn from a normal distribution whose mean and standard deviation both increase over time.

```
N <- 1000
t <- seq(10, 20, length.out = N)
x <- rnorm(N, mean = t, sd = exp(t/10))
```

The underlying distribution of these values can be estimated both with and without including a trend in the location and scale parameters of the distribution. Note that `sigma.formula` is an optional argument to `gamlss()` that allows the scale parameter of the distribution to depend on predictors.

```
preds <- data.frame(t = t)
# standardised indices without trend
si_st <- std_index(x, dist = "norm")
# standardised indices with trend in mean
si_nst <- std_index(x, dist = "norm", preds_new = preds)
# standardised indices with trend in mean and sd
si_nst2 <- std_index(x, dist = "norm", preds_new = preds, sigma.formula = ~ t)
```

Figure 2 displays the time series of standardised indices corresponding to these values under three different assumptions: there is no trend, there is a linear trend in the mean but not the standard deviation of the distribution, and there is a trend in both the mean and standard deviation. When the trend is not included, higher index values unsurprisingly occur at higher time points, and the variation of the indices also increases in time. If a trend is included in the mean of the distribution, then the standardised indices do not increase in time, but still become more variable. If a trend is included in both the mean and the standard deviation, then the index values become stationary.

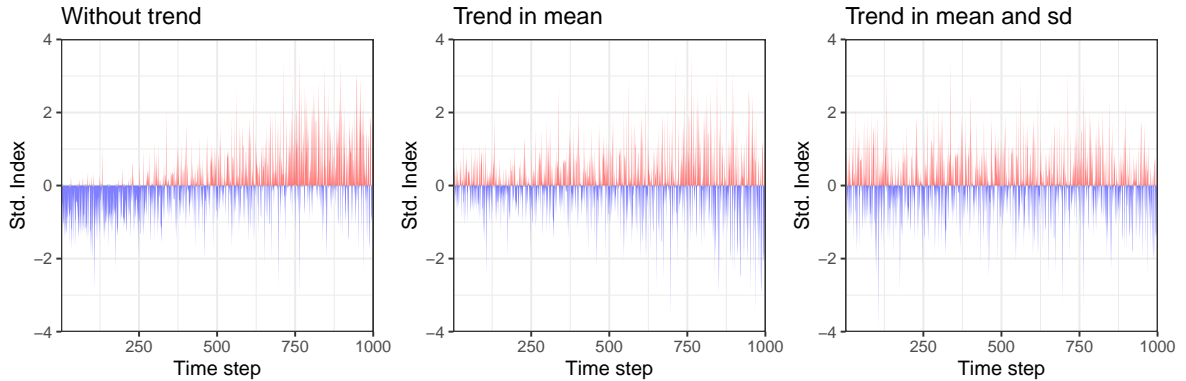


Figure 2: Standardised indices corresponding to a time series of values drawn from a non-stationary normal distribution. Results are shown when the estimated normal distribution does and does not include a trend.

### 4.3. Grouping

Rather than estimating one distribution across all observations, whether dependent on predictors or not, the variable of interest may have a distribution that changes in different subsets of the data. These subsets could correspond to different seasons, for example. Standardised indices corresponding to different distributions can be derived from `std_index()` via the `gr_new` and `gr_ref` arguments. These should be factor vectors of the same length as `x_new` and `x_ref` respectively, with each factor corresponding to a different subset of the data from which a distribution should be estimated. The values in `x_ref` are stratified according to `gr_ref`, and a separate distribution is estimated from each subset; the values in `x_new` are then stratified according to `gr_new`, and the distribution corresponding to each factor level is applied to the subset of values in `x_new` that correspond to the same level. The factor levels in `x_new` and `x_ref` should therefore be the same. In particular, `x_new` should not contain any levels that are not also present in `x_ref`, since in this case there would be no data in `x_ref` from which to estimate the relevant distribution. This returns an error.

Non- and semi-parametric distributions can flexibly adapt to situations where the shape of the distribution changes within the different subsets of the data. However, if a large number of factor levels are present in `x_ref` and `x_new`, then the amount of data to estimate the distributions can become small. To counteract this, the `dist` argument in `std_index()` can be a vector of the same length as the number of levels in `gr_ref`. The first element of `dist` denotes the distribution to be fit to the data corresponding to the first factor of `gr_ref`, and so on. Currently, for non-stationary distributions, the predictors do not depend on the grouping, and thus the same relationship between the variable of interest and the predictors must be assumed for each subset.

### 4.4. Censoring

The theory underlying standardised indices in Section 2 relies on the variable of interest being continuous, and the distributions available in **SEI** are therefore also all continuous. However, standardised indices may also be informative when the variable of interest is not continuous. A particular example is when the variable is censored; that is, it is bounded either above

or below (or both), with positive probability of being exactly equal to the boundary points. Precipitation, for example, is censored below at zero.

Depending on how they are used, the definition of standardised indices may need to be adjusted to account for the censoring. This has been considered, for example, by [Stagge \*et al.\* \(2015\)](#). How to deal with censored variables is discussed in detail in the appendix. Censored data can be handled in **SEI** using the `lower`, `upper`, and `cens` arguments in `std_index()`: `lower` and `upper` specify the lower and upper bounds of the data, respectively, while `cens` determines the method used to address the censoring. The available methods are outlined in the appendix.

