



Royal Netherlands
Meteorological Institute
*Ministry of Infrastructure and the
Environment*

Time series transformation tool version 3.1

Description of the program to generate time series consistent with the KNMI'14 climate scenarios

A. Bakker

De Bilt, 2015 | Technical report; TR-349

Time series transformation tool version 3.1

Description of the program
to generate time series
consistent with
the KNMI'14 climate scenarios

Alexander Bakker

November 17, 2015

Colophon

Title	Time series transformation tool version 3.1: Description of the program to generate time series consistent with the KNMI'14 climate scenarios
Project number	KNMI'14
Contact person	Janette Bessembinder (bessembi@knmi.nl) KNMI Utrechtseweg 297 3731 GA De Bilt Postbus 201 3730 AE De Bilt
Authors	Alexander Bakker

Acknowledgements

I want to thank Janette Bessembinder and Jules Beersma for their comments on early versions of this technical report.

Contents

1 Introduction	4
2 Precipitation	6
2.1 Changes in the KNMI'14 climate scenarios	6
2.2 Transformation procedure	6
3 Temperature	10
3.1 Changes in the KNMI'14 climate scenarios	10
3.2 Transformation procedure	11
4 Global radiation	13
4.1 Changes in the KNMI'14 climate scenarios	13
4.2 Transformation procedure	13
5 Perspectives on other climate variables	15
5.1 Reference crop evapotranspiration	15
5.2 Relative humidity	15
5.3 Wind	15
References	16
Appendices	17
A R-scripts for transformation program Precipitation	18
B R-scripts for transformation program Temperature	28
C R-scripts for transformation program Temperature	35

1. Introduction

In May 2014, KNMI presented the KNMI'14 climate change scenarios: four new scenarios for future climate change in the Netherlands (KNMI 2014).

"Each scenario provides a consistent picture of the changes in 12 climate variables, including temperature, precipitation, sea level, and wind. Not only the changes in the mean climate are depicted, but also changes in the extremes such as the coldest winter day and the maximum hourly precipitation per year. The changes are for two different time horizons: around 2050 and around 2085, relative to the reference period of 1981-2010, published in the KNMI climate atlas (Sluijter et al. 2011). The KNMI'14 scenarios are the four combinations of two possible values for the global temperature increase, 'Moderate' and 'Warm', and two possible changes in the air circulation pattern, 'Low value' and 'High value' (see figure 1.1). Together they span the likely changes in the climate of the Netherlands according to the newest insights. By providing these KNMI'14 scenarios KNMI offers a guide for evaluating the consequences of climate change and for developing options and strategies for climate adaptation. They will enable users to include climate change when making decisions to ensure that the Netherlands will have a safe and sustainable future."

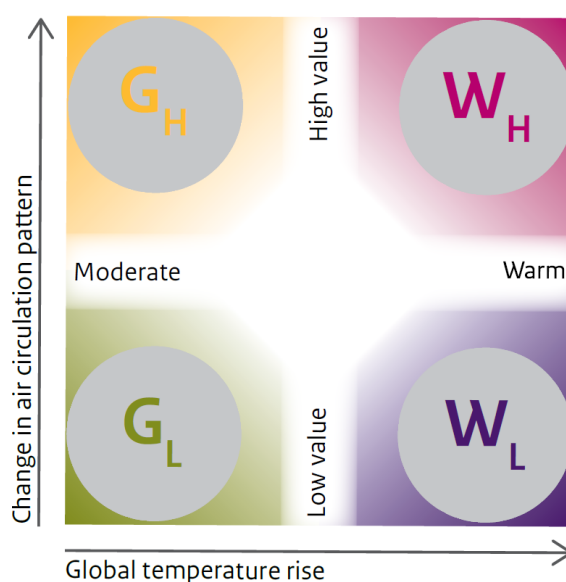


Figure 1.1: Overview of drivers of the four KNMI'14 climate change scenarios for the Netherlands.

"The KNMI'14 climate scenarios translate the global findings in the IPCC 2013 report (IPCC 2013) to the situation in the Netherlands. The IPCC report does not provide results for individual countries. The climate scenarios for the Netherlands are based on observational evidence and the latest climate model calculations performed for the IPCC, augmented with calculations performed using the KNMI climate model for Europe."

The output of the climate models cannot be directly used to calculate changes in climate indices nor to drive impact models, since they contain substantial biases. For this, the 'transformation program' (or 'advanced delta method') developed for the previous climate change scenarios KNMI'06 (Bakker and Bessembinder 2012) has been updated and extended (version 3.1). Basically, the program *transforms* meteorological time series by applying monthly 'change factors' or 'deltas'. In this way, future time series are constructed that match the KNMI'14 climate change scenarios.

This report presents the 'Transformation Program version 3.1 KNMI'14'^{1,2}. Chapter 2 to 4 are devoted to the transformation of a particular climate variable. Chapter 2 deals with precipitation, chapter 3 with temperature and chapter 4 with global radiation. These chapters first briefly introduce the deltas and subsequently present the actual transformation. Finally, chapter 5 comments on reference evapotranspiration, relative humidity and wind.

¹The report is restricted to the transformation program itself (code included in appendices). For information on the deltas and reference data used is referred to Van den Hurk et al. (2014). Coefficients for the time horizons 2030, 2050 and 2085 compared to 1981-2010 are available in the "Toelichting transformatieprogramma" (http://www.klimaatsscenarios.nl/toekomstig_weer/transformatie/index.html).

²Note that the official version (based on R, presented in the appendices) and the online version (based on Fortran) may show small differences.

2. Precipitation

2.1 Changes in the KNMI'14 climate scenarios

Apart from the G_H and W_H summer scenarios³, KNMI'14 projects an increase in precipitation associated with the higher temperatures (KNMI 2014). In contrast, the H-scenarios show a decrease in summer precipitation as a result of the more frequent dry easterlies.

The scenarios distinguish 'wet' and 'dry' days, dependent on the (non)-exceedance of wet-day threshold $th_{wet} = 0.1$ [mm]. For the construction of future time series that match the KNMI'14 scenarios, monthly *relative* changes in wet-day frequency ΔF [%], mean precipitation on wet days ΔP_{av} [mm] and the 99th percentile of all wet day amounts ΔP_{99} [mm] (see figure 2.1) are provided for around 2050 and 2085 compared to 1981-2010⁴.

The response of local (wet-day) precipitation extremes is very uncertain and hardly depends on the circulation changes (Van den Hurk et al. 2014, pp. 43-47). In order to capture this additional uncertainty, a lower and upper limit are provided for ΔP_{99} (see figure 2.1, right panels).

