Royal Netherlands
Meteorological Institute
*Ministry of Infrastructure
and Water Management*

Standard method for determining a climatological trend

C.F. de Valk

De Bilt, 2020 | Technical report; TR-389

Contents

1. Introduction

KNMI needs a standard method for determining a trend, to be used routinely and in standard KNMI publications[1]. A trend is a representation of the long-term change, separated from short-term fluctuations. It is derived from a trendline (also known as “smooth”): a smooth approximation of the time-series.

This report compares several common methods for determining a trendline according to a list of criteria. The standard method was selected by a committee, following consultation of a group of experts. It is a specific version of local linear regression (LOESS), tuned to match the climatological 30-year running average. Further details on this method and its application are provided. These include testing whether the data are consistent with no long-term change over a certain time-interval.

Theo Brandsma, Rob van Dorland, Benadet Overbeek, Gerard van der Schrier, Peter Siegmund, Werenfried Spit, Philippe Steeghs and Cees de Valk participated in the selection of the standard method. Theo Brandsma kindly reviewed this report.

In addition, valuable input was provided by Jules Beersma, Henk van den Brink, Irene Garcia, Geert Lenderink, Geert Jan van Oldenborgh, Aart Overeem, Rinus Scheele, Andreas Sterl and Kirien Whan.

[1] This does not include research reports and scientific articles.

2. Strengths and weaknesses of several common methods

Below, four common candidate methods for trendline estimation are discussed. They are compared on the basis of the following criteria:

- Is it simple? A simple method is easy to explain to a broad audience, and also easy to compute.
- Can it be applied in the same way to every variable?
- Is the trendline strictly local in time? In other words, is the value of the trendline for a given year only dependent upon values within a bounded time interval? The advantage of a local trendline is that for all years sufficiently far into the past, the trendline is fixed and will not change in the future when new data becomes available.
- Is it flexible? That is, can it approximate any trendline shape, as long as the trendline is smooth enough? A flexible trendline avoids the bias associated with an imposed shape.
- Is it compatible with climatological convention? If it does, then we can relate trendline values to reports written even many years ago, and meteorologists will have little trouble with interpretation.
- Is it free from arbitrary choices, or depends only weakly on such choices?
- Is it defined throughout the time-interval covered by the data?

2.1 Traditional approach

The traditional definition of a climatological normal is a 30-year running average. Averaging over a sliding 30-year window produces a trendline which is much smoother than the original time series of annual values. The approach has several advantages:

strengths:

- it is simple,
- it can be applied in the same way to every variable,
- the trendline is strictly localized: it only depends on the previous 15 years and the subsequent 15 years[2],
- it is flexible ,
- it agrees with climatological convention,
- no arbitrary choices are involved (the fixed time-scale is a convention, so it is not arbitrary).

weakness:

- it is not defined throughout the interval covered by the time-series: for a time-series covering a long period of time, the trendline is missing for the first 14 years, and can only be determined up to 15 years before the current year.

With respect to this last bullet, we could extend the 30-year average towards either end of the time-series by keeping the trendline constant there (so it is still based on 30 years of data), or use

[2] This is true on January 1 of the 16th year at 00:00.

averages over shorter time intervals, which are less precise. In the former case, the trendline looks rather artificial (as if a decrease or increase stops near the end of the series). In the second case, the trendline will fluctuate wildly near the beginning and the end of the series, which is not realistic either. So neither option is attractive.

2.2 Linear trend

Another simple way to represent a trendline is as a linear function plus an offset; the offset and slope can be determined by least-squares regression.

strengths:

- simple,
- defined throughout the interval covered by the time-series.

weaknesses:

- not applicable in the same way to all variables: for example, for a nonnegative variable (e.g. precipitation), the trendline might become negative (although this seems an extreme case),
- not localized in time,
- not flexible,
- does not agree with the climatological convention.
- involves an arbitrary choice (the trendline must be a linear function plus an offset).

2.3 GAM smoothing

A GAM (generalized additive model) is a flexible but smooth curve, which can follow any type of smooth trendline, if given enough degrees of freedom. It represents a chosen invertible function of, say, temperature (for example temperature itself, or its logarithm) as a linear combination of smooth basis functions plus an offset (a spline). In the estimation of the coefficients (the weights of the basis functions), the roughness of this function (its mean-square curvature) is penalized in order to produce a smooth function, and to ensure that the solution is unique. The smoothness depends on a parameter, which can be estimated from the data (e.g. by REML; see Wood, 2011); this works rather well in practice. It can happen that the estimated trend reduces to a simple linear trend, if fluctuations in the data around the linear trend cannot be distinguished from noise.

Fitting a GAM requires an assumption about the distribution function of the data; then the smooth function above represents a (time-varying) parameter of this distribution, for example the mean value. In advanced methods, several parameters of the distribution can be represented by smooth functions, each with its own degree of smoothness which can be estimated. The simplest and most frequently used version is for a normal distribution with constant variance but varying mean, so the only smooth function to be estimated is the function representing the mean value. This version can be seen as a least-squares estimator, even if the data are not normally distributed conditional on the trendline. Therefore, in this aspect, it is consistent with a local average (which is also a least-squares estimate).

Spline smoothing is roughly equivalent to the more familiar Kriging method; the estimation of the smoothness parameters in GAM replaces the estimation of the covariance function or semivariogram prior to computation of the Kriging estimate (the equivalence is described in Wahba, 1990).

strengths:

- defined throughout the interval covered by the time-series,
- flexible,
- no subjective choice is involved (smoothness is self-tuned).

weaknesses:

- not simple (the lengthy explanation above is still far from complete)
- not localized in time,
- does not agree with convention.

As for linear regression, the method is in principle not applicable in the same way to all variables, as the trendline may exceed bounds on the variable. However, because of the flexibility of a GAM trendline, this will not often occur in practice, and can be handled by imposing the bounds.

2.4 Local polynomial regression (LOESS smoothing)

LOESS smoothers have been popular as flexible tools for estimating nonlinear trendlines, certainly before the development of GAMs. A LOESS smoother generalizes the local average to a (weighted) local linear or (more generally) local polynomial regression (Cleveland and Devlin, 1988). In contrast to GAM smoothing, it has the advantage of being localized in time, just like the local average.

In fact, the trendline estimate for the midpoint of a 30-year interval obtained by unweighted local linear regression (a LOESS smooth of order 1 with uniform weights) is identical to the local average over the 30-year interval: in the midpoint, the local linear regression is always equal to the average. Therefore, the local linear regression and the local average only differ near the beginning and end of the time-series. There, the linear regression is generally less biased than a constant trendline, and more stable (less variable) than averages over increasingly narrow intervals. So local linear regression with uniform weights over 30-year intervals offers an effective way to extend the 30-year averages towards the beginning and the end of the time-series.

Standard implementations of LOESS (e.g. loess in R) do not use uniform weights on the data over an interval, but a function increasing from zero to its maximum from the boundary toward the centre. The standard choice is the so-called tricubic function. For an interval $[t_0 - d/2,\ t_0 + d/2]$ of width d, it is given by

$$f(t) = \max\{0,\ 1 - (2|t - t_0|/d)^3\}^3\ ;$$

This function is drawn in the next figure for the case $d = 2$, $t_0 = 0$.

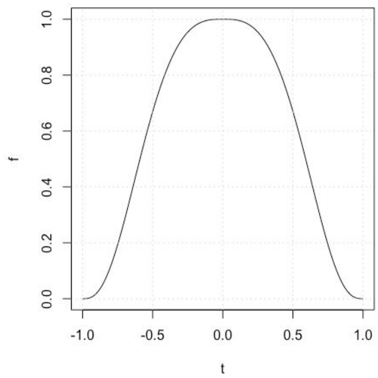

Figure: Tricubic weight function

Using such a smooth weight function gives a trendline with a smooth appearance. However, to obtain a trendline with the same variance as the trendline computed using a uniform weight function, the width d of the interval needs to be increased. Away from the beginning and end of the time-series, this width can be approximated based on eq. (2.44) in Loader (1999). For the tricubic weight function, a window of 42 years gives the same variance as a 30-year window with uniform weights. This is a modest increase, so the trendline estimate is still fairly localized: trendline values for 21 or more years before the end of the time-series will not change in the future. Applied to several datasets, the trendlines computed using the tricubic weight function appear to be close to the 30-year averages, but small-scale wiggles in the 30-year averages are smoothed, due to the continuity of the tricubic weight function; see the examples in Chapter 5.

Regardless of whether uniform or tricubic weights are chosen, the specific version of the linear LOESS trendline described above has the following properties.

strengths:

- it is relatively simple (away from the beginning and end of the time-series, it is close to the 30-year average)
- the trendline is localized in time
- it is flexible (it can approximate any trendline shape, as long as the trendline is smooth enough),
- it agrees exactly or approximately with climatological convention (so we can relate values to reported 30-year averages)
- no arbitrary choice is involved (the fixed time scale agrees with convention).
- it is defined throughout the interval covered by the time-series, and is computed everywhere using the same algorithm.
- it can be applied in the same way to every variable (just as the 30-year average), except possibly near the beginning and the end of the time-series (so for bounded variables, some measures need to be taken to ensure that bounds are respected there).

The favourable scoring on all criteria applies specifically to a time-series sampled with a uniform time step, and is due to the fact that we can fix the degree of smoothing in order to approximate a

30-year average. Without the criterion of agreement to the convention of a 30-year average, the degree of smoothing to be applied in the LOESS regression would have to be chosen, and GAM (which estimates the degree of smoothing from the data) would have one advantage over LOESS.

3. Proposed method

Based on the criteria considered above, the methods considered above would be ranked as follows (beginning with the most suitable method):

1. Linear LOESS trendline (with constant weight over 30-year intervals or tricubic weight over 42-year intervals)
2. 30-year average
3. GAM
4. Linear regression.

We have seen that the linear LOESS trendline with properly chosen settings extends the 30-year average trendline to the beginning and the end of the time-series either exactly or approximately. There are other options for extending the 30-year average. A combination of averages over short intervals and GAM was investigated. It performs rather well, but not as well as the linear LOESS trendline with tricubic weight over a 42 year window.