#### 4.5. Time series manipulations

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 "mins", "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 indices, 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,
  agg_period = 1,
  agg_scale = c("days", "mins", "hours", "weeks", "months", "years"),
  agg_fun = "sum",
  timescale = c("days", "mins", "hours", "weeks", "months", "years"),
  na_thres = 10
)
```

The interpretation of the arguments is equivalent to in the discussion above: `x` represents the `xts` time series to be aggregated, `agg_period` and `agg_scale` are the length and time scale of the aggregation period, `agg_fun` is the function used in the aggregation, and `timescale` is the time scale of `x`. The final argument `na_thres` represents the maximum proportion of values that can be missing (i.e. `NA`) in the aggregation period, before the aggregation itself returns `NA`. For example, if 23 hours of precipitation accumulations are missing on one day so that only one hourly accumulation is available, this sole observation is not representative of the daily accumulation. By default, `NA` is returned if more than 10% of the values in the aggregation period are missing.

#### 4.6. Plotting standardised indices

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

```
plot_sei(
  x,
  type = c("ts", "hist", "bar"),
  title = NULL,
  lab = "Std. Index",
  xlims = NULL,
  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"`), a histogram (`type = "hist"`), or a bar plot (`type = "bar"`). If `type = "hist"`, the function is essentially a wrapper for `geom_histogram()` in **ggplot2**. The histogram and bar plot both display the distribution of the values in `x`. The histogram is particularly useful when superimposing a density onto the histogram, while the bar plot is also made available since it can be more robust when the data in `x` is bounded. Here, `n_bins` can be used to specify the number of bins in the plot.

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

#### 4.7. Drought definitions

Having obtained a time series of standardised indices, one might wish to perform an analysis of shortages, or droughts, in the variable of interest. As discussed, this can be achieved using standardised indices with the definitions of droughts in Table 1. To facilitate such analyses, the `get_drought()` function takes a time series or vector `x` as input, and outputs a dataframe containing information regarding drought occurrences and characteristics.

```
get_drought(
  x,
  thresholds = c(1.28, 1.64, 1.96),
  exceed = TRUE,
  cluster = 0,
  lag = NULL
)
```

The argument `thresholds` is a vector containing the thresholds used to define the droughts; by default, these correspond to the 90<sup>th</sup>, 95<sup>th</sup>, and 97.5<sup>th</sup> percentiles of a standard normal distribution. It is also assumed by default that a drought is defined when the standardised indices exceed these thresholds. This can be specified using the `exceed` argument. If a drought should instead correspond to an instance where the indices fall below the given thresholds, `exceed` should be `FALSE`.

When defining meteorological droughts, it is often common to introduce a lag such that the drought does not end when the standardised index no longer exceeds the drought threshold, but rather when the index falls below a lower threshold (such as zero for the SPI) (e.g. McKee *et al.* 1993). This accounts for cases where there are small fluctuations in the index around the drought threshold, classing this as one persistent drought rather than several shorter droughts. This can be implemented in **SEI** by specifying the `lag` argument, which is a numeric value denoting the weaker threshold to use when lagging the indices.

Alternatively, these fluctuations could be avoided by clustering together drought events separated by a small number of days (e.g. Otero *et al.* 2022b). The number of days between which drought events should be clustered can be specified via the `cluster` argument in `get_drought()`. The default is to not cluster droughts.

As an example, consider the previous time series of standardised indices corresponding to a random sample from a gamma distribution.

```
head(get_drought(x_std))

##           x ins occ dur      mag
## 1  0.9398288  0  0  0 0.000000
## 2 -0.1052623  0  0  0 0.000000
## 3  2.8787919  3  1  1 2.878792
```

```
## 4 0.2787468 0 0 0 0.000000
## 5 1.6555924 2 1 1 1.655592
## 6 -0.2168268 0 0 0 0.000000
```

The output is a dataframe containing the original vector or 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 (see [McKee \*et al.\* 1993](#), for details). If only one threshold is used to define a drought, rather than the three used in [Table 1](#), then the intensity is equivalent to the occurrence, and is therefore not returned.

## 5. Applications

### 5.1. Renewable energy production

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

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

##           date country      PWS
## 1 2019-01-01 00:00:00 Austria 0.07823861
## 2 2019-01-01 01:00:00 Austria 0.05330285
## 3 2019-01-01 02:00:00 Austria 0.06773745
## 4 2019-01-01 03:00:00 Austria 0.09971320
## 5 2019-01-01 04:00:00 Austria 0.18676895
## 6 2019-01-01 05:00:00 Austria 0.29001299
```

The dataframe `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).

For concision, 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)
```

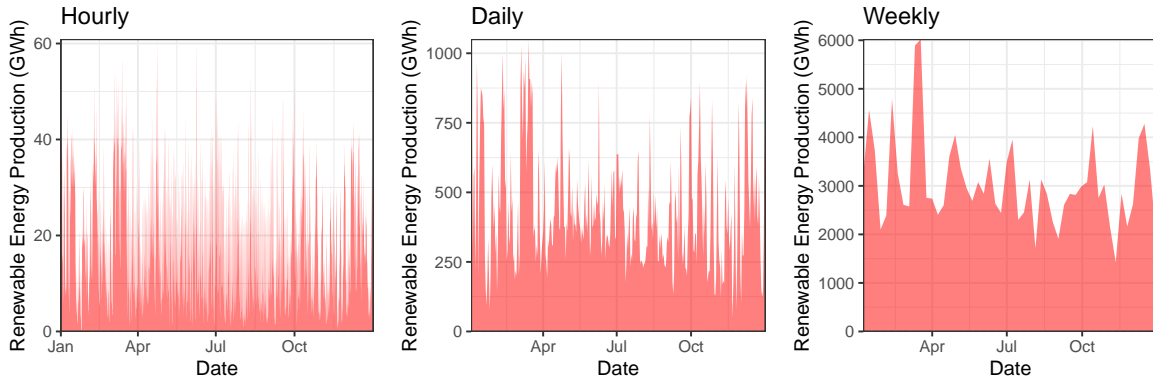


Figure 3: Time series of 2019 renewable energy production in Germany at hourly, daily, and weekly time scales.

```
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. Figure 3 demonstrates that 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). Time series of hourly, daily, and weekly SREPI are presented in Figure 4.

```
srepi_h <- std_index(de_supply_h)
srepi_d <- std_index(de_supply_d)
srepi_w <- std_index(de_supply_w, dist = "kde")
```

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, whereas the empirical distribution function is only recommended if at least 100 reference values are available. Instead, for weekly data, we estimate the distribution of production values using kernel density estimation (`dist = "kde"`). Alternative distributions can be implemented by changing the `dist` argument in `std_index()`.