2.2 Transformation procedure

Future time series of daily precipitation sums P^f are obtained by the transformation of a historical reference dataset (usually derived from observations). For this, the relative changes are applied to the historical precipitation records by a two-step transformation procedure. First, the wet-day frequency F is adjusted and second, a power-law transformation is applied to perturb the distribution of wet-day amounts. This procedure is very similar to the one applied for KNMI'06 (Bakker and Bessembinder 2012).

Adjustment of the wet-day frequency

In case $\Delta F < 0$, $|\Delta F|$ of the wet days have to be 'dried' or 'removed' by setting P to zero. In case $\Delta F > 0$, $|\Delta F|$ dry days have to be 'wetted' or $|\Delta F|$ wet days have to be 'added'.

The adjustment of the wet-day frequency consists of two steps. First, for all twelve calendar months a set of target precipitation amounts/days P_{target} is defined that is indicative for the

³Actually, W_L projects slightly less summer precipitation around 2085 too, but not around 2050.

⁴In the online version for the time horizons in between 1981-2010 and 2050 and in between 2050 and 2085 linear interpolation of the relative changes is used (not in the R-version). Also a scenario of 2030 is given in the KNMI'14 climate scenarios (KNMI 2014, p. 19).

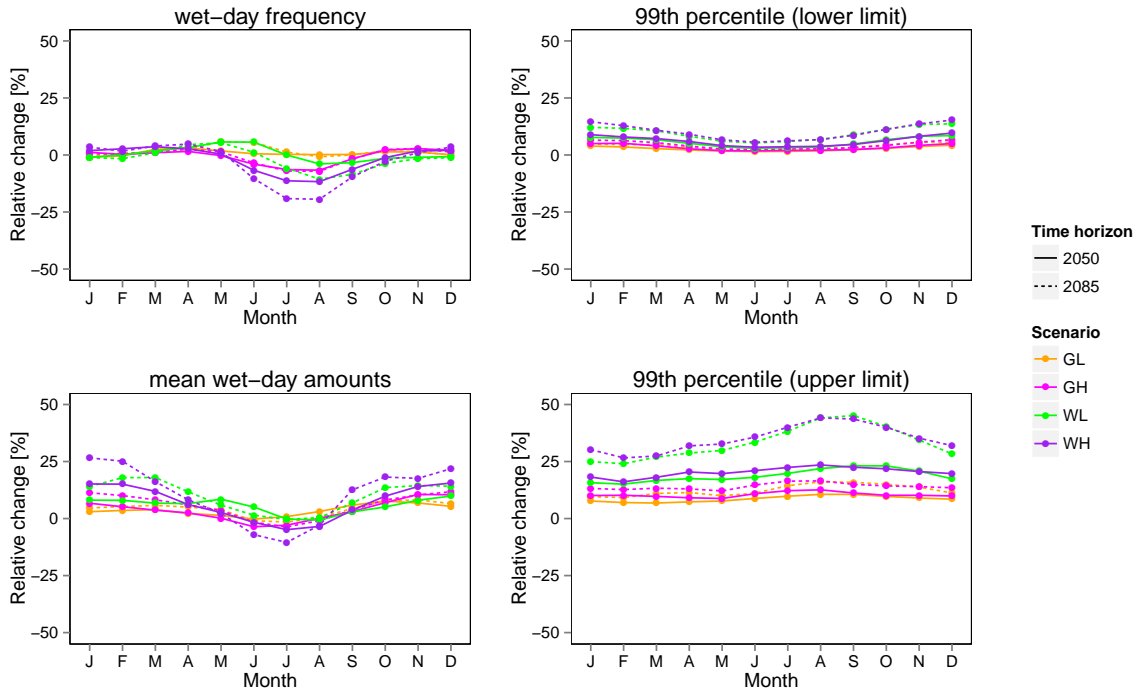


Figure 2.1: Projected relative changes per month according to KNMI'14 (around 2050 (continuous lines) and 2185 (dashed lines) relative to 1981-2010) for wet-day frequency (upper left panel), mean precipitation (upper right panel), 99th percentile wet-day precipitation amounts, lower limit (lower left panel) and upper limit (lower right panel) (Van den Hurk et al. 2014).

$n_{target} = |\Delta F| \cdot n_{wet}$ amounts that should be dried (or added). Second, the wet days are selected that should be dried (or dry days that should be added) .

Selection of 'target precipitation amounts' P_{target} is composed such that the distribution of wet-day amounts is hardly affected by the wet-day removal (or addition). This means that the Cumulative Distribution Function (CDF) of removed (added) precipitation amounts should also closely resemble the 'original' CDF of the wet-day amounts. Therefore, P_{target} is uniformly sampled from the 'quantile function' of the reference precipitation amounts. So, when 3 days have to be removed (or added) for a particular calendar month (say October), P_{target} is composed by values close to the 25%, 50% and 75% quantiles of all October wet-day amounts within the reference period.

In order to enable a non-random selection of wet-days to dry, P_{target} refers to a specific day (e.g. October 10th 1986) that represents a quantile rather than to the quantile itself (see next paragraph). This is done as follows. First all wet (e.g. October)-days are assigned a unique rank based on precipitation amount and in case of ties (i.e. equal values) based on chronology. So, a day with 3 mm gets a higher rank than a day with 2 mm and in case of equal amount a day further in the past gets the lower rank. Second, P_{target} is composed by uniformly sampling from the unique ranks. So, in case of 401 wet (e.g. October)-days of which 3 have to be dried, the days with rank numbers 101, 201 and 301 are selected.

'Removal of wet days' / 'Drying wet days' After composing P_{target} , a set of wet days is selected for drying of which the amounts resemble the target amounts P_{target} as closely as possible. In order to maintain a usual correlation structure wet-days are only selected for drying if the day before or after is already dry. This limits the amount of isolated wet (or dry) days within dry (or wet) spells after wet-day adjustment.

The selection procedure is as follows. First, a day is selected with an amount as close as possible to the target amount and in case of ties chronologically as close as possible. Second, it is checked if the day before or after is dry. If this criterion is satisfied, the specific day is dried. If not, the second closest day is checked and so on until a day satisfies the criterion and can be dried. Note, that the first trial always refers to the day the target amount is referring to.

For KNMI'14, first all target values of all twelve calendar months are determined and then all wet days are dried from low to high rank (so from low to high target value and in case of ties in chronological order (see previous paragraph)). For, the transformation tool of KNMI'06 (Bakker and Bessembinder 2012), this procedure was applied per calendar month. So, first all January days were dried, then the February days and so on. This small change in procedure is introduced to increase the probability to find a value close to the target value which otherwise appears hard for the higher values.

'Adding of wet days' / 'Wetting of dry days' After sampling P_{target} , dry days to be 'wetted' are selected on the basis of the number of preceding wet days.