In Section 5.1, trendlines estimated on some time-series of daily mean, maximum and minimum temperature are shown. These trendlines are computed using the linear LOESS trendline with tricubic weight over a 42 year window, the 30-year average, and (for temperature data) the GAM trendline. The following is observed:

1. The linear LOESS trendline matches the 30-year averages closely, but without the wiggles in the latter.
2. In some cases, the GAM trendline switches rather abruptly from a linear fit to a nonlinear fit if additional data are added. This is not the case with the LOESS trendline: it adapts smoothly to new data.
3. Furthermore, a linear LOESS as above is much more localized in time than a GAM, so the trendline value is eventually fixed, unlike the GAM trendline.

Furthermore, in Section 5.2, linear LOESS trendlines with tricubic weight over a 42 year window fitted to annual maxima of daily precipitation are shown. Despite the high skewness of these data, the LOESS trendlines are smooth but match the 30-year averages closely.

Based on the information in this document and further discussions, the locally linear (LOESS) least-squares regression with tricubic weight over a 42 year time interval is selected as the standard method for determining a trendline from a time-series.

4. Additional details on the linear LOESS trendline

4.1 Simple description of the trendline method

The linear LOESS trendline with tricubic weight over a 42-year time interval could be described in simple terms to a general audience as “Smooth trendline approximating a 30-year running average”[3].

Technical description: local linear least squares regression (LOESS) with tricubic weight function over a 42-year window. Standard references are Cleveland and Devlin (1988) and Loader (1999).

4.2 Implementation

A frequently used implementation is http://www.netlib.org/a/dloess. It is made available in the R core as the function loess.R, which calls a library function.

The standard parameter settings for loess.R are not suitable (they involve approximations which are not needed). Given that the years are stored in a vector t and the values of the variable of interest are stored in a vector y. Then the correct call to loess.R is:

```
control <- loess.control(surface = "direct", statistics= "exact")
span <- 42/length(t)
mdl <- loess(y~t, data= data.frame(t= t, y=y), span= span, degree= 1, control=
control)
pre <- predict(mdl, newdata= data.frame(t= t), se= TRUE)$fit
```

Now the trendline values are found in pre$fit, and the standard deviations of the trendline values are found in pre$se.fit. Note: the “family” argument of loess.R is set to “gaussian” by default, so it does not need to be specified.

For the purpose of trendline computation at KNMI, a wrapper called climatrend.R is written which takes care of the parameter settings, computes confidence limits (see Section 4.4) and optionally, carries out a test of the trend (see Section 4.5).

4.3 Computation of the trendline near the beginning and end of the time-series

Key to the stability of the LOESS trendline near both ends of the time-series is that LOESS always uses the same number of data points for the local regression. In our case, with a 42-year window and annual values, this is 42. For an instant of time t near the say (end) of the time-series, the trendline is computed from data within a 42-year interval which is not centred at t. In that case, the linear term in the local regression model becomes important. However, the weight function is still centred at t (it is cut-off at the end of the series and stretched to cover the 42 years). Therefore, the trendline is approximately, but not exactly linear near the end of the time-series. In practice, this offers a good compromise between bias and variability.

[3] In Dutch, for example: “Gladde trendlijn, bij benadering het 30-jaar lopend gemiddelde”.

4.4 Confidence intervals

Confidence intervals are a way to show the effect of short-term fluctuations in the data on the computed trendline, when these fluctuations are regarded as arbitrary (they could have been different, in view of the chaotic nature of weather). For the computation of confidence intervals, it is assumed that successive deviations from the trendline in the time-series are uncorrelated. This may not be exactly true, but for the purpose of determining the precision of the local trendline of an annually sampled meteorological variable in Europe, it seems to be adequate (spectra for time scales below 50 years tend to be rather flat, despite phenomena like NAO).

The confidence intervals are symmetric, and based on the normal approximation of errors in the local trendline, computed in the usual manner for weighted linear regression. Using a 42-year window and tricubic weight function to compute the trendline at more than 21 years from either end, the trendline is just a weighted average over 42 years (corresponding to an unweighted average over 30 data points; see Chapter 3), so a normal approximation of the distribution of this average is generally considered acceptable.

To check this, tests with independent standard lognormal random variables were carried out. The trendline is flat in this case, and equal to the mean value 1.65 of the standard lognormal distribution. The lognormal distribution is highly skewed, which tends to skew the true confidence interval. As test data, 100,000 time-series of 100 values were simulated, from which the linear LOESS trendlines were estimated with their confidence intervals.

The next figure shows the simulated mean of the estimated trendlines (full) and the 2.5% and 97.5% quantiles of the estimated trendlines (dashed). The latter can be regarded as the exact 2-sided 95%-confidence limits. In blue, the mean values of the 95%-confidence limits estimated from the LOESS trendline and its standard deviation under the normal approximation (blue) are shown.

We can see that there is an error, but it is not large (7%-8% for both limits in the central part of the time-series). In fact, the coverage probability (the probability that the true trendline is in the estimated confidence interval) is 93%-94% over the entire length of the time-series. Compared to the ideal value of 95%, this is excellent.

Since the lognormal distribution is rather strongly skewed, we can conclude that the confidence intervals computed from LOESS combined with the normal approximation are fit for purpose. As an alternative, we could use a sophisticated bias-correcting version of bootstrapping for estimating confidence limits. However, in view of the good coverage probability in the example and the limitations of bootstrap methods, this is not worth the effort.

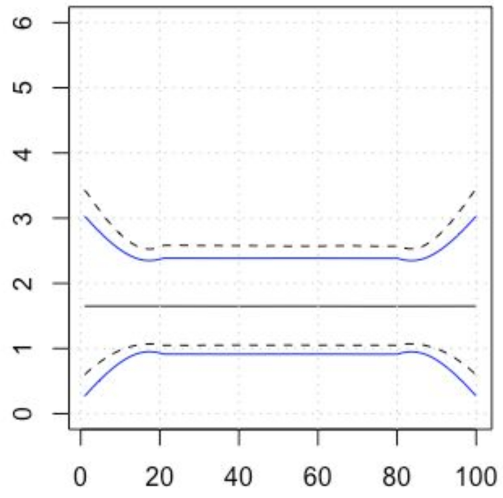

Figure: Exact confidence limits (dashed) and average of estimated confidence limits (blue) for the linear LOESS trendline with tricubic weight over a 42-year window. Confidence level is 95%. See also text.

4.5 Testing for absence of a trend

Testing for “absence of a trend” (i.e., testing of the proposition of no long-term change) is best done based on the computed trendline values $y(t_1)$ and $y(t_2)$ for two widely separated years t_1 and t_2. Preferably, both years are not too close to the beginning or the end of the time-series, where the trendline is more accurate. The exact choice of these years should be made without considering the actual data or the trendline estimate.

Then the standard deviations of the LOESS trendline estimates and the normal approximation of errors in the trendline estimates can be used for testing the proposition that the difference between the trendline values in these years is zero. If the gap between the years t_1 and t_2 is wide enough, the two trendline values $y(t_1)$ and $y(t_2)$ can be treated as independent, and their difference is approximately normal with mean zero and variance equal to the sum of the variances of the two trendline estimates. Therefore, we can easily compute the p-value p: the probability of obtaining a difference $y(t_2) - y(t_1)$ which is at least as large in magnitude as has been observed. It is given by

$$p = 2\{1 - \Phi(T)\} \qquad \text{with} \qquad T := \frac{|y(t_2) - y(t_1)|}{\sqrt{(\sigma_2^2 + \sigma_1^2)}} ,$$

in which y is the trendline estimate, Φ is the standard normal (cumulative) distribution function, and σ_1 and σ_2 are the standard deviations of $y(t_1)$ and $y(t_2)$, respectively.

The p-value should be regarded as a basic metric of compatibility between the data and the proposition of no long-term change, taking fluctuations in the data into account. The trendline script contains the option to compute this p-value for two selected years t_1 and t_2.

Alternatively, we could compute a one-sided p-value, to test the proposition of no (long-term) change against the specific alternative of a positive trend, or a negative trend (in the former case, for example, the p-value is $p_+ = 1 - \Phi(T_+)$ with $T_+ := \{y(t_2) - y(t_1)\}/(\sigma_2{}^2 + \sigma_1{}^2)^{1/2}$) . However, to avoid testing multiple propositions simultaneously or sequentially, which is more complicated if done properly, it is recommended to test only the simple proposition of no change against its negation. For the same reason, it makes little sense to compute p-values for a range of years. Therefore, the simple two-sided test for given values of t_1 and t_2 is implemented in de code in Section 6.1.

A check of the procedure on the lognormal data of the previous paragraph, comparing values at $t_1 = 20$ and $t_2 = 80,$ gives excellent results; the simulated p-values match the exact ones almost perfectly. This is partly explained by the fact that bias in the confidence limits is approximately uniform over the time-series (see previous figure), so this error is largely cancelled when taking the difference between $y(t_2)$ and $y(t_1)$.

To ensure independence of the trendline estimates for the years t_1 and t_2 , t_1 and t_2 should ideally be separated by more than 42 years (the width of the LOESS window). However, in checks on lognormal data, a spacing of 30 years (the “effective width” of the LOESS window) appears to be sufficient in practice. This remains so even if t_1 is close to the beginning, or t_2 is close to the end of the time-series. However, it is recommended to keep t_1 and t_2 away from the beginning and end of the time-series, if possible, as this gives somewhat better performance of the test.

4.6 Communicating the outcome of the trend test

To communicate a p-value in publications aimed at a general audience, it is recommended to keep the language simple and non-technical (even many scientists do not properly understand statistical testing, and “statistical significance” is a highly questionable concept for many reasons). If the p-value is small (e.g. < 0.05), we could say “the data indicate a long-term increase (or decrease) in temperature of 1.2 degrees between 1950 and 2019”. If the p-value is not that small, then we could say “the data indicate little long-term change in temperature between 1950 and 2019” (in the latter case, the temperature difference does not need to be mentioned). For those familiar with statistical testing, the p-value may be quoted in the form “(p= 0.013)”. However, it is not advisable to try to explain the concept of a p-value in simple terms to the general audience.