```
lab <- "SREPI"
ylims <- c(-3, 3)
plot_h <- plot_sei(srepi_h, lab = lab, ylims = ylims, title = "Hourly")
plot_d <- plot_sei(srepi_d, lab = lab, ylims = ylims, title = "Daily")
plot_w <- plot_sei(srepi_w, lab = lab, ylims = ylims, title = "Weekly")
grid.arrange(plot_h, plot_d, plot_w, nrow = 1)
```

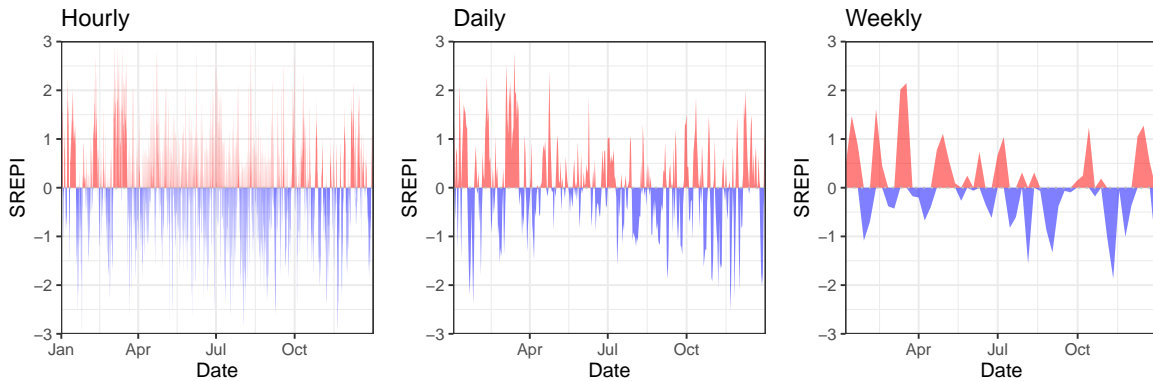


Figure 4: Time series of 2019 SREPI values in Germany at hourly, daily, and weekly time scales.

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 5. As discussed in Section 3, such a plot can be used to validate the distributional assumptions made when calculating the indices. While the raw renewable energy production follows a heavily skewed distribution, the SREPI values resemble a sample from a standard normal distribution, suggesting the empirical distribution function is appropriate in this example. If probability or bounded indices were used, then the histogram of SREPI values should be flat, rather than normal. This is illustrated in Figure 6.

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

```

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)

```

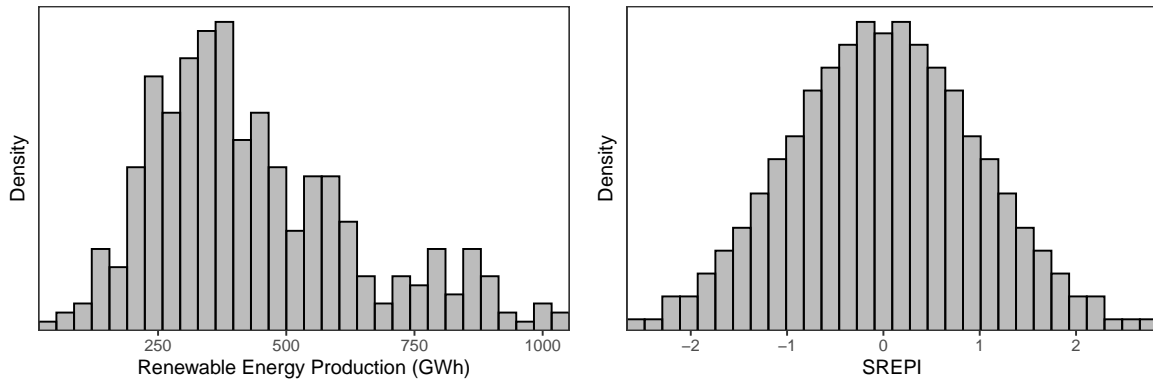


Figure 5: Histogram of 2019 daily renewable energy production and SREPI values in Germany.

```

srepi_d_prob <- std_index(de_supply_d, index_type = "prob01")
plot_prob <- plot_sei(srepi_d_prob, type = "bar", lab = "SREPI",
                    ylims = c(0, 0.1), title = "Probability")
srepi_d_bnd <- std_index(de_supply_d, index_type = "prob11")
plot_bnd <- plot_sei(srepi_d_bnd, type = "bar", lab = "SREPI",
                    ylims = c(0, 0.1), title = "Bounded")
grid.arrange(plot_prob, plot_bnd, nrow = 1)

```

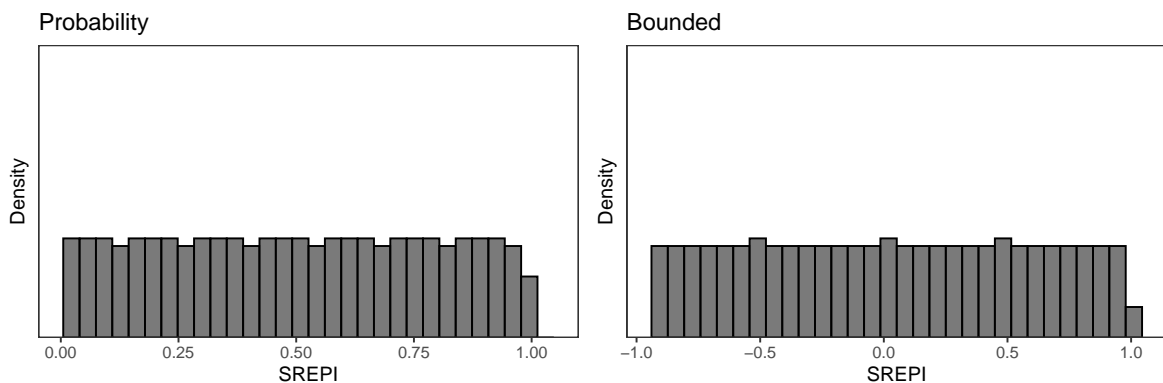


Figure 6: Histogram of 2019 daily renewable energy production and SREPI values in Germany, where the indices are probability and bounded indices.