This is done for every calendar month separately. For e.g. October, all October days (wet and dry) are stepped through chronologically (from the very first October day to the very last October day in the entire reference period). A wet day counter is increased by ΔF with each wet day. Each time a dry day is passed while the counter exceeds unity, this dry day is selected for 'wetting' (and the counter is decreased by 1). This procedure automatically ensures that every new wet day succeeds an already existing wet day (no isolated wet days are created). This ensures a better maintenance of the temporal correlation. Note that the procedure allows 'wetting' of two or more succeeding days if the preceding wet period is long enough (because the counter is not reset to zero, but decreased by 1 after wetting a dry day).

After selecting the dry days to 'wet', the target amounts are assigned to the selected dry days. This is done according to the amounts of the preceding wet day amount (in case of the first day of the month, the preceding day is the last day of the preceding month). So, the lowest target amount is assigned to the day with the lowest amount at the preceding day.

Adjustment of the wet-day amounts

In the second step, a power-law transformation is applied to the wet-day adjusted precipitation P^* .

$$P^f = \begin{cases} P^*, & \text{if } P^* < th_{wet} = 0.1 \\ \max(th_{wet}, a(P^*)^b), & \text{if } th_{wet} \leq P^* \leq P_{99} \\ (1 + \Delta P_{99}) P^*, & \text{if } P_{99} < P^* \end{cases} \quad (2.1)$$

where the coefficients a and b are iteratively estimated such that the wet-day mean P_{av} and the 99th percentile P_{99} of the original values P are perturbed nearly by the relative changes ΔP_{av} and ΔP_{99}

$$\begin{aligned} P_{av}^f &= (1 + \Delta P_{av})P_{av} \\ P_{99}^f &= (1 + \Delta P_{99})P_{99} \end{aligned} \tag{2.2}$$

Estimation of P_{99} The estimation of b requires a 'robust' estimate of P_{99} for each calendar month. The number of data for this estimation is usually limited (~ 450 wet days/calendar month for a 30-yr period, in the Netherlands). Therefore, P_{99} is estimated by multiplying P_{90} ⁵ by a coefficient $c = P_{99}/P_{90}$. Values per calendar month for this coefficient are approximately homogeneous for the Netherlands (i.e. no distinct spatial trends were identified, not shown) and are estimated by taking the median from 240 estimates from 240 precipitation stations (see figure 2.2). This method provides more robust estimates for P_{99} than estimating the P_{99} for every station separately.

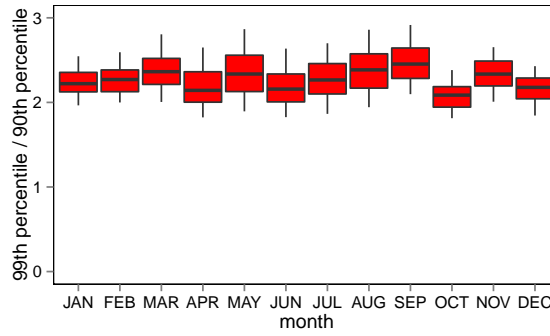


Figure 2.2: Boxplots of the ratio between P_{90} and P_{99} per calendar month for 240 precipitation stations. Black line represents the median (used to estimate c in equation 2.1), the box the 1st and 3rd quartile and the whiskers 5th and 95th percentiles.

⁵For KNMI'06 P_{50} was used (Bakker and Bessembinder 2012).

3. Temperature

3.1 Changes in the KNMI'14 climate scenarios

The KNMI'14 scenarios project higher temperatures for all seasons. Especially the increases the first percentile of the daily minima in winter and the 99th percentile of daily maxima in summer are pronounced. Absolute changes in the 1st, 5th, 50th, 95th and 99th percentiles⁶ are provided for every calendar month for the daily mean Tg , minimum Tn and maximum Tx temperature [°C] separately for around 2050 (see figure 3.2 for country-average values) and 2085 compared to 1981-2010⁷. Next to the country-average percentile changes, also regional change factors are provided for five different regions (figure 3.1). These values may differ from the country-average values up to about 0.5°C (Van den Hurk et al. 2014, Figure 3.11).

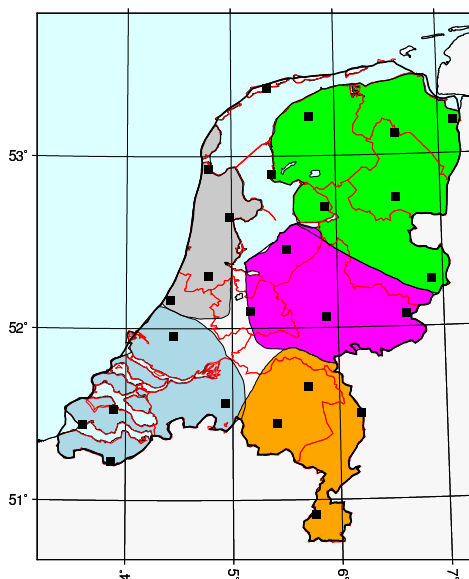


Figure 3.1: Regions for which different temperature percentile changes are provided (Noordoost Nederland (NON): green; Middenoost Nederland (MON): purple; Zuidoost Nederland (ZON): orange; Zuidwest Nederland (ZWN): light blue; Noordwest Nederland: grey).

⁶With respect to KNMI'06, a different set of percentiles is used (1st, 5th, 50th, 95th and 99th rather than the 10th, 50th and 90th) because this set is supposed to be more representative for the quantile changes.

⁷In the online version for the time horizons in between 1981-2010 and 2050 and in between 2050 and 2085 linear interpolation of the relative changes is used (not in the R-version). Also a scenario of 2030 is given in the KNMI'14 climate scenarios (KNMI 2014, p. 19).