There is no “magic threshold” for the p-value which allows us to confidently reject or confirm the hypothesis of no change. This is not really a problem as long as we use cautious language as suggested above to communicate the result, and avoid terms like “significant” which give a false impression of certainty. Another reason for using cautious language is that not all assumptions underlying the test may be strictly valid.

5. Examples

5.1 Trendlines for temperature

To compare methods, several trendlines for the annual mean temperature at de Bilt are shown below. The panel on the left shows 30-year averages. In the centre, we see the GAM trendline, and on the right, the local linear (LOESS) trendline with tricubic weight function over 42-year intervals.

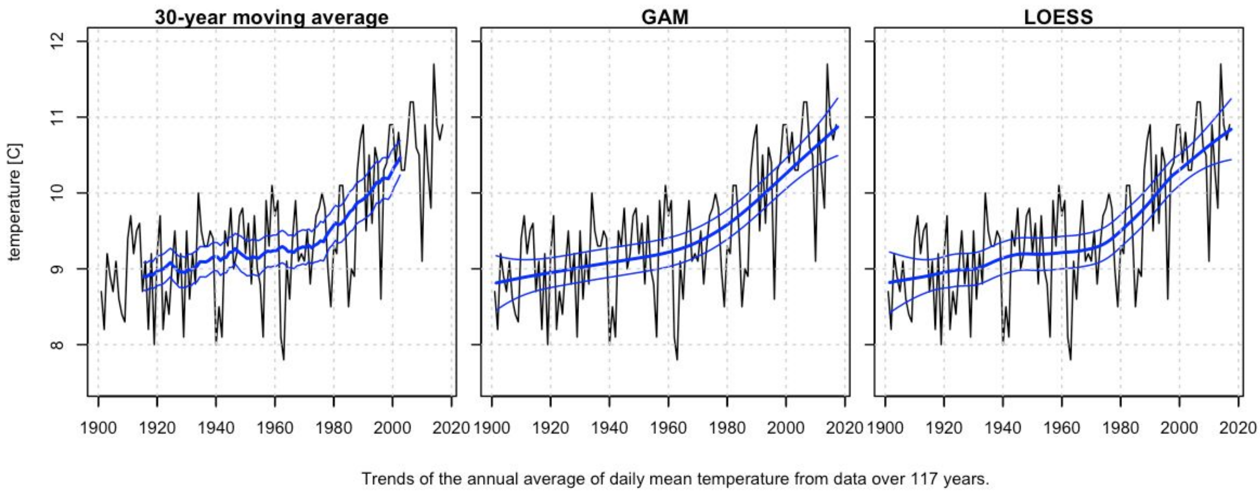

Figure 1: trendlines for annual mean temperature at de Bilt: moving averages (left), GAM smooth (centre), and linear LOESS smooth with 95% confidence interval (right).

It is important how the trendline changes when data are added at the end of the time-series. This is shown in the next figures.

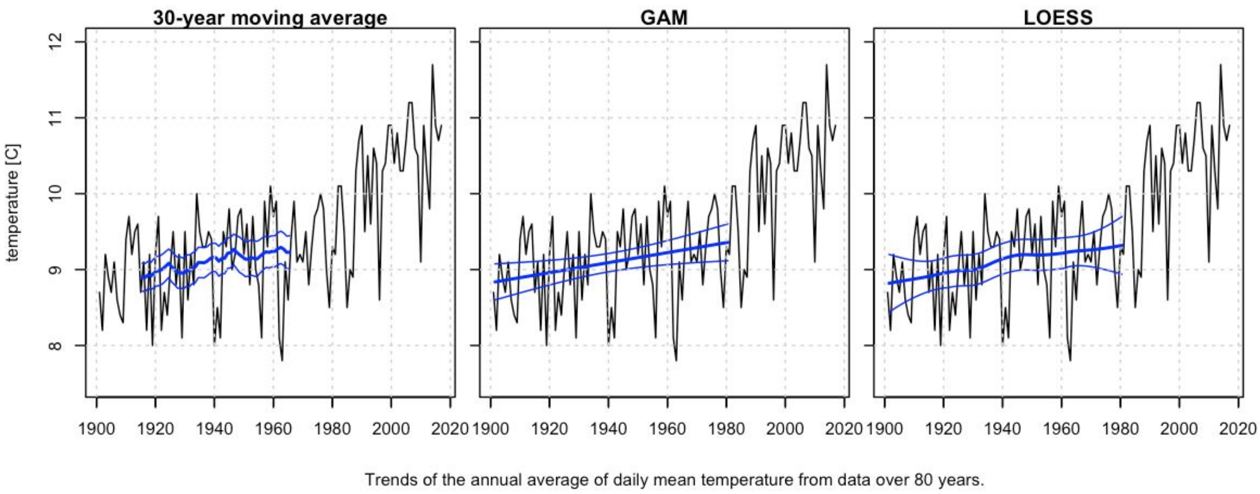

Figure 2: Same, but for the first 80 years.

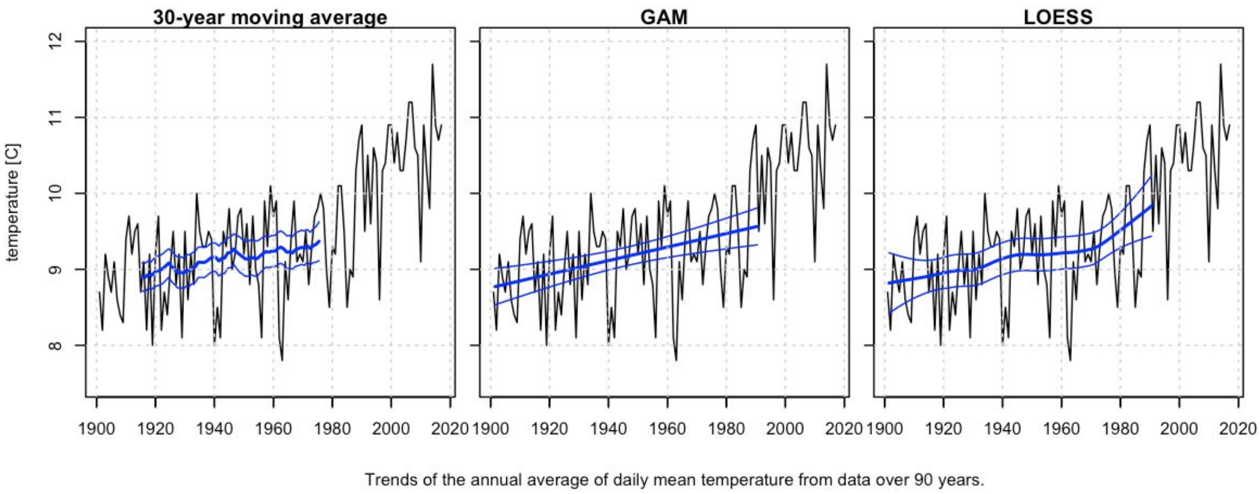

Figure 3: Same, but for the first 90 years.

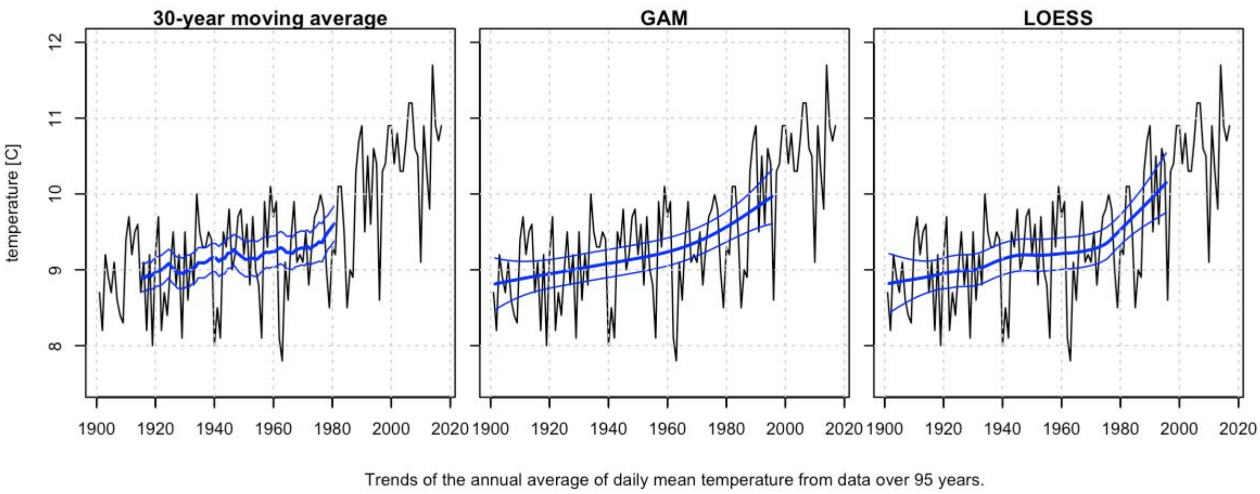

Figure 4: Same, but for the first 95 years.