```
thresholds <- -qnorm(c(0.9, 0.95, 0.975)) # -1.28, -1.64, -1.96
drought_df <- get_drought(srepi_d, thresholds, exceed = F)
```

From the dataframe `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,

```
num_ev <- table(drought_df$ins)
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,

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

##	1	2	3
##	11	3	6

or we could calculate the average drought magnitude,

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

##	[1]
##	2.985504

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 \(2023\)](#).

## 5.2. Wind speed

Consider now a second application of standardised indices to weather and climate variables. While standardised indices are most commonly used to construct the SPI and SPEI, they can also be applied to other climate variables (see Section 1 for examples). In this section, we use the framework discussed herein to define standardised wind speed indices. Monitoring wind speed is crucial in many applications, including aviation safety, renewable energy planning, and the design and insurance of infrastructure.

Hourly estimates of near-surface (10m) wind speed are publicly available for several European countries between 1979 and 2019 ([Hersbach \*et al.\* 2023](#)). As in [Otero \*et al.\* \(2022a\)](#), we aggregate these wind speeds to get daily nationwide averages. Consistent with the previous application to renewable energy production, the analysis is restricted to Germany, and this subset of data is available from **SEI**. For clarity of visualisation, the following analysis is limited to the three year period between 2017 and 2019.

```
lab <- "Wind speed (m/s)"
ylims <- c(0, 8)
plot_ws_d <- plot_sei(de_wind_d, lab = lab, ylims = ylims, title = "Daily")
plot_ws_w <- plot_sei(de_wind_w, lab = lab, ylims = ylims, title = "Weekly")
plot_ws_m <- plot_sei(de_wind_m, lab = lab, ylims = ylims, title = "Monthly")
grid.arrange(plot_ws_d, plot_ws_w, plot_ws_m, nrow = 1)
```

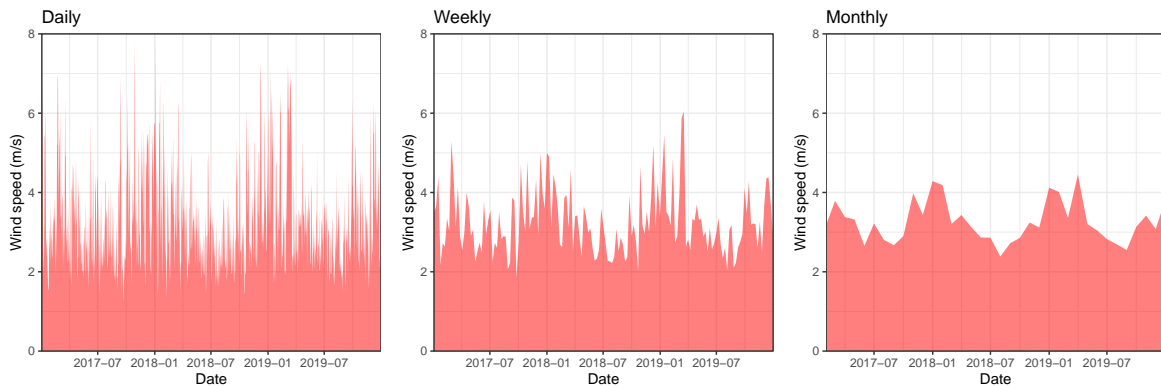


Figure 7: Daily, weekly and monthly time series of average wind speed in Germany between 2017 and 2019.

```
data("data_wind_de", package = "SEI")
data_wind_de <- subset(data_wind_de, format(date, "%Y") >= 2017)
head(data_wind_de)
```

```
##           date      wsmean
## 13882 2017-01-01 23:00:00 3.479744
## 13883 2017-01-02 23:00:00 5.391327
## 13884 2017-01-03 23:00:00 6.801471
## 13885 2017-01-04 23:00:00 3.822558
## 13886 2017-01-05 23:00:00 2.101849
## 13887 2017-01-06 23:00:00 3.180605
```

As before, the data can be rescaled using `xts` functionality to obtain time series of the weekly and monthly average wind speed in Germany. We label the resulting daily, weekly, and monthly time series `de_wind_d`, `de_wind_w`, and `de_wind_m`, respectively, all of which are `xts` objects. These time series are displayed using `plot_sei()` in Figure 7.

The raw data can be transformed to a time series of standardised indices using `std_index()`. As discussed, constructing standardised indices requires estimating the distribution of the variable of interest. While we argue that the empirical distribution provides a flexible means to achieve this, we demonstrate in this application how the **SEI** package allows parametric distributions to be used for this purpose.

Various distributions can be fit to the daily wind speeds using the `fit_dist()` function. Consider, for example, a gamma distribution, log-normal distribution, and Weibull distribution.

```

out_gamma <- fit_dist(data_wind_de$wsmean, dist = "gamma")
out_lnorm <- fit_dist(data_wind_de$wsmean, dist = "lnorm")
out_weibull <- fit_dist(data_wind_de$wsmean, dist = "weibull")

```

The three distributions can be compared with respect to their AIC, which is output from `fit_dist()`.

```

aic_vec <- c(out_gamma$fit['aic'],
            out_lnorm$fit['aic'],
            out_weibull$fit['aic'])
names(aic_vec) <- c("Gamma", "Log-normal", "Weibull")
print(aic_vec)

##      Gamma Log-normal      Weibull
## 3276.025 3228.770 3441.518

```

The AIC provides a measure of distributional fit whilst also accounting for model complexity, with a smaller value indicating a better fit. The AIC values above suggest that the log-normal distribution provides the best fit to the data, while the Weibull distribution provides the worst fit.