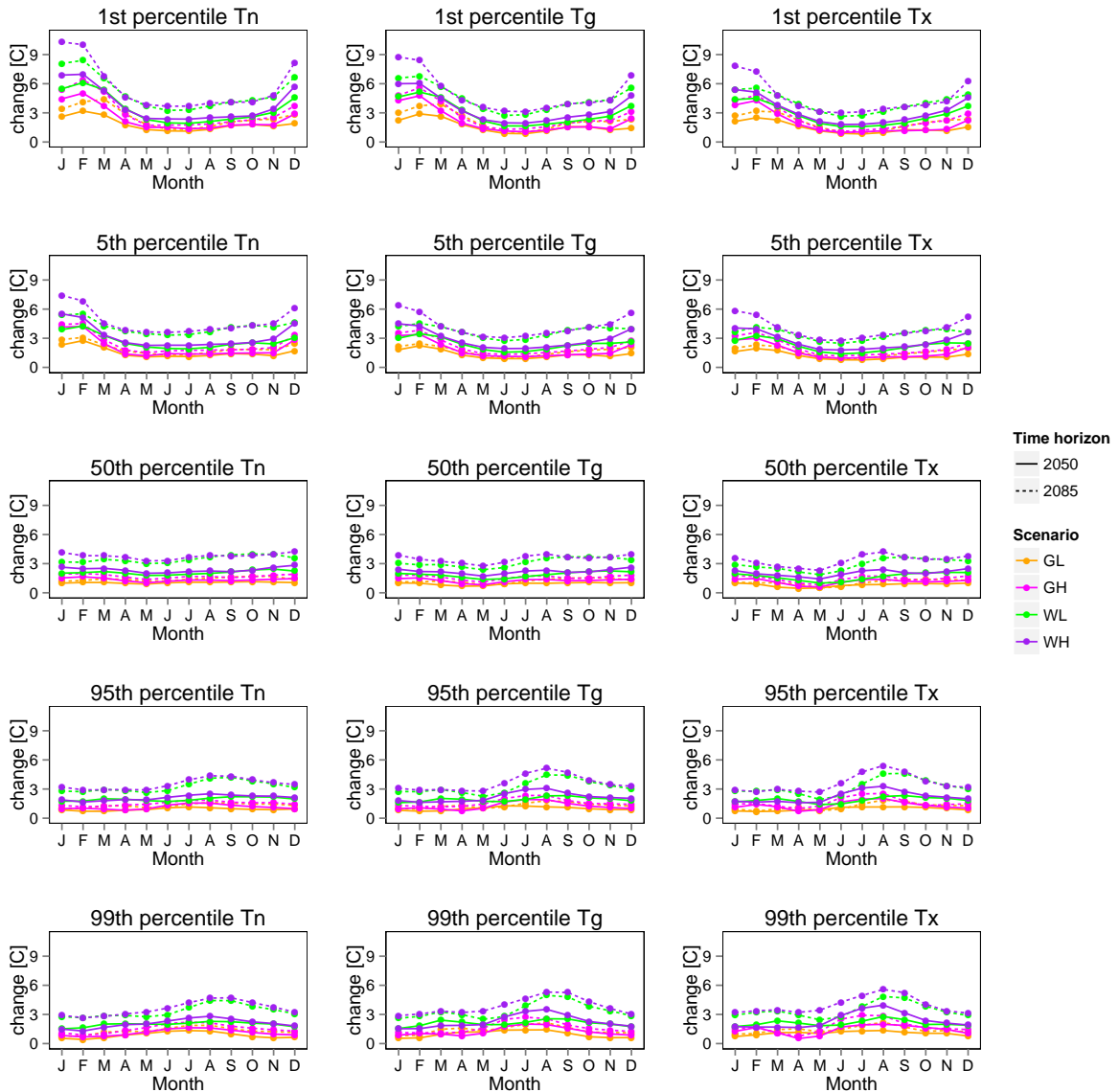


Figure 3.2: Projected relative changes per calendar month according to KNMI'14 (around 2050 (continuous lines) and 2185 (dashed lines) relative to 1981-2010) for 1st, 5th, 50th, 95th and 99th percentiles (from upper to lower panels) of the daily minimum T_n (left panels), daily mean T_g (central panels) and daily maximum T_x temperature (right panels).

3.2 Transformation procedure⁸

For T_g , T_n and T_x , a linear quantile scaling is applied to transform a reference time series T into a time series representative for the specific climate scenario T^f . This quantile scaling is based on the provided percentile changes.

⁸Apart from the different set of percentiles used and a slightly different notation, the method is basically the same as applied for KNMI'06.

$$\begin{aligned}
T_{01}^f &= T_{01} + \Delta T_{01} \\
T_{05}^f &= T_{05} + \Delta T_{05} \\
T_{50}^f &= T_{50} + \Delta T_{50} \\
T_{95}^f &= T_{95} + \Delta T_{95} \\
T_{99}^f &= T_{99} + \Delta T_{99}
\end{aligned} \tag{3.1}$$

The change for values in between these percentiles are estimated by linear interpolation and outside the 98%-range (for $T < T_{01}$ or $T > T_{99}$) by extrapolation

$$T^f = \begin{cases} a_1 T + b_1, & \text{if } T \leq T_{05} \\ a_2 T + b_2, & \text{if } T_{05} \leq T < T_{50} \\ a_3 T + b_3, & \text{if } T_{50} \leq T < T_{95} \\ a_4 T + b_4, & \text{if } T_{95} < T \end{cases} \tag{3.2}$$

where the a_1 - a_4 and b_1 - b_4 coefficients are estimated as follows

$$a_1 = \frac{T_{05}^f - T_{01}^f}{T_{05} - T_{01}}, \quad b_1 = T_{05}^f - a_1 T_{05} \tag{3.3}$$

$$a_2 = \frac{T_{50}^f - T_{05}^f}{T_{50} - T_{05}}, \quad b_2 = T_{50}^f - a_2 T_{50} \tag{3.4}$$

$$a_3 = \frac{T_{95}^f - T_{50}^f}{T_{95} - T_{50}}, \quad b_3 = T_{95}^f - a_3 T_{95} \tag{3.5}$$

$$a_4 = \frac{T_{99}^f - T_{95}^f}{T_{99} - T_{95}}, \quad b_4 = T_{99}^f - a_4 T_{99} \tag{3.6}$$

4. Global radiation

4.1 Changes in the KNMI'14 climate scenarios

The more frequent easterlies in summer the H-scenarios cause a small increase in global radiation Q . Figure 4.1 shows the monthly changes for the mean global radiation.

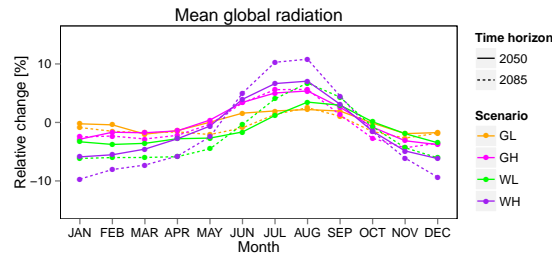


Figure 4.1: Projected changes in monthly mean solar radiation [%] according to KNMI'14 (around 2050 (continuous lines) and 2185 (dashed lines) relative to 1981-2010).

4.2 Transformation procedure⁹

Future time series of daily global radiation Q^f are obtained by a linear transformation, constrained by the fact that the global radiation should not exceed 75% of the incoming shortwave radiation at the Top of Atmosphere R_{TOA} ,

$$Q^f = \min(\alpha \cdot Q, 0.75R_{TOA}) \quad (4.1)$$

where R_{TOA} is a function of day-number and latitude (Allen et al. 1998, see also the function Angot in Appendix C of this report)^{10,11} and the coefficient α ($\geq 1 + \Delta Q_{av}$) is iteratively estimated such that the mean Q_{av} (per calendar month) of the original values Q is exactly perturbed by the relative change ΔQ_{av} ,

⁹ Note that the procedure is independent of applied unit, but that it is important to apply the unit consistently through the procedure. The transformation tool described in Appendix C derives R_{TOA} in [KJ/m²] from the provided latitude and day number. In accordance, the reference global radiation should be in KJ/m² too.