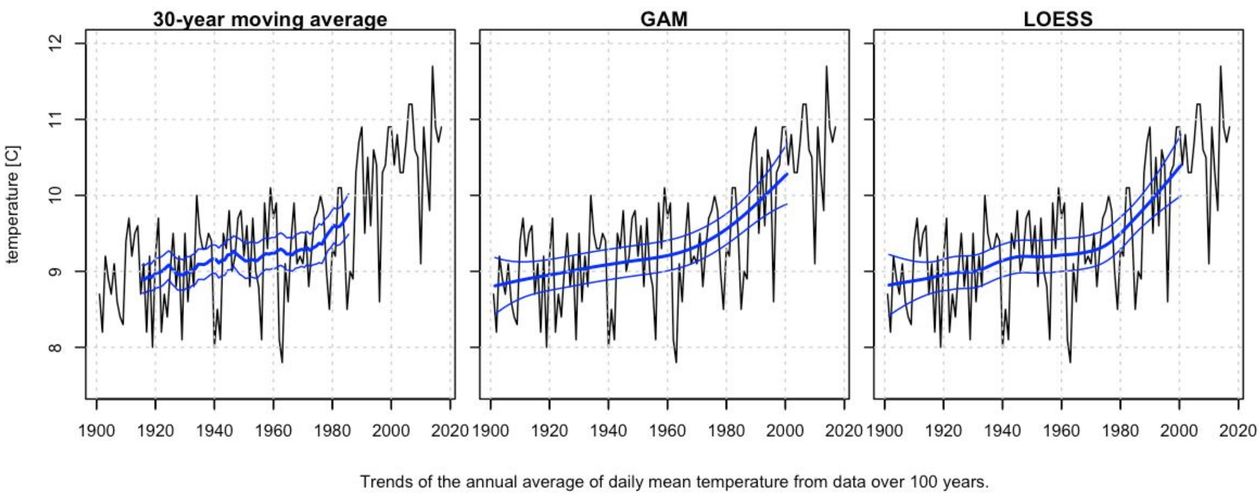

Figure 5: Same, but for the first 100 years.

Below, the same results are shown for the annual mean of daily minimum temperature.

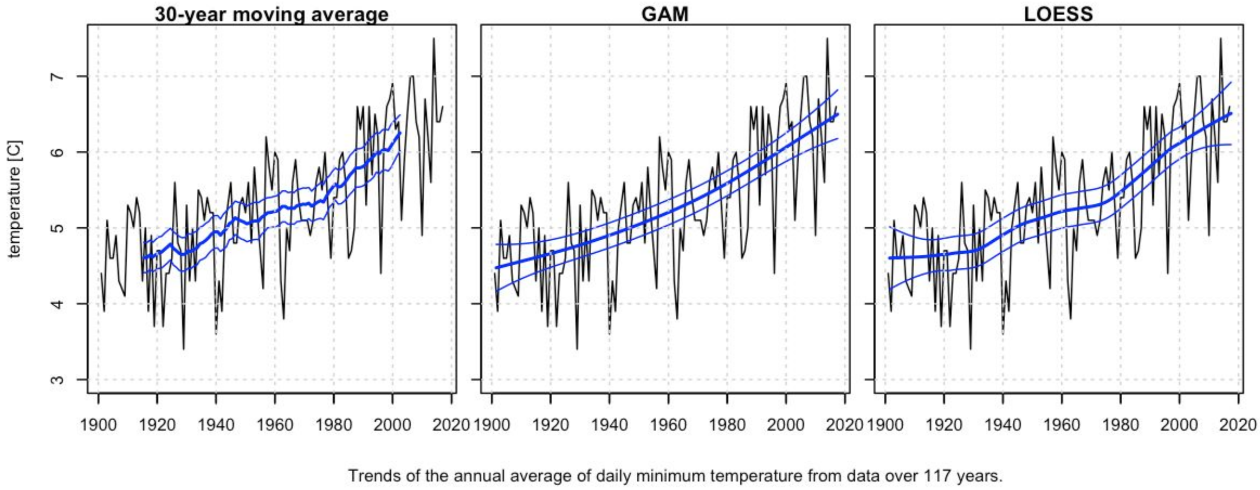

Figure 6: trendlines for annual mean of daily minimum temperature at de Bilt: moving averages (left), GAM smooth (centre), and LOESS smooth with 95% confidence interval (right).

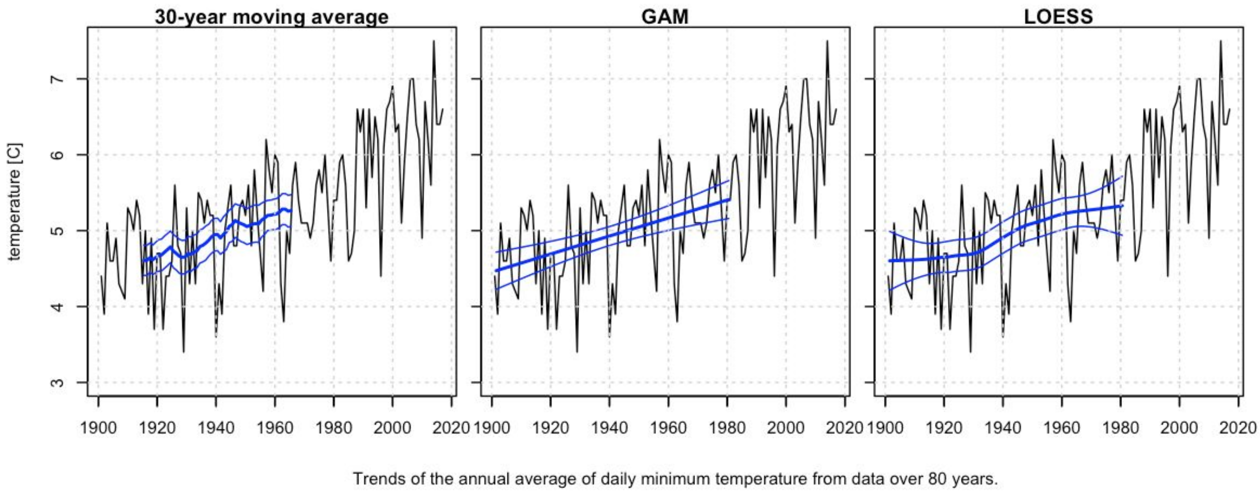

Figure 7: Same, but for the first 80 years.

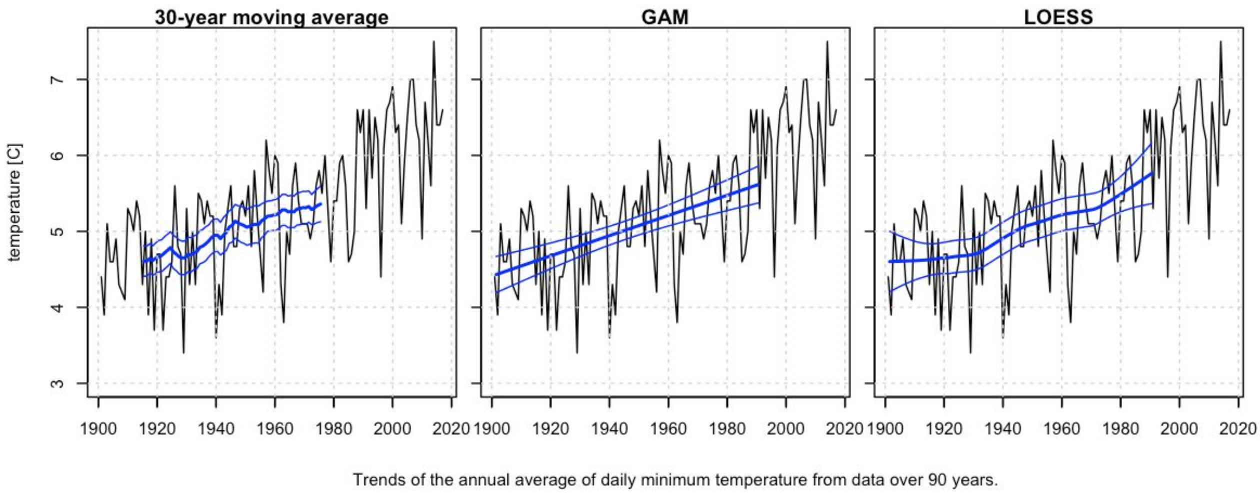

Figure 8: Same, but for the first 90 years.

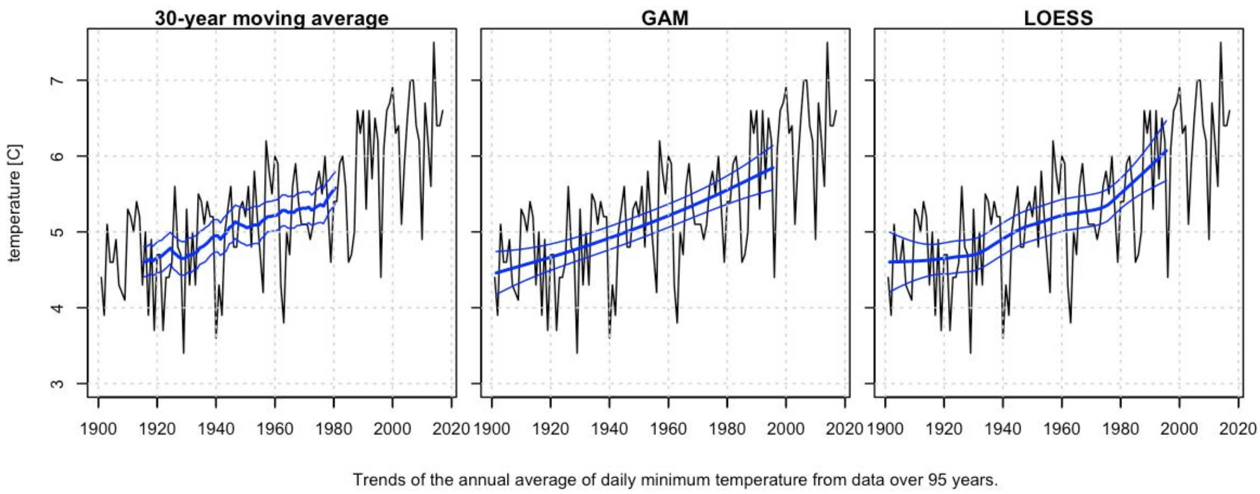

Figure 9: Same, but for the first 95 years.