This can be verified by looking at p-values corresponding to the Kolmogorov-Smirnov test.

```

ksp_vec <- c(out_gamma$fit['ks_pval'],
            out_lnorm$fit['ks_pval'],
            out_weibull$fit['ks_pval'])
names(ksp_vec) <- c("Gamma", "Log-normal", "Weibull")
print(round(ksp_vec, 4))

##      Gamma Log-normal      Weibull
## 0.0044 0.1105 0.0000

```

The null hypothesis is that the sample of data has been drawn from the parametric distribution. The Kolmogorov-Smirnov test is rejected at the 0.5% level for both the gamma and Weibull distributions, providing evidence to suggest that these distributions do not accurately describe the data. On the other hand, there is insufficient evidence to reject the null hypothesis that the wind speed data was drawn from a log-normal distribution.

The fit of the three distributions to the data is presented in Figure 8. The `plot_sei()` function can be used to plot histograms of the daily wind speeds. Since `plot_sei()` returns a **ggplot2** object, the estimated density functions can be added to the histograms. For example, for the gamma distribution

```

x <- seq(0, 9, length.out = length(de_wind_d))
xlab <- "Wind speed (m/s)"
pars_gam <- out_gamma$params

```

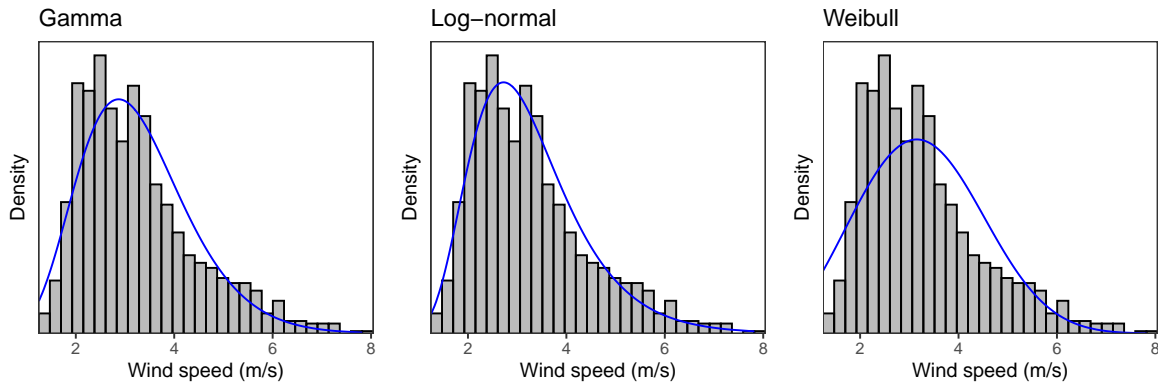


Figure 8: Distribution of daily wind speeds in Germany between 2017 and 2019, along with estimated gamma, log-normal, and Weibull densities.

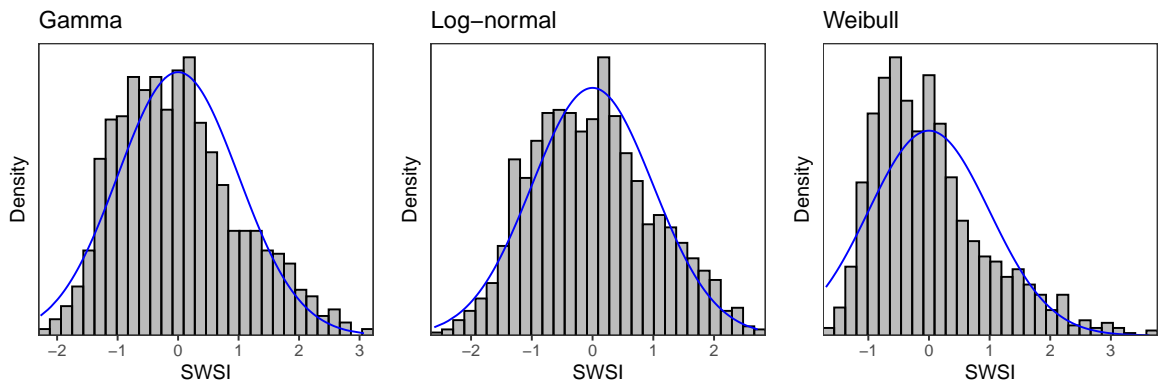


Figure 9: Distribution of standardised wind speed indices (SWSI) constructed using the gamma, log-normal, and Weibull distributions. The standard normal density is displayed in blue.

```
plt_gam <- plot_sei(de_wind_d, type = "hist", lab = xlab, title = "Gamma") +
  geom_line(aes(x = x, y = dgamma(x, pars_gam[1], pars_gam[2])), col = "blue")
```

Finally, the adequacy of the distributions can also be visualised by plotting a histogram of the standardised indices obtained using the three distributions. The standardised indices can be calculated using the `std_index()` function, with the `dist` argument corresponding to the distribution of interest.

```
sei_ws_gam <- std_index(de_wind_d, dist = "gamma")
sei_ws_lnorm <- std_index(de_wind_d, dist = "lnorm")
sei_ws_weib <- std_index(de_wind_d, dist = "weibull")
```

Histograms of the standardised indices, constructed using `plot_sei()`, are displayed in Figure 9. Similarly to before, a standard normal distribution is superimposed onto the histograms. As expected, no parametric distribution fits the data perfectly. The standardised wind speed indices constructed using a log-normal distribution appear closer to a standard normal distri-



bution, reinforcing the previous conclusions that this distribution provides a more accurate fit to the daily wind speed data.

Having constructed these wind speed indices, it would be straightforward to conduct an analysis of wind droughts (or extreme wind speeds) using the `get_drought()` function in **SEI**, akin to how renewable energy production droughts were studied in the previous application. This is omitted here for concision, but could be analysed in more detail in the future.

## 6. Summary

This vignette documents the **SEI** package in R, which provides comprehensive functionality to calculate time series of 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, particularly in the context of drought analysis. While there is no unique way to define standardised indices, several widely-adopted indices are constructed using the same, simple procedure. We outline this 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.