¹⁰So, the latitude should be specifically provided to the transformation tool (see Appendix C).

¹¹see footnote 9

$$Q_{av}^f = \sum_{i=1}^n \frac{Q(i)^f}{n} = \sum_{i=1}^n \frac{\min(\alpha \cdot Q(i), 0.75R_{TOA})}{n} = (1 + \Delta Q_{av})Q_{av} \quad (4.2)$$

where n is the number of observations in the particular calendar month.

5. Perspectives on other climate variables

5.1 Reference crop evapotranspiration

Future reference crop evapotranspiration (according to Makkink) EV_{mk}^f is **not** directly obtained by the transformation of a reference record EV_{mk} . Rather EV_{mk}^f is calculated from the transformed time series of daily mean temperature T_g^f and global radiation Q^f by the application of the Makkink equation (see Bakker 2015).

5.2 Relative humidity

Projected changes of relative humidity RH are small (Van den Hurk et al. 2014) and until now no transformation for the KNMI'14 scenarios is available. The projected changes per calendar month are available for around 2050 and 2085 compared to 1981-2010 and can be found in the "Toelichting transformatieprogramma" on http://www.klimaatscenarios.nl/toekomstig_weer/transformatie/index.html.

A similar approach as for global radiation can be applied, where RH cannot exceed 100%.

5.3 Wind

No transformation programme is available for wind. The changes in the KNMI'14 scenarios are also not significant. Principally, a proportional scaling can be applied to transform wind speed. However, changes per calendar month have not been determined until now.

Projected changes in wind direction are generally small and no transformation program is available. Until now, there is no satisfying method to obtain change factors of wind direction that provides mutually consistent direction frequencies for all directions.

References

- Allen, R. G., L. S. Pereira, D. Raes, and M. Smith (1998). *Crop Evapotranspiration: guidelines for computing crop water requirements*. FAO Irrigation and Drainage paper 56. FAO.
- Bakker, A. and J. Bessembinder (2012). *Time series transformation tool: description of the program to generate time series consistent with the KNMI'06 climate scenarios*. technical report TR-326. De Bilt, the Netherlands: KNMI. URL: <http://www.knmi.nl/bibliotheek/knmipubTR/TR326.pdf>.
- Bakker, A. (2015). *Climatological time series for the KfC project High-Quality Climate Projections (Theme 6 WP3)*. technical report TR-348. De Bilt, the Netherlands: KNMI. URL: <http://www.knmi.nl/bibliotheek/knmipubTR/TR348.pdf>.
- IPCC (2013). *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Ed. by T. Stocker, D. Qin, G.-K. Plattner, M. Tignor, S. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex, and P. Midgley. Cambridge, United Kingdom and New York, NY, USA: Cambridge University Press, p. 1535.
- KNMI (2014). *KNMI'14 climate scenarios for the Netherlands; A guide for professionals in climate adaptation*. Ed. by A. Klein Tank, J. Beersma, J. Bessembinder, B. van den Hurk, and G. Lenderink. De Bilt, the Netherlands: KNMI, p. 34. URL: http://www.climatescenarios.nl/images/Brochure_KNMI14_EN.pdf.
- Sluijter, R., C. Reijmerink, B. Overbeek, J. Bessembinder, R. van Dorland, A. Kattenberg, H. Geurts, P. Siegmund, A. Feijt, and J. Buisman (2011). *Bosatlas van het klimaat*. Groningen, the Netherlands: Noordhoff, p. 112. URL: www.klimaatatlas.nl.
- Van den Hurk, B., P. Siegmund, A. Klein Tank (Eds.), J. Attema, A. Bakker, J. Beersma, J. Bessembinder, R. Boers, T. Brandsma, H. van den Brink, S. Drijfhout, H. Eskes, R. Haarsma, W. Hazeleger, R. Jilderda, C. Katsman, G. Lenderink, J. Loriaux, E. van Meijgaard, T. van Noije, G. J. van Oldenborgh, F. Selten, P. Siebesma, A. Sterl, H. de Vries, M. van Weele, R. de Winter, and G.-J. Zadelhoff (2014). *KNMI'14: Climate Change scenarios for the 21st Century - A Netherlands perspective*. WR2014-01. De Bilt, the Netherlands: KNMI. URL: <http://www.knmi.nl/bibliotheek/knmipubWR/WR2014-01.pdf>.

Appendices

A. R-scripts for transformation program Precipitation¹²

```
#####  
#  
# rr_trans_3.1_KNMI14.R    March 10, 2015  
#  
5 # author: Alexander Bakker (KNMI)  
#  
# Function 'transforms' a specific reference-dataset with daily precipitation  
# sums [mm] to a dataset representative for a future climate scenario.  
#  
10 # Transformation developed for KNMI'14 climate change scenarios for the  
# Netherlands:  
#  
# Bakker, A. (2015), Time series transformation tool version 3.1: description of  
# the program to generate time series consistent with the KNMI'14 climate  
15 # scenarios, Technical Report TR-349, De Bilt, the Netherlands  
#  
# Version History:  
#   version 3.1 / version 3.1.1 - March 10, 2015  
#  
20 # THIS CODE IS PROVIDED AS-IS WITH NO WARRANTY (NEITHER EXPLICIT  
# NOT IMPLICIT). KNMI SHARES THIS CODE IN HOPES THAT IT IS USEFUL,  
# BUT KNMI IS NOT LIABLE FOR THE BEHAVIOR OF THIS CODE IN YOUR OWN  
# APPLICATION. YOU ARE FREE TO SHARE THIS CODE SO LONG AS THE  
# AUTHOR(S) AND VERSION HISTORY REMAIN INTACT.  
25 #  
#  
# arguments:  
#  
# obs          data.frame or matrix:  
30 #             first column provides datestring YYYYMMDD  
#             other columns provide precipitation [mm] time series (each  
#             column represents specific station)  
#  
# deltas      data.frame or matrix that contains deltas (=change factors for  
35 #             the transformation)  
#             should contain following columns indicated by following headers  
#             HEADER  
#             "wdf"          relative change [%] in wet-day frequency  
#                             (wet day is defined as day with 0.1 mm or more  
#                             precipitation)  
40 #             "ave"        relative change [%] in mean precipitation  
#             "p99"        relative change [%] in the 99th percentile of  
#                             wet-day amounts  
#
```