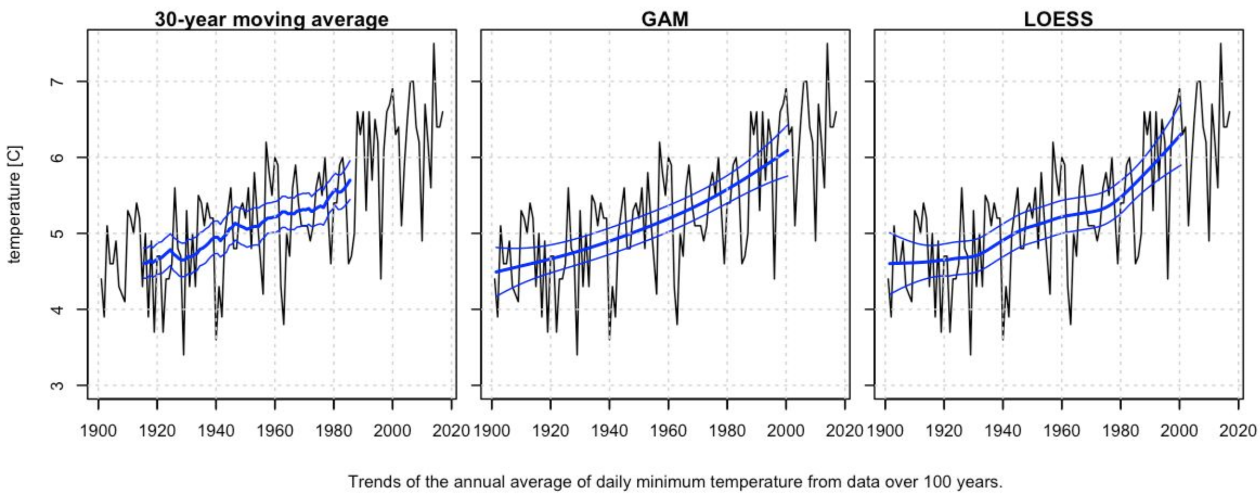

Figure 10: Same, but for the first 100 years.

The following results are for the annual mean of daily maximum temperature:

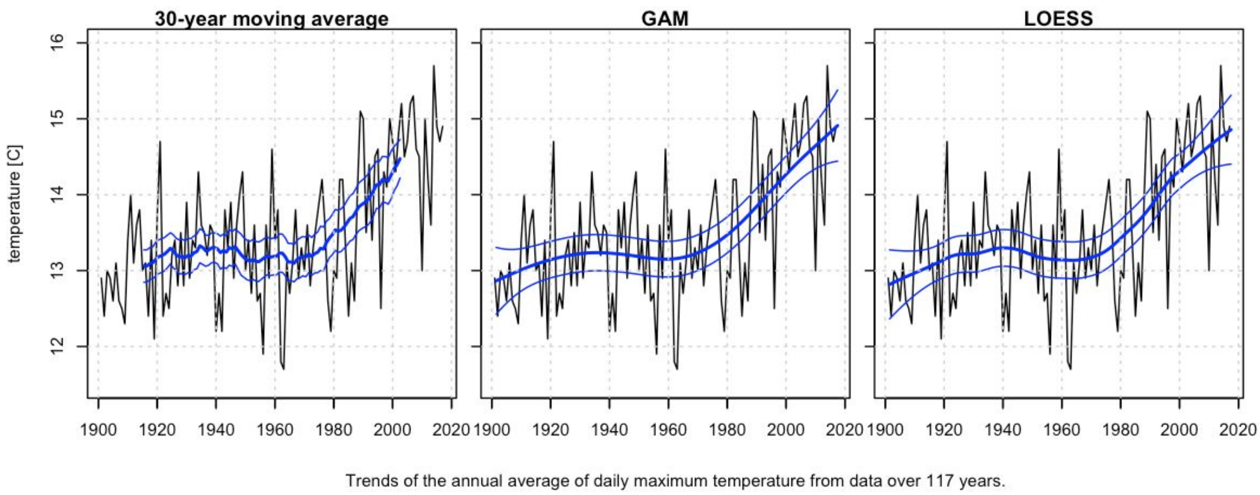

Figure 11: annual means of daily maximum temperature at de Bilt: moving averages (left), GAM smooth (centre), and LOESS smooth with 95% confidence interval (right).

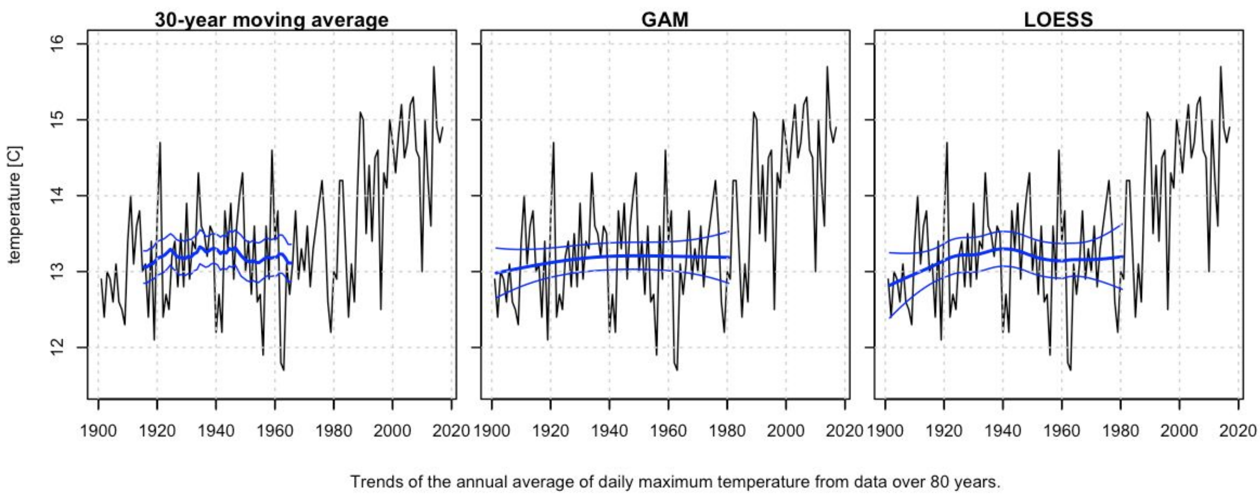

Figure 12: Same, but for the first 80 years.

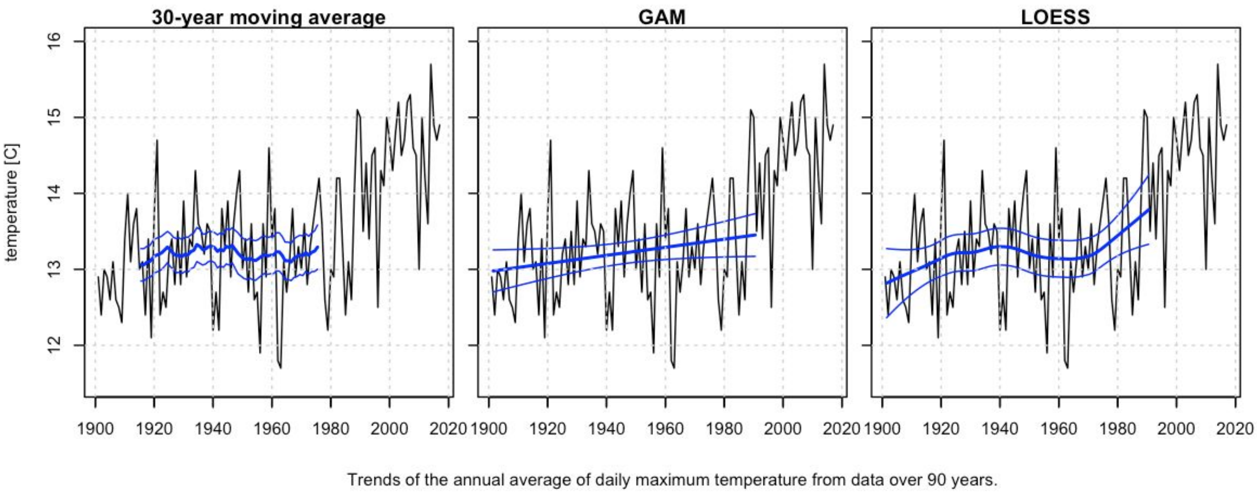

Figure 13: Same, but for the first 90 years.

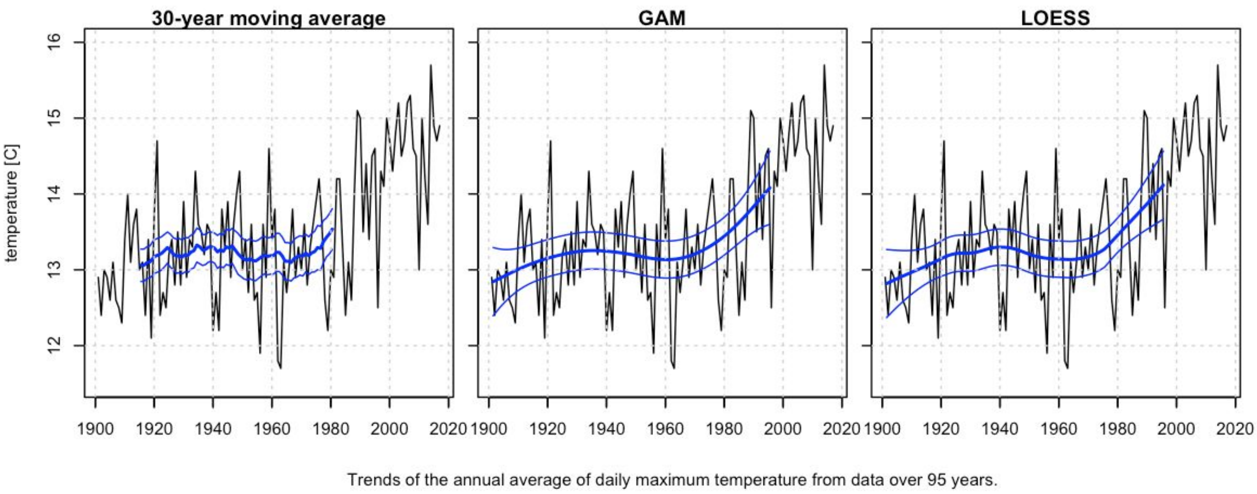

Figure 14: Same, but for the first 95 years.