The **SEI** package converts a time series of observations to a time series of standardised indices. The package allows the time series to be aggregated and rescaled to different time scales, provides plot capabilities to visualise the resulting indices, and offers a range of distributions with which to calculate the indices, including flexible non-parametric and non-stationary methods. The package additionally allows users to define and analyse shortages, or droughts, of the variable under consideration. Two examples are presented in Section 5 whereby the package is employed to calculate standardised indices to monitor shortages in renewable energy production, and to calculate standardised wind speed indices.

The package has been designed to facilitate applications of standardised indices in practice. While standardised indices have received considerable attention within the field of climate science, such indices would also be relevant in several other domains. The **SEI** package is applicable in broad generality, though the package could also be extended in light of more recent extensions of standardised indices that have been proposed in the literature. For example, the package currently only considers indices corresponding to univariate variables, though multivariate extensions have also been studied in the literature (e.g. [Erhardt and Czado 2018](#); [Hao \*et al.\* 2019](#)). Similarly, functionality could also be added to automatically implement methods that remove seasonality and temporal dependence in the data ([Erhardt and Czado 2018](#)).

Moreover, while the package extends existing packages by providing flexible density estimation methods, alternative such methods could also be made available, particularly methods that are appropriate when estimating the distribution of bounded variables. Non-stationary distribution estimation is currently only available for parametric distributions via the GAMLSS framework, and further extensions could therefore also incorporate non-parametric distributional regression methods such as isotonic distributional regression ([Henzi \*et al.\* 2021](#)). Finally, the functionality of the package could be transferred to software packages in other programming languages, facilitating the implementation of standardised indices in monitoring systems that have not been designed in R.

## 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).

## Appendix: Censored data

The definition of standardised indices relies on the well-known result that if  $X$  is a continuous random variable with cumulative distribution function  $F$ , then the probability integral transform  $F(X)$  is uniformly distributed between zero and one. If the random variable  $X$  is not continuous, then the PIT will generally not be uniformly distributed. This becomes an issue when calculating standardised indices. Of particular interest is the case when the variable  $X$  is censored; that is, the distribution of  $X$  has a point mass at the boundary (or boundaries) of its support, but is continuous otherwise. Precipitation, for example, is censored below at zero.

While standardised indices for censored variables can be defined as in Section 2, [Stagge \*et al.\* \(2015\)](#) remark that the resulting indices may be misleading. For example, if  $X$  denotes daily precipitation accumulations and no precipitation occurs on 90% of days in the time series, then  $F(0) = 0.9$ . The (normal) standardised index value corresponding to a precipitation of zero would therefore be  $\Phi^{-1}(0.9) = 1.28$ , the threshold often used to define excessively large values of a variable, despite zero precipitation clearly not being extreme; similarly, if the smallest precipitation value attains a standardised index of 1.28, then the system will never enter a drought state according to the drought definitions typically employed in practice, regardless of what precipitation occurs.

To circumvent this, [Stagge \*et al.\* \(2015\)](#) propose to replace the PIT values corresponding to zero precipitation events with an alternative constant, chosen such that the resulting standardised indices “maintain statistical interpretability”. In particular, they propose to choose the constant so that the mean of the standardised indices is the same as it would be if  $X$  were continuous (zero, in the case of normal indices).

We can formalise this as follows. Suppose we are interested in a random variable  $X$  that is censored below at some value  $a$ , precipitation accumulations being a particular example with  $a = 0$ . While standardised indices are typically defined using  $F(X)$ , we want to introduce a real-valued constant  $c$  and define an adapted variable,  $\tilde{F}(X)$ , such that

$$\tilde{F}(X) = \begin{cases} F(X), & X > a, \\ c, & X = a. \end{cases}$$

Note that for  $X > a$ ,  $F(X)$  can also be written as  $F(X) = F(a) + (1 - F(a))F(X | X > a)$ , which is equivalent to the expression given in [Stagge \*et al.\* \(2015, Equation4\)](#).

For probability standardised indices (`index_type = "prob01"`), the constant should be chosen such that  $E[\tilde{F}(X)] = 1/2$ , since the expectation  $E$  of a standard uniform random variable is equal to one half. We have

$$E[\tilde{F}(X)] = E[F(X) | X > a]P(X > a) + cP(X = a).$$

Since  $F(X) | X > a$  is uniformly distributed between  $P(X = a)$  and 1, the first expectation is equal to  $(P(X = a) + 1)/2$ . Rearranging gives that  $E[\tilde{F}(X)] = 1/2$  when  $c = P(X = a)/2$ . Hence, probability indices at the censored values should be replaced with  $c = P(X = a)/2$ .

This is also the case for bounded standardised indices (`index_type = "prob11"`). In this case, we want to choose  $c$  such that  $E[2F(\tilde{X}) - 1] = 0$ , which is exactly the same requirement as for the probability indices. Using analogous arguments, if  $X$  is censored above at some value  $b$  (rather than below at some value  $a$ ), then the probability and bounded indices at the censored values should be replaced with  $c = 1 - P(X = b)/2$ .

The constant proposed by [Stagge \*et al.\* \(2015, Equation 3\)](#) is a sample estimate of  $P(X = a)/2$ . However, since the Gaussian inverse function  $\Phi^{-1}$  is a non-linear function, for an arbitrary random variable  $Y$  generally  $E[\Phi^{-1}(Y)] \neq \Phi^{-1}(E[Y])$ . Hence, normal standardised indices will generally not have expectation zero when the constant  $c$  is chosen as above; this is also highlighted by Figure 1b in [‘Stagge \*et al.\* \(2015\)](#), where the mean of the normal standardised indices with their proposed censoring methodology is generally not zero.