¹²Note that the online tool is based on Fortran and may provide slightly different time series.

```

#
45 # version      "v3.1" [default] official version belonging to KNMI'14
#             "v3.1.1" alternative way to dry wet days
#
#####
50 rr_trans_KNMI14 <- function(obs,
                             deltas,
                             version="v3.1") {

# DEFINE CONSTANTS
55 th      <- 0.1    # wet-day threshold
qq1     <- 0.99    # quantile of wet-day amounts that is used to estimate
# transformation coefficients
qq2     <- 0.90    # quantile of wet-day amounts that is used to estimate qq1
# (robustly)
60 ratio  <- c(2.22,
              2.271,
              2.366,
              2.147,
              2.346,
65              2.166,
              2.276,
              2.404,
              2.476,
              2.087,
70              2.336,
              2.18) # national median of monthly ratios between qq1 and qq2 for
# 240 precipitation stations

# PREPARE DATA
75 # explore observations
ns      <- ncol(obs) - 1      # number of stations
# (= number of columns minus 1)
mm      <- (obs[,1]%/%100)%%100 # the month that a day belongs to
# (1, 2, ..., 12)
80 nr      <- length(mm)      # total number of days (in reference file)

# determine observed wet-day frequency (wdf.obs), mean (mean.obs), wet-day
# mean (mwet.obs), wet-day 99th percentile
85 wdf.obs  <- as.matrix(aggregate(obs[,-1],by=list(mm),
                                function(x) mean(x[x]>=th) )),[-1]
mean.obs  <- as.matrix(aggregate(obs[,-1],by=list(mm),
                                function(x) mean(x) )),[-1]
mwet.obs  <- as.matrix(aggregate(obs[,-1],by=list(mm),
                                function(x) mean(x[x]>=th) )),[-1]
90 q2.obs  <- as.matrix(aggregate(obs[,-1],by=list(mm),
                                function(x) quantile(x[x]>=th],0.90))),[-1]
q1.obs    <- q2.obs*ratio

# apply deltas to observed climatology to obtain future climatology
95 fut     <- obs; fut[,-1] = NA      # future values (filled with NA)
wdf.fut   <- wdf.obs * (1 + deltas$wdf/100) # future climatologies
mean.fut  <- mean.obs * (1 + deltas$ave/100)
mwet.fut  <- mean.fut / wdf.fut
100 q1.fut  <- q1.obs * (1 + deltas$P99/100)

# TRANSFORMATION
# apply transformation per station / time series

```

```

105 for(is in 1:ns) {
    print(is)
    Y <- obs[,is+1] # original time series of station 'is'
                    # that will be adjusted

# DRYING WET DAYS #####
110 if(sum(deltas$wdf<0) > 0) { # check if reduction in wet days is needed

    # VERSION V3.1 is official KNMI14 "drying procedure" (see TR-349)
    if(version=="v3.1") {

115         # select target values
        target.values <- vector() # 'target precipitation amounts' to dry
        target.months <- vector() # specific month to which this target
                                   # values belong

120         # add very small number (based on datestring) to ensure that all numbers
        # in time series are unique. This is necessary for the selection of
        # 'days to dry'
        X <- ifelse(Y<th,Y,Y + 0.01*obs[,1]/100000000)

125         # loop all months for which a reduction of the wet day is projected
        for(im in which(deltas$wdf<0)) {

            # sorted vector of all wet day amounts that fall in month <im>
            Xw <- sort(X[which(X>=th & mm==im)])

130

            # number of days 'to dry'
            ndry <- round((-1*deltas$wdf[im]/100)*length(Xw))
            if(ndry > 0) {
                step <- length(Xw)/ndry # step size to step through wet day
                                         # amount vector <Xw> (NOT AN INTEGER)

135

                # determine target values for month <im> (homogeneously selected
                # from subset <Xw>) and remember specific month <im> that belongs
                # to target.values
140                target.values <- c(target.values,Xw[round(((1:ndry)-0.5)*step)])
                target.months <- c(target.months,rep(im,ndry))
            }
        }

145         # assign rank order to all target.values (and belonging month-id)
        target.months <- target.months[order(target.values)]
        target.values <- target.values[order(target.values)]

        # selection of days to dry
        droogmaken <- vector() # 'days to dry'
150         for(idry in 1:length(target.values)) { # step through all target
                                                    # values from small to large

            # select all days that are currently available for drying (during
            # the drying procedure new wet days may become available for drying)
155             available <- which(mm == target.months[idry] &
                                   # all days within same
                                   # month as target
                                   # value
                                   X >= th &
                                   # all wet days
                                   (c(0, X[-nr]) <th |
                                   # all days preceded
                                   # and/or succeeded by
                                   # dry day
                                   c( X[-1] ,0) <th) )

```

```

165     # determine which of all available days is closest to the target.value
# and put day_id in vector containing days to dry
droogmaken <- c(droogmaken,available[
    which(abs(X[available]-target.values[idry]) ==
        min(abs(X[available]-target.values[idry])))]))

170     # dry specific day in vector of adjusted values
X[droogmaken[idry]] <- 0
}
# actual drying of original time series
Y[droogmaken] <- 0

175     # END VERSION V3.1 #

} else {

180     # VERSION V3.1.1 has alternative procedure (not documented in TR349)

# loop all months for which a reduction wdf is projected
for(im in which(deltas$wdf<0)) {
185     rows <- which(mm==im & Y>=th) # identify all wet days in month
# <im>

Xw <- sort(Y[rows]) # sort wet days amounts
ndry <- round((-1*deltas$wdf[im]/ # number of wet days to dry
190     100)*length(Xw))

if(ndry > 1) {
    c <- Xw[ndry] # c = constant to subtract of
# daily values

195     # check if it is better to lower c with respect to tied data
# (i.e. days with same value) and apply if necessary.
# This is important in case of small <c> and small <ndry>
if(abs(length(which(Xw<=c)) - ndry) >
    abs(length(which(Xw< c)) - ndry)) {
200     c <- ifelse(Xw[1]==c,0,max(Xw[which(Xw<c)]))
}

# actual drying
Y[rows] <- ifelse(Y[rows]<=c,0,Y[rows]-c)

205     # adjust empirical PDF of 'dried' data to match original empirical
# PDF
n.wd <- which(mm==im & Y>=th) # wet days after drying
# (in contrast to rows)
PP.Yw <- rank(Y[n.wd])/length(n.wd) # wet days empirical
# frequency of non-exceedance
210     Y[n.wd] <- quantile(Xw,PP.Yw) # apply distribution of Xw to
# non-exceedance frequencies
}
}
} # END VERSION V3.1.1 #
} # END DRYING WET DAYS

# WETTING DRY DAYS #####
220 X <- Y # time series after drying
X1 <- c(1,X[-nr]) # precipitation of preceding day
# (preceding day of first day is
# assigned 1 mm)