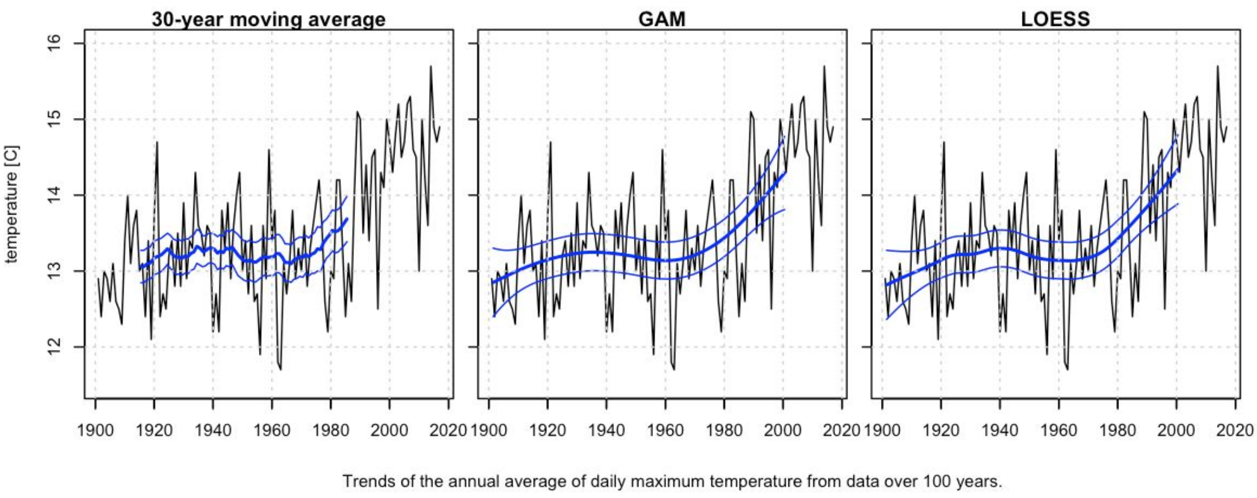

Figure 15: Same, but for the first 100 years.

We can see the following:

1. The linear LOESS trendline matches the 30-year averages closely without the wiggles in the latter.
2. In some cases, GAM switches rather abruptly from a linear fit to a nonlinear fit if additional data are added: compare Figures 3 and 4 (centre) and Figures 13 and 14 (centre).
3. This is not the case with the LOESS smoother: the trendline adapts smoothly to new data, while up to 21 years before the last year of data used in the fit, the trendline remains fixed. This difference in behaviour is as expected: a linear LOESS trendline is much more localized in time than a GAM, so the trendline value is eventually fixed, unlike the GAM trendline.

5.2 Trendlines for the annual maximum of precipitation

Finally, for annual maxima of 24-hour precipitation at several randomly chosen ECA&D stations (see https://www.ecad.eu), trendline estimates computed using the same LOESS smoother (blue) are shown. Also shown are the 30-year averages (black points). Note how LOESS smoothes the wiggles in the 30-year averages.

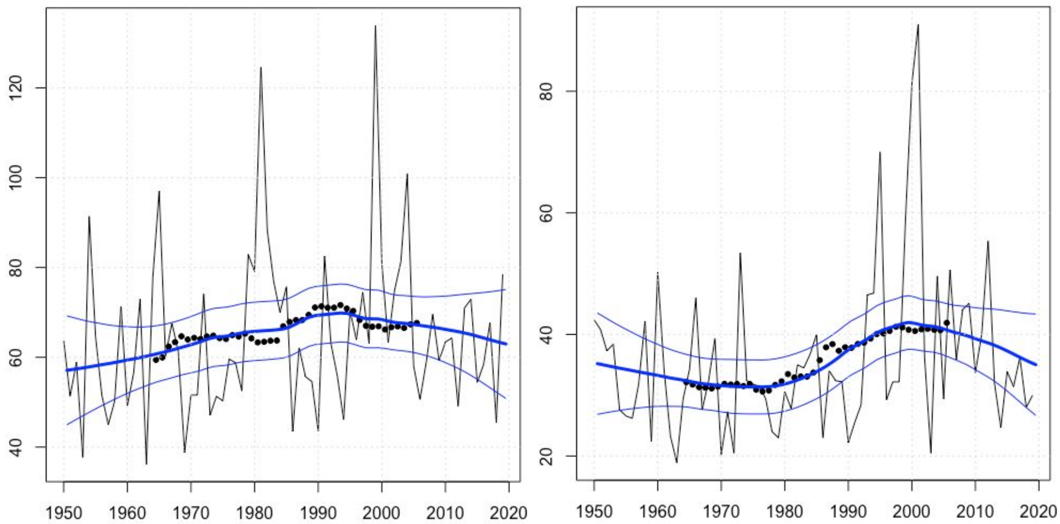

Figure 16: Trendlines of annual maxima of 24-hour precipitation (mm) at eight ECA&D stations: LOESS smooth with 95% confidence interval (blue) and running 30-year average (black dots) (continued on next page).

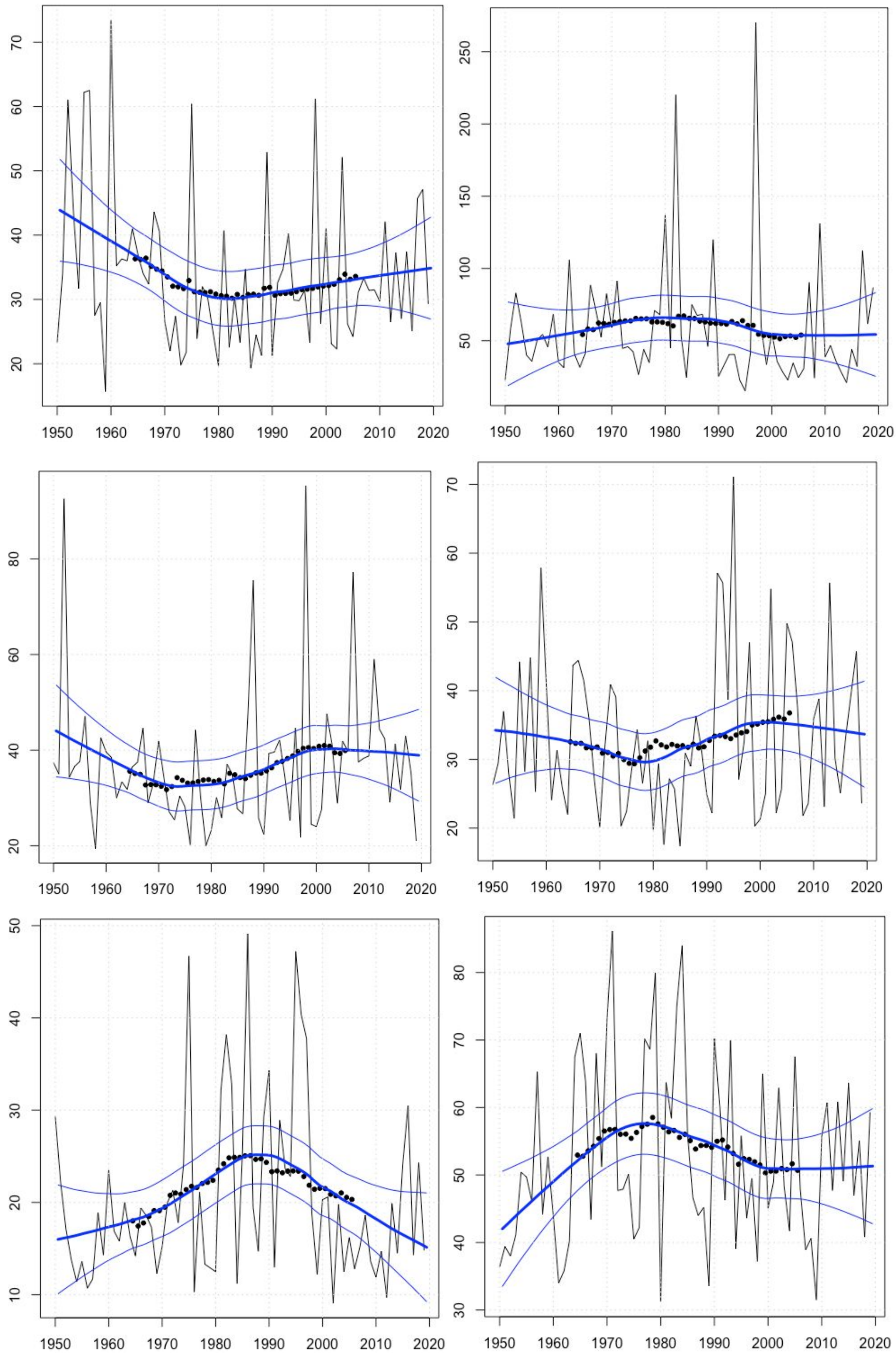

Figure 16: Continued from last page.

6. Code for standard trendline computation and testing

6.1 General

The standard trendline method is implemented in the R-function `climatrend.R`, but it can also be run directly from the linux or Windows command line, provided that R is installed on the system.

The code is distributed in the package “TrendKNMI” at GitLab at

```
https://gitlab.com/cees.de.valk/trend_knmi.git
```

You can simply use the download button (next to the blue Clone button) to download the code and unzip it into a folder (R users: don’t try to install it as a formal package, because it is not).

As an alternative, the version of November 2020 is printed in Section 6.4; the modules can be copied to text files which are given the proper file names and stored together in a folder. Further instructions follow below.

6.2 Run trendline computation in a linux terminal

R must be installed on your system. If you type `R` in a terminal, you should see information about the R version etc.; you can quit R by typing `q()`.

To compute the trendline, you run a command of the form

```
Rscript runtrend.R datafile option1 option2 ....
```

from the folder where de R code is stored. Detailed instructions on usage are found in the header of the file `runtrend.R` which can be read using a text editor or viewer; see Section 6.4.

The input data (containing years and annual values) needs to be stored in a .csv file. The output is in a .csv file with the same name as the input file, but with `trend_` attached to the front; it is saved to the folder containing the input file. If a plot is requested, a .png image is returned with the same name as the output .csv file.