Instead, we propose an alternative constant when normal standardised indices are used, which results in a mean index value equal to zero. We now want to find the constant  $c$  such that

$$E[\Phi^{-1}(\tilde{F}(X))] = E[\Phi^{-1}(F(X)) | X > a]P(X > a) + \Phi^{-1}(c)P(X = a) = 0.$$

Since  $F$  and  $\Phi^{-1}$  are increasing functions, we have that

$$E[\Phi^{-1}(F(X)) | X > a] = E[\Phi^{-1}(F(X)) | \Phi^{-1}(F(X)) > \Phi^{-1}(F(a))],$$

which is just the expectation of a normal distribution truncated below at  $l = \Phi^{-1}(F(a)) = \Phi^{-1}(P(X = a))$ . This is therefore equal to

$$\frac{\phi(l)}{1 - \Phi(l)},$$

where  $\phi$  is the probability density function of the standard normal distribution. Similarly,  $P(X > a) = P(\Phi^{-1}(F(X)) > l) = 1 - \Phi(l)$ . Rearranging the above equality then yields

$$c = \Phi \left[ \frac{-\phi(l)}{P(X = a)} \right],$$

and the normal indices at the censored values should therefore be replaced with  $\Phi^{-1}(c) = -\phi(l)/P(X = a)$ . If  $X$  is instead bounded above at  $b$ , then the constant  $c$  becomes

$$c = \Phi \left[ \frac{\phi(u)}{P(X = b)} \right],$$

where  $u = \Phi^{-1}(1 - P(X = b))$ .

In practice, we observe a sample of realisations of the variable  $X$ . The terms  $P(X = a)$  and  $P(X = b)$  can be estimated from a sample using the relative number of realisations that are censored, and these can then be used to determine the constant  $c$ . Note that since these terms have been calculated on a population level, the sample mean of the standardised indices will generally not be exactly zero (or 1/2 for probability indices), but it will be close to zero in large enough samples.

Variables could also be censored both below at  $a$  and above at  $b$ . For example, the fraction of cloud cover over a domain is bounded between  $a = 0$  and  $b = 1$ , with positive probability of taking on either of the boundary values. In this case, we must choose how to assign values at both boundary points. This adds an extra degree of freedom, since there are infinite

combinations of two points that would lead to the standardised indices having mean zero (or mean one half for probability indices). A second criterion could be introduced, such as the variance of the indices being equal to one (or 1/12). However, such calculations go beyond the scope of this paper, and we therefore leave this for future work.

To implement censoring within the **SEI** package, the arguments `lower` and `upper` to the functions `std_index()` and `get_pit()` allow the user to indicate that the data is censored below at  $a$  (`lower`) and/or above at  $b$  (`upper`). The argument `cens` then specifies the method used to deal with censoring. This can be one of four options: if `cens = "none"`, then no censoring is performed, i.e.  $\tilde{F} = F$ ; if `cens = "prob"`, then the approach above is used that results in the probability indices having mean equal to 1/2, which is essentially the same as the original approach proposed by Stagge *et al.* (2015); if `cens = "normal"`, then the approach above is used that results in the normal indices having mean equal to zero; alternatively, `cens` can be a single numeric value, allowing the user to manually specify the constant  $c$ . As discussed above, we recommend using `cens = "prob"` when working with probability or bounded standardised indices, and `cens = "normal"` when normal standardised indices are used. If the variable is bounded both below and above (i.e. `lower` and `upper` are both finite), then `cens` must be a numeric vector of length two, specifying the constants at which to assign values at the lower and upper boundary points, respectively.

## References

- Allen S, Otero N (2023). “Standardised Indices to Monitor Energy Droughts.” *Renewable Energy*, **217**, 119206. doi:10.1016/j.renene.2023.119206.
- Ayala-Bizarro I, Zúñiga-Mendoza J (2016). *SPIGA: Compute SPI Index Using the Methods Genetic Algorithm and Maximum Likelihood*. R package version 1.0.0, URL <https://CRAN.R-project.org/package=SPIGA>.
- Beguéría 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.
- Beguéría S, Vicente-Serrano SM (2023). *SPEI: Calculation of the Standardized Precipitation-Evapotranspiration Index*. R package version 1.8.1, URL <https://CRAN.R-project.org/package=SPEI>.
- 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.
- Bloomfield J, Marchant B (2013). “Analysis of Groundwater Drought Building on the Standardised Precipitation Index Approach.” *Hydrology and Earth System Sciences*, **17**, 4769–4787. doi:10.5194/hess-17-4769-2013.
- Erhardt TM, Czado C (2018). “Standardized Drought Indices: A Novel Univariate and Multivariate Approach.” *Journal of the Royal Statistical Society Series C*, **67**, 643–664. doi:10.1111/rssc.12242.

- Gudmundsson L, Stagge JH (2016). **SCI: Standardized Climate Indices such as SPI, SRI or SPEI**. R package version 1.0-2, URL <https://CRAN.R-project.org/package=SCI>.
- Guttman NB (1999). “Accepting the Standardized Precipitation Index: A Calculation Algorithm.” *Journal of the American Water Resources Association*, **35**, 311–322. doi: [10.1111/j.1752-1688.1999.tb03592.x](https://doi.org/10.1111/j.1752-1688.1999.tb03592.x).
- Hao Z, Hao F, Singh VP, Zhang X (2019). “Statistical Prediction of the Severity of Compound Dry-hot Events Based On El Niño-Southern Oscillation.” *Journal of Hydrology*, **572**, 243–250. doi: [10.1016/j.jhydrol.2019.03.001](https://doi.org/10.1016/j.jhydrol.2019.03.001).
- Henzi A, Ziegel JF, Gneiting T (2021). “Isotonic Distributional Regression.” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, **83**, 963–993. doi: [10.1111/rssb.12450](https://doi.org/10.1111/rssb.12450).
- Hersbach H, Bell B, Berrisford P, Biavati G, Horányi A, Muñoz Sabater J, Nicolas J, Peubey C, Radu Rand Rozum I, Schepers D, Simmons A, Soci C, Dee D, Thépaut JN (2023). “ERA5 Hourly Data on Single Levels From 1940 to Present. Copernicus Climate Change Service (C3S) Climate Data Store (CDS).” <https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-single-levels?tab=overview>. Accessed: 01-09-2022.
- Kneib T, Silbersdorff A, Säfken B (2023). “Rage Against the Mean—A Review of Distributional Regression Approaches.” *Econometrics and Statistics*, **26**, 99–123. doi: [10.1016/j.ecosta.2021.07.006](https://doi.org/10.1016/j.ecosta.2021.07.006).
- Li J, Wang Y, Li S, Hu R (2015). “A Nonstationary Standardized Precipitation Index Incorporating Climate Indices as Covariates.” *Journal of Geophysical Research: Atmospheres*, **120**, 12–082. doi: [10.1002/2015JD023920](https://doi.org/10.1002/2015JD023920).
- 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](https://doi.org/10.5194/hess-25-1587-2021).
- Maetens W (2019). **standaRdized**. GitHub repository release v.1.0, URL <https://github.com/WillemMaetens/standaRdized>.
- Massey Jr FJ (1951). “The Kolmogorov-Smirnov Test for Goodness of Fit.” *Journal of the American Statistical Association*, **46**, 68–78. doi: [doi.org/10.1080/01621459.1951.10500769](https://doi.org/10.1080/01621459.1951.10500769).
- 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.
- Nussbaumer E (2021). **standard-precip: Functions to Calculate SPI and SPEI**. Python package version 1.0, URL <https://pypi.org/project/standard-precip/>.
- Otero N, Martius O, Allen S, Bloomfield H, Schaeffli B (2022a). “Characterizing Renewable Energy Compound Events Across Europe Using a Logistic Regression-based Approach.” *Meteorological Applications*, **29**, e2089. doi: <https://doi.org/10.1002/met.2089>.