```



```

# loop through 12 calendar months
225 for(im in 1:12) {
    if(deltas$wdf[im]>0) {
        # in case an increase of wdf is
        # projected

        rows <- which(mm==im) # identify all days in month <im>
        Xm <- X[rows] # subset all days in month <im>
230 X1m <- X1[rows] # and all preceding days
        Xw <- sort(Xm[which(Xm>=th)]) # sort all wet day values
        dwet <- round((deltas$wdf[im]/ # number of 'dry days to wet'
            100)*length(Xw))

        # in case days to wet is larger than zero
        if(dwet > 0) {
            # select target values
            step <- length(Xw)/dwet # step size to step
            # through sorted step

240 # determine target.values for month <im> (homogeneously selected from
            # sorted subset)
            target.values <- Xw[round(((1:dwet)-0.5)*step)]

245 # select days to wet
            preceding.wet <- cumsum(Xm>=th) + step/2 # cumulative number of
            # preceding wet days in
            # month <im> + step/2

            add <- vector() # vector with days that
            # should be wetted

250 for(id in 1:dwet) {
                add <- c(add,
                    which(Xm < th & # select 'first' 'dry' day that
                    X1m >= th & # succeeds a wet' day,
                    preceding.wet >= step)[1]) # for which <preceding.wet>
            # exceeds the <step> size
            # and add this day(id) to
            # vector <add>

255 # if possible
            if(is.na(add[id])) {
                add <- add[-id]
            } else {
                preceding.wet <- preceding.wet - step # and decrease vector
                # <preceding.wet> with
                # <step>

260 preceding.wet[1:add[id]] <- 0
            }
        }

270 # Finally, target.values are assigned to selected days on the basis of
        # the rank order of the precipitation amount of the preceding wet day
        Y[rows[add]] <- target.values[rank(X1m[add],ties.method="first")]

275 } # dfwet > 0
    } # days need to be added
} # calendar month

280 # TRANSFORM WET-DAY AMOUNTS #####
#determine coefficients

# NOTE that from here Y represents wet-day frequency adjusted time series

```

```

285 for(im in 1:12) {
      wet.im <- which(im==mm & Y>=th)           # identify all wet days within
                                              # calendar month <im>
      Xm      <- Y[wet.im]                     # select all wet day amounts
290      mobs   <- as.numeric(mwet.obs[im,is])   # get climatologies for reference
                                              # and future period
      qobs    <- as.numeric(q1.obs[im,is])
      mfut    <- as.numeric(mwet.fut[im,is])
      qfut    <- as.numeric(q1.fut[im,is])

295      # determine exponent 'b' of power-law correction function

      # function to minimise to find optimal value for coefficient 'b'
      f <- function(b) {
300          qfut/mfut - (qobs^b)/mean(iffelse(Xm<qobs,Xm^b,Xm*(qobs^b)/qobs))
      }

      # root finding algorithm requires that both sides of search space are of
      # opposite sign
      if(f(0.1)*f(3)<0) {
305          rc <- uniroot(f,lower=0.1,upper=3,tol=0.001) # root finding
          b <- rc$root
      } else {
                                              # if root is non-existent,
                                              # alternative estimation
                                              # value closest to zero is
310          # searched for
          bs <- (1:300)/100                    # determine search space
                                              # for 'b'
          fs <- bs                             # fs = f(bs)
          for(ifs in 1:length(fs)) {
315              fs[ifs] <- f(bs[ifs])
          }
          b <- bs[which(abs(fs)==min(abs(fs)))] # b for which f(b) is
                                              # smallest is chosen
      }

320      # straightforward estimation of coefficients a and c
      a <- qfut/(qobs^b)
      c <- a*(qobs^b)/qobs # multiplication factor for values larger than q99

325      # actual transformation of wet-day amounts (application of transformation
      # function)
      Y[wet.im] <- iffelse(Xm<qobs,a*Xm^b,c*Xm)

      Y[wet.im][which(Y[wet.im]<th)] <- th # prevent days being dried by the
330          # wet-day transformation
    }
    # END TRANSFORMATION WET-DAY AMOUNTS

    fut[,is+1] <- round(Y,1) # write transformed series to table
335
  } # END TRANSFORMATION LOOP

  return(fut)
340 } # end rr_trans_KNMI14

```

Listing A.1: Precipitation Transformation version 3.1

```

#####
#
# neerslag_transformatie_KNMI14.R      March 10, 2015
#
5 # author: Alexander Bakker (KNMI)
#
# Function reads 'reference data' with daily precipitation sums [mm] and
# 'change factors' from input files
# and applies them to function 'rr_trans_KNMI14' to obtain
10 # 'future time series' that match a certain climate
#
# Transformation developed for KNMI'14 climate change scenarios for the
# Netherlands:
#
15 # Bakker, A. (2015), Time series transformation tool version 3.1: description of
# the program to generate time series consistent with the KNMI'14 climate
# scenarios, Technical Report TR-349, De Bilt, the Netherlands
#
#
20 # THIS CODE IS PROVIDED AS-IS WITH NO WARRANTY (NEITHER EXPLICIT
# NOT IMPLICIT). KNMI SHARES THIS CODE IN HOPES THAT IT IS USEFUL,
# BUT KNMI IS NOT LIABLE FOR THE BEHAVIOR OF THIS CODE IN YOUR OWN
# APPLICATION. YOU ARE FREE TO SHARE THIS CODE SO LONG AS THE
# AUTHOR(S) AND VERSION HISTORY REMAIN INTACT.
25 #
#
# arguments:
#
# ifile          Name of the input file (ASCII) that contains reference data
30 #              (all numerics) in which the columns provide time series for
#              specific stations. The first column should provide either
#              00000000 or a datestring YYYYMMDD:
#              Rows starting with 00000000 are considered station info
#              (station number, lat, lon etc.) and are ignored. Rows starting
35 #              with a datestring refer to a specific day in the time series.
#              Rows starting with "#" are completely ignored and returned
#              unchanged.
#
# ofile          (DEFAULT="uitvoer.txt") Name of the output file to write the
40 #              transformed data to. Format is similar to ifile
#
# delta.file     [optional] Name of file that contains deltas (changes factors
#              for the transformation). File should contain following
#              compulsory columns identified with compulsory headers
45 #              "wdf"      relative change in wet-day frequency
#                          (wet days are defined as days with 0.1 mm or more
#                          precipitation)
#              "ave"      relative change in mean precipitation
#              "p99.lower" lower estimate of the relative change in the 99th
50 #              percentile of wet-day amounts
#              "p99.cent" central estimate of the relative change in the 99th
#              percentile of wet-day amounts
#              "p99.upper" upper estimate of the relative change in the 99th
#                          percentile of wet-day amounts
#
55 #              (If delta.file is not provided, predefined deltas are derived
#              depending on <sc>, <p> and <scaling>)
#
# sc            scenario          ["GL", "GH", "WL", "WH"]
# p            time horizon      [2030 (=DEFAULT), 2050, 2085]