6.3 Run trendline computation in a Windows command window

R must be installed on your system[4]. If installed, R should be present in the Windows start menu.

First you open a command window (by clicking on the start button and typing `cmd`). Then you navigate to the folder containing the code. It contains a batch file (`runtrend.bat`) which needs to be configured using a text editor in order to use it. Instructions are in the file header. Next, you run a command of the form

[4] For KNMI staff: if R is not installed, SSC Campus may install it on your request.

```
runtrend.bat datafile option1 option2 ....
```

to compute a trendline. Detailed instructions on usage are found in the header of the file `runtrend.R` which can be read using a text editor or viewer; see Section 6.4.

The input data (containing years and annual values) needs to be stored in a .csv file. The output is in a .csv file with the same name as the input file, but with `trend_` attached to the front; it is saved to the folder containing the input file. If a plot is requested, a .png image is returned with the same name as the output .csv file.

6.4 Run in an R environment

The function `climatrend.R` can be used within an R environment where the data are already present. See the header for further details of its usage.

For the application using file I/O, see under 6.2 or 6.3 above. In fact, `runtrend.R` can also be used in the R console, but it needs to be adjusted (you need to comment out the line containing `arg = commandArgs(trailingOnly= TRUE)` and write the filename and options to a string vector called `arg` before running `runtrend.R`).

6.5 Code of the version of November 2020

To use the code printed below, copy it into a text file and save it as `climatrend.R`. It is recommended to download the code from `gitlab.com/cees.de.valk/trend_knmi.git` instead.

```
#' @name  climatrend
#'
#' @title climatrend
#'
#' @description Fit a trendline to an annually sampled time-series by local linear regression
(LOESS)
#'
#' @param t years, increasing by 1 (double(n))
#' @param y annual values; missing values as blanks are allowed near beginning and end
(double(n))
#' @param p (optional) confidence level for error bounds (default: 0.95)) (double)
#' @param t1 (optional) first year for which trendline value is compared in test (double)
#' @param t2 (optional) second year (see t1)
#' @param ybounds (optional) lower/upper bound on value range of y (default: c(-Inf, Inf))
(double(2))
#' @param drawplot (optional): FALSE/TRUE or label on y-axis (default: FALSE) (logical or
character)
#' @param draw30 (optional): add 30-year moving averages to plot (default: FALSE) (logical).
#'
#' @usage Value <- climatrend(t, y, p, t1, t2, ybounds, drawplot, draw30)
#'
#' @return A list, with members:
#'   \item{t}{years}
#'   \item{trend}{trendline in y for years in t}
#'   \item{p}{confidence level}
```

```
#'   \item{trendubound}{lower confidence limit}
#'   \item{trendlbound}{upper confidence limit}
#'   \item{averaget}{central value of t in a 30-year interval}
#'   \item{averagey}{30-year average of y}
#'   \item{averageysd}{standard deviation of averaget}
#'   \item{t1}{first year for which trendline value is compared in test}
#'   \item{t2}{second year for which trendline value is compared in test}
#'   \item{pvalue of test of no longterm change}
#'
#' @details
#'   The trendline can be regarded as an approximation of a 30-year average, which has a smooth
#'   appearance and is extended toward the beginning and end of the time-series.
#'
#'   It is based on linear local regression, computed using the R-function loess.
#'   It uses a bicubic weight function over a 42-year window. In the central part
#'   of the time-series, the variance of the trendline estimate is approximately equal to
#'   the variance of a 30-year average.
#'
#'   To test the proposition of no long-term change between the years t1 and t2, these years
#'   need to be supplied. The result is the p-value: the probability (under the proposition)
#'   that the estimated trendline values in t2 and t1 differ more than observed.
#'
#'   version: 26-Nov-2020
#'
#' @references
#' KNMI Technical report TR-389 (see http://bibliotheek.knmi.nl/knmipubTR/TR389.pdf)
#'
#' @author Cees de Valk \email{cees.de.valk@knmi.nl}
#'
#' @export
climatrend <- function(t, y, p, t1, t2, ybounds, drawplot, draw30)
{
  # fixed parameters
  width <- 42
  control <- loess.control(surface = "direct", statistics= "exact",
                          iterations= 1)
  # check input
  # t and y
  if (missing(t) | missing(y)) {
    list()
    stop("t or y missing.")
  }
  if (length(t)< 3 | length(y)!= length(t)) {
    list()
    stop("t and y arrays must have equal lengths greater than 2.")
  }
  # check input
  if (any(is.na(t)) | sum(!is.na(y))< 3) {
    list()
    stop("t or y contain too many NA.")
  }

  # p
  if (missing(p)) {
    p <- NA
  }
  if (length(p)!= 1) {
    p <- NA
  }
```

```
if (is.na(p) | !(p> 0 & p< 1)) {
  p <- 0.95   # default confidence level
}
# t1 and t2
if (missing(t1) | missing(t2)) {
  t1 <- Inf; t2 <- -Inf
}
if (length(t1)!= 1 | length(t2)!= 1) {
  t1 <- Inf; t2 <- -Inf
}
if (is.na(t1) | is.na(t2)) {
  t1 <- Inf; t2 <- -Inf
}
# ybounds
ybounds0 <- c(-Inf, Inf)
if (missing(ybounds)) {
  ybounds <- ybounds0
}
if (length(ybounds)!= 2) {
  ybounds <- ybounds0
}
id <- is.na(ybounds)
ybounds[id] <- ybounds0[id]
ybounds <- sort(ybounds)

# drawplot
ylab= ""
if (missing(drawplot)) {
  drawplot <- FALSE
}
drawplot <- drawplot[1]
if (is.character(drawplot)) {
  ylab <- drawplot
} else if (is.na(drawplot) | !is.logical(drawplot)) {
  drawplot <- FALSE
}

# draw30
if (missing(draw30)) {
  draw30 <- FALSE
}
draw30 <- draw30[1]
if (!is.logical(draw30)) {
  draw30 <- FALSE
}

# dimensions etc.
t <- as.vector(t, mode= "double")
y <- as.vector(y, mode= "double")
dt <- diff(t[1:2])
n <- length(y)
ig <- !is.na(y)
yg <- y[ig]
tg <- t[ig]
ng <- sum(ig)
if (any(diff(tg)!= dt)) {
  error("NA may only occur in y only at beginning and/or end of time-series")
}
```

```
# check values of bounds
if (any(yg< ybounds[1]) | any(yg> ybounds[2])) {
  stop("Stated bounds are not correct: y takes values beyond bounds.")
}

# averages over 30 time-steps
avt <- avy <- avysd <- NULL
if (ng> 29) {
  avt <- tg+dt/2  # time (end of time-step, for 30-year averages)
  avy <- filter(yg, rep(1, 30)/30, method = "convolution")
  avy2 <- filter(yg^2, rep(1, 30)/30, method = "convolution")
  avysd <- sqrt(avy2-avy^2)
  ind <- 15:(ng-15)
  avt <- avt[ind]
  avy <- avy[ind]
  avysd <- avysd[ind]
}

# linear LOESS trendline computation
span <- width/ng
mdl <- loess(y ~ t, data= data.frame(t= tg, y=yg), span= span,
             degree= 1, control= control)
# mdl <- loess(y ~ t, data= data.frame(t= t, y=y), span= span, degree= 1)
pre <- predict(mdl, newdata= data.frame(t= t), se= TRUE)
trend <- pre$fit          # trendline
trendsd <- pre$se.fit     # standard deviation of trendline

# confidence limits (normal approximation)
trendub=  trend + trendsd*qnorm(1-(1-p)/2)
trendlb=  trend - trendsd*qnorm(1-(1-p)/2)

# apply bounds
trend <- pmin(trend, ybounds[2])
trend <- pmax(trend, ybounds[1])
trendub <- pmin(trendub, ybounds[2])
trendub <- pmax(trendub, ybounds[1])
trendlb <- pmin(trendlb, ybounds[2])
trendlb <- pmax(trendlb, ybounds[1])

# pvalue for trend
pvalue <- NULL
if (t2 %in% t & t1 %in% t & t2>= t1+30) {
  y1 <- trend[t1== t]
  y2 <- trend[t2== t]
  y1sd <- trendsd[t1== t]
  y2sd <- trendsd[t2== t]
  # two-sided test for absence of trend
  pvalue <- (1-pnorm(abs(y2-y1)/sqrt(y1sd^2+y2sd^2)))*2
} else if (t2> -Inf & t1< Inf) {
  warning("no p-value: t1 or t2 not in t, or t2 too close to t1.")
} else {
  t1 <- NULL
  t2 <- NULL
}

# plotting
if (!(drawplot== FALSE)) {
  par(pty= "s", cex= 1)
  ylim <- c(min(pmin(y, trendlb), na.rm = TRUE), max(pmax(y, trendub), na.rm = TRUE))
```

```
    ylim[2] <- ylim[1] + (ylim[2]-ylim[1])*1.0
    plot(t, y, type= "l", xlab= "", ylab= ylab, ylim= ylim)
    grid()
    if (draw30== TRUE) {
      points(avt, avy, pch= 18, cex= 0.5, col= "black")
    }
    points(t, y, pch= 15, cex= 0.75)
    lines(t, trend, lwd= 2)
    lines(t, trendub, lwd= 1, lty= 2)
    lines(t, trendlb, lwd= 1, lty= 2)
    legendpos <- "topleft"
    tr <- y[!is.na(y)]
    lr <- length(tr)
    ir <- 1:(lr/2)
    id <- which.max(c(mean(tr[ir])-ylim[1], mean(tr[lr+1-ir])-ylim[1],
                      ylim[2]-mean(tr[ir]), ylim[2]-mean(tr[lr+1-ir])))
    pid <- c(min(tr[ir])-ylim[1], min(tr[lr+1-ir])-ylim[1],
         ylim[2]-max(tr[ir]), ylim[2]-max(tr[lr+1-ir]))
    id <- which.max(pid)
    legendpos <- c("bottomleft", "bottomright", "topleft", "topright")[id]
    if (draw30== TRUE) {
      legend(legendpos, c("trendline", paste("its ", 100*p, "% confid.", sep=""), "30-yr
average"),
             lwd= c(2, 1, NA), lty= c(1, 2, NA), pch= c(NA, NA, 18), cex= 0.75, bty= "o",
             y.intersp= 0.8)
    } else {
      legend(legendpos, c("trendline", paste("its ", 100*p, "% confid.", sep="")),
              lwd= c(2, 1), lty= c(1, 2), cex= 0.75, bty= "o",
              y.intersp= 0.8)
      }
  }
  list(t= t, trend= trend, p= p, trendubound= trendub, trendlbound= trendlb,
       averaget= avt, averagey= avy, averageysd= avysd,
       t1= t1, t2= t2, pvalue= pvalue, ybounds= ybounds)
}
```