- Otero N, Martius O, Allen S, Bloomfield H, Schaeffi B (2022b). “A Copula-Based Assessment of Renewable Energy Droughts Across Europe.” *Renewable Energy*, **201**, 667–677. doi: [10.1016/j.renene.2022.10.091](https://doi.org/10.1016/j.renene.2022.10.091).
- Python (2017). *Python Software, Version 3.6.4*. Beaverton, OR. URL <https://www.python.org/>.
- Rashid MM, Beecham S (2019). “Development of a Non-Stationary Standardized Precipitation Index and its Application to a South Australian Climate.” *Science of the Total Environment*, **657**, 882–892. doi: [10.1016/j.scitotenv.2018.12.052](https://doi.org/10.1016/j.scitotenv.2018.12.052).
- R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Rigby RA, Stasinopoulos DM (2005). “Generalized Additive Models for Location, Scale and Shape.” *Journal of the Royal Statistical Society Series C: Applied Statistics*, **54**, 507–554. doi: [10.1111/j.1467-9876.2005.00510.x](https://doi.org/10.1111/j.1467-9876.2005.00510.x).
- Russo S, Dosio A, Sterl A, Barbosa P, Vogt J (2013). “Projection of Occurrence of Extreme Dry-wet Years and Seasons in Europe with Stationary and Nonstationary Standardized Precipitation Indices.” *Journal of Geophysical Research: Atmospheres*, **118**, 7628–7639. doi: [10.1002/jgrd.50571](https://doi.org/10.1002/jgrd.50571).
- Shao S, Zhang H, Singh VP, Ding H, Zhang J, Wu Y (2022). “Nonstationary Analysis of Hydrological Drought Index in a Coupled Human-Water System: Application of the GAMLSS with Meteorological and Anthropogenic Covariates in the Wuding River Basin, China.” *Journal of Hydrology*, **608**, 127692. doi: [10.1016/j.jhydrol.2022.127692](https://doi.org/10.1016/j.jhydrol.2022.127692).
- Shukla S, Wood AW (2008). “Use of a Standardized Runoff Index for Characterizing Hydrologic Drought.” *Geophysical Research Letters*, **35**, L02405. doi: [10.1029/2007GL032487](https://doi.org/10.1029/2007GL032487).
- Silverman BW (2018). *Density Estimation for Statistics and Data Analysis*. Routledge, New York. doi: [10.1201/9781315140919](https://doi.org/10.1201/9781315140919).
- Stagge JH, Tallaksen LM, Gudmundsson L, Van Loon AF, Stahl K (2015). “Candidate Distributions for Climatological Drought Indices (SPI and SPEI).” *International Journal of Climatology*, **35**, 4027–4040. doi: [10.1002/joc.4267](https://doi.org/10.1002/joc.4267).
- Stasinopoulos DM, Rigby RA (2008). “Generalized Additive Models for Location Scale and Shape (GAMLSS) in R.” *Journal of Statistical Software*, **23**, 1–46. doi: [10.18637/jss.v023.i07](https://doi.org/10.18637/jss.v023.i07).
- Stasinopoulos MD, Rigby RA, Heller GZ, Voudouris V, De Bastiani F (2017). *Flexible Regression and Smoothing: Using GAMLSS in R*. CRC Press.
- 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](https://doi.org/10.1175/2009JCLI2909.1).
- Vicente-Serrano SM, López-Moreno JI, Beguería S, Lorenzo-Lacruz J, Azorin-Molina C, Morán-Tejeda E (2012). “Accurate Computation of a Streamflow Drought Index.” *Journal of Hydrologic Engineering*, **17**, 318–332. doi: [10.1061/\(ASCE\)HE.1943-5584.0000433](https://doi.org/10.1061/(ASCE)HE.1943-5584.0000433).



- Vonk M (2017). **spei**: *A Simple Python Package to Calculate Drought Indices for Time Series such as the SPI, SPEI and SGI*. Python package version 2.0.0, URL <https://pypi.org/project/spei/>.
- Wang Y, Li J, Feng P, Hu R (2015). “A Time-Dependent Drought Index for Non-Stationary Precipitation Series.” *Water Resources Management*, **29**, 5631–5647. doi: [10.1007/s11269-015-1138-0](https://doi.org/10.1007/s11269-015-1138-0).
- Zargar A, Sadiq R, Naser B, Khan FI (2011). “A Review of Drought Indices.” *Environmental Reviews*, **19**, 333–349. doi:[10.1139/a11-013](https://doi.org/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](https://doi.org/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](mailto: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](mailto:noelia.otero@unibe.ch)  
*and*  
Oeschger Centre for Climate Change Research