```

```

60 # scaling      scaling extreme precipitation ["lower", "centr" (=DEFAULT),
#                                                    "upper"]
#
# version      "v3.1" [DEFAULT] official version that belongs to KNMI'14
#              "v3.1.1" alternative procedure to dry wet days
65 #
#####
neerslag_transformatie_KNMI14 <- function(ifile,
70                                     ofile="uitvoer.txt",
                                     delta.file=NA,
                                     sc,
                                     p=2030,
                                     scaling="centr",
                                     version="v3.1") {
75
# READ REFERENCE DATA FROM ifile
# select lines with "#" from reference file and ignore them
H.comments <- scan(ifile, character(0), sep = "\n")
H.comments <- H.comments[grepl("#",H.comments)] # (only necessary for
80 # output file)
obs      <- read.table(ifile,header=F)         # read reference data
header   <- obs[which(obs[,1]==0),]          # header containing
# stations meta-data etc.
header[,1] <- "00000000"
85 names(obs) <- c("date",round(obs[1,-1],0)) # station names are read
# from first line
obs      <- obs[which(obs[,1]!=0),]          # actual data

# READ CHANGE FACTORS (DELTAS)
90 if(!is.na(delta.file)) {
  deltas <- read.table(delta.file,header=T)   # deltas are provided in
# file <delta.file>
} else {
  if(p=="2030") {
95     # 2030 decadal prediction if p=2030
    deltas <- read.table("deltas-KNMI14_rr_____2030.txt",header=T)
  } else {
    # str.ext is a function used to construct <file.names> if <delta.file> is
    # not explicitly provided
100    str.ext <- function(var,ch,n) {
      paste(var,substr(paste(rep(ch,n),collapse=""),1,n-nchar(var)),sep="") }

    deltas <- read.table(paste("deltas-KNMI14_rr___",
105                          str.ext(sc,"_",3),"_",
                          str.ext(p,"_",4),".txt",
                          sep=""),header=T)
  }
}
# choose scaling ("lower", "centr" or "upper")
110 deltas$P99 <- deltas[,paste("p99",scaling,sep=".")]

# TRANSFORMATION
source("rr_trans_3.1_KNMI14.R")
fut <- rr_trans_KNMI14(obs=obs, deltas=deltas, version="v3.1")
115
# OUTPUT
sink(ofile)

# comments

```

```

120 writeLines("# Transformed daily precipitation sums [mm] according to KNMI'14
      transformation tool,")
writeLines(paste("# version ",version,sep=""))
if(is.na(p) | is.na(scaling)) {
  writeLines(paste("# Deltas are derived from ",sc,sep=""))
} else {
125   if(p=="2030") {
      writeLines("# Deltas are derived from the 2030 decadal prediction")
    } else {
      writeLines(paste("# Deltas are derived from ",sc," scenario",sep=""))
      writeLines(paste("# around the time horizon ",p,sep=""))
130   }
}
writeLines(paste("# wet day scaling:",scaling))
writeLines("#")
writeLines("# Bakker A. (2015), Time series transformation tool: description of
      the program to")
135 writeLines("# generate time series consistent with the KNMI'14 climate
      scenarios, TR-349")
writeLines("#")
for(i in 1:length(H.comments)) writeLines(H.comments[i])

# header
140 write.table(format(header[1,],
                      width=10,
                      justify="right"),
              row.names=F,
              col.names=F,
              quote=F)
145 write.table(format(header[-1,],
                      width=10,
                      justify="right"),
              row.names=F,
              col.names=F,
              quote=F)
150 # transformed data
write.table(format(fut,
                  width=10,
155                  digits=2,
                  justify="right"),
            row.names=F,
            col.names=F,
            quote=F)

160 sink()
}

```

Listing A.2: R-function to apply tool to specific dataset

```

# EXAMPLE FILE TRANSFORMATION TOOL KNMI'14

# clean environment and set directory
rm(list = ls())
5 graphics.off()

# PRECIPITATION TRANSFORMATION
source("neerslag_transformatie_KNMI14.R")

10 ifile="neerslag_ref" # input file
   ofile="uitvoer.txt" # output file (DEFAULT="uitvoer.txt")

```

```

15  delta.file=NA           # file containing deltas
                                # if delta.file is not provided (DEFAULT)
                                # KNMI'14 deltas are used

# deltas are derived from KNMI'14 deltas if delta.file is not specified
# following arguments are used
sc="WH"                       # scenario ("GL", "GH", "WL", "WH")
p=2085                         # time horizon (2030, 2050, 2085)
20  scaling="centr"         # scaling subscenario ("lower", "centr" [DEFAULT],
                                # "upper")
version="v3.1."               # version of transformation ("v3.1" [DEFAULT], "v3.1.1")

25  neerslag_transformatie_KNMI14(ifile=ifile,
                                ofile=ofile,
                                delta.file=delta.file,
                                sc=sc,
                                p=p,
                                scaling=scaling,
30  version=version)

```

Listing A.3: Example how to call R-function

B. R-scripts for transformation program Temperature¹³

```
#####  
#  
# tm_trans_3.1_KNMI14.R    March 11, 2015  
#  
5 # author: Alexander Bakker (KNMI)  
#  
# Function 'transforms' a specific reference-dataset with time series of daily  
# of mean, minimum or maximum temperature [degrees Celsius] to a dataset  
# representative for a future climate scenario.  
10 #  
# Transformation developed for KNMI'14 climate change scenarios for the  
# Netherlands:  
#  
# Bakker, A. (2015), Time series transformation tool version 3.1: Description of  
# the program to generate time series consistent with the KNMI'14 climate  
15 # scenarios, Technical Report TR-349, De Bilt, the Netherlands  
#  
# Version History:  
# v3.1 - March 11, 2015 (initial version)  
#  
20 # THIS CODE IS PROVIDED AS-IS WITH NO WARRANTY (NEITHER EXPLICIT  
# NOT IMPLICIT). KNMI SHARES THIS CODE IN HOPES THAT IT IS USEFUL,  
# BUT KNMI IS NOT LIABLE FOR THE BEHAVIOR OF THIS CODE IN YOUR OWN  
# APPLICATION. YOU ARE FREE TO SHARE THIS CODE SO LONG AS THE  
# AUTHOR(S) AND VERSION HISTORY REMAIN INTACT.  
25 #  
#  
# arguments:  
#  
# obs          data.frame or matrix:  
30 #             first column provides datestring YYYYMMDD  
#             other columns provide precipitation [mm] time series (each  
#             column represents specific station)  
#  
# deltas       data.frame or matrix that contains deltas (=change factors for  
35 #             the transformation) should contain following