To use the code printed below, copy it into a text file and save it as `runtrend.R`. It is recommended to download the code from `gitlab.com/cees.de.valk/trend_knmi.git` instead.

```
#
# purpose: Compute KNMI standard trendline and optionally, test for absence of long-term change
#          (trend) between two specified years t1 and t2 (implemented in climatrend.R), by a
#          call from a terminal or Windows command prompt.
#
# usage:   Rscript runtrend.R datafile option1 option2 ....
#
#          datafile: name of .csv file containing
#               - an optional line with the options: option1 option2 .... (if not provided
#                 as arguments)
#               - a line with the column names for year and variable (separated by comma)
#               - lines containing year and annual value (separated by comma; containing at
#                 least 3 years)
#
#          option1, option2, etc.: parameter assignment of the form
#
#                                    parameter=value
#
#          (without spaces). For example: p=0.95 t1=1962 t2=2018 drawplot=TRUE.
#          WARNING: for options entered on the command line, if value is itself a string, then
#          put the entire assignment between quotes which differ from those used for the value,
#          as in 'drawplot="temperature (C)"' (do not do this with options in the data file).
#
#          All parameters are optional. They are (see also header of climatrend.R):
#               - p: confidence level for confidence interval (default: 0.95)) (double)
#               - t1: first year of trendline value used in testing (double) (default: no test)
#               - t2: second year of trendline value used in testing (double) (default: no test)
#               - lbound: lower bound on value range of variable (default: -Inf) (double)
#               - ubound: upper bound on value range of variable (default: Inf) (double)
#               - drawplot: FALSE/TRUE or text on y-axis (default: FALSE) (logical or character)
#               - draw30: add 30-year moving averages to plot (default: FALSE) (logical)
#
#          Example 1 of content of datafile: (parameter assignments in file)
#          p=0.95 drawplot="TG (C)"
#          year, TM
#          1990, 12.6
#          1991, 11.2
#          1992, 12.0
#
#          Example 2 of content of datafile: (no parameter assignments in file)
#          year, TM
#          1990, 12.6
#          1991, 11.2
#          1992, 12.0
#
# output:  A .csv file named "trend_"filename, containing five columns:
#          year, annual value, trend, trendlowerbound, trendupperbound, 30-yr average
#
#          The trendline is in the column labelled "trend"; its 100p%-confidence interval
#          is bounded by trendlowerbound and trendupperbound.
#          The 30-year average is given AT THE LAST YEAR of the 30-year interval.
#
# warning: The R function climatrend.R is called. If this file is in another directory
```

```
#          then the current one, then the sourcepath in the code below needs to be
#          changed to the path to climatrend.R.
#
# version: 26-Nov-2020
#
# reference: KNMI Technical report TR-389 (see http://bibliotheek.knmi.nl/knmipubTR/TR389.pdf)
#
# author: Cees de Valk \email{cees.de.valk@knmi.nl}
#

# set here the path to the file climatrend.R
# e.g. for the current directory:
# sourcepath <- ""
sourcepath <- ""  # change if needed! path to input and output file

# source code
source(paste(sourcepath, "climatrend.R", sep= ""))

# arguments
arg = commandArgs(trailingOnly= TRUE)
print(arg)

# the first argument is the name the input file (otherwise,
# a default filename is used)
la <- length(arg)
if (la> 0) {
  fname <- arg[1]
} else {
  stop("No name supplied for the datafile.")
}

sk <- 0
if (la> 1) {
  # take parameter assignments from arguments
  parama <- arg[2:la]
  la <- length(parama)
  temp <- readLines(fname, n= 1)
  if (grepl("=", temp)) {
    sk <- 1
  }
} else {
  # read parameter assignments from file
  # parama <- scan(fname, nlines= 1, what= "character", sep= " ")
  temp <- readLines(fname, n= 1)
  temp <- gsub("'","\"", temp)
  parama <- strsplit(temp, "\\s(?=(?:[^\"]*\"[^\"]*\")*[^\"]*$)", perl= TRUE)[[1]]
  la <- length(parama)
  sk <- 1
}
print(parama)

# default parameter values, overwritten if specified
p= 0.95
t1 <- t2 <- NULL
ubound <- Inf
lbound <- -Inf
drawplot <- FALSE
draw30 <- FALSE
if (la> 0) {
```

```
  for (i in 1:la) {
    temp1 <- tryCatch({eval(parse(text= parama[i]))}, error= function(err){NA})
  }
}
ybounds <- c(lbound, ubound)

# read data of annual values
Data <- read.csv(fname, skip= sk)
colData <- colnames(Data)

# change filename for purpose of output
s <- .Platform$file.sep
path <- strsplit(fname, s)[[1]]
lp <- length(path)
fname <-  paste("trend_", path[lp], sep= "")
if (lp> 1) {
  fname <- paste(paste(path[1:(lp-1)],collapse= s), fname, sep= s)
}

# prepare plotting to png file
if (drawplot== TRUE) {
  drawplot <- colData[2] # get y-axis label for plot
}

# convert to plain matrix and remove NA's
Data <- data.matrix(Data)
dd <- dim(Data)
if (dd[2]!= 2) {
  stop("Data in datafile not properly formatted.")
} else if (dd[1]< 3) {
  stop("Time series is too short (must have at least 3 years).")
}

# extract valid data
# If Data[j, 1] is good but Data[j, 2] is NA,then the trend will be
# extrapolated/interpolated to year Data[j, 1]
# idgood <- !apply(is.na(Data), 1, any)
# idgood <- !is.na(Data[, 1])
# t <- Data[idgood, 1]
# y <- Data[idgood, 2]
t <- Data[, 1]
y <- Data[, 2]

if (!(drawplot== FALSE)) {    # open plotfile
  plotfname <- sub("csv", "png", fname)
  fac= 1.25
  png(plotfname, units="in", width=6*fac, height=5*fac, res=200)
}

# trend computation/plotting
temp <- climatrend(t, y, p, t1, t2, ybounds, drawplot, draw30)

if (!(drawplot== FALSE)) {    # close plotfile
  dev.off()
}

# prepare/format output
Data <- cbind(Data, array(NA, dim= c(dd[1], 4)))
# Data[idgood, 2] <- temp$trend
```

```
# Data[idgood, 3] <- temp$trendlbound
# Data[idgood, 4] <- temp$trendubound
Data[, 3] <- temp$trend
Data[, 4] <- temp$trendlbound
Data[, 5] <- temp$trendubound
if (length(temp$averaget)> 0) {
  endyear <- ceiling(temp$averaget)+14
  id <- Data[, 1] %in% endyear
  Data[id, 6] <- temp$averagey
}
# Data[, 3:6] <- signif(Data[, 3:6], 4)

colnames(Data) <- c(colData, "trend", "trendlowerbound", "trendupperbound", "30-yr average")

pstring <- " "
if (length(temp$pvalue)> 0) {
  pstring <- paste("In the absence of a long-term trend between ", t1, " and ", t2,
              ", the size of the observed difference in trend values would be exceeded",
              " with a probability (p-value) of:", sep= "")

}
paramstring <- "The probability of the interval between the confidence bounds is "

# output to .csv file with information in header
a <- file.create(fname, overwrite= FALSE)
cat(pstring, "\n", temp$pvalue, "\n", paramstring, paste(p), ".\n", sep= "",
    file= fname, append= TRUE)
defaultW <- getOption("warn")
options(warn = -1)
write.table(Data, file = fname, append= TRUE, row.names= FALSE, col.names= TRUE,
            na= " ", sep= ",")
options(warn = defaultW)
```

To use the code printed below, copy it into a text file and save it as `runtrend.bat`. It is recommended to download the code from `gitlab.com/cees.de.valk/trend_knmi.git` instead.

```
rem
rem   To configure this runtrend.bat for your environment, do the following.
rem
rem   Look first for the path to Rscript.exe: in cmd window, go to C:\Users, and
rem   type:   where /r . Rscript.exe
rem
rem   Then copy one of the results, and paste it between the quotes in the command
rem   line at the end of this file.
rem
rem   Now you can run runtrend.bat with the same arguments as used for runtrend.R
rem   (see header of runtrend.R).
rem
rem   version: 26-Nov-2020
rem
rem   author: Cees de Valk (cees.de.valk@knmi.nl)
rem
rem   reference: KNMI Technical report TR-389 (see
http://bibliotheek.knmi.nl/knmipubTR/TR389.pdf)
rem
"C:\Users\All
Users\App-V\2C43EEE5-D738-431B-80BC-A8EC97345439\BE788D1A-CBF7-4AD7-8FCD-C84F21CE2D15\Root\R-4.0.
2\bin\Rscript.exe" runtrend.R %1 %2 %3 %4 %5 %6 %7
```

References

Cleveland, W.S., Devlin, S.J. (1988), Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting. Journal of the American Statistical Association 83:403, 596-610.

Loader, C. (1999), Local regression and likelihood. Springer.

Wahba, G. (1990), Spline models for observational data. SIAM.

Wood, S.N. (2011), Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. Journal of the Royal Statistical Society (B) 73, 3-36.

Royal Netherlands Meteorological Institute

PO Box 201 | NL-3730 AE De Bilt
Netherlands | www.knmi.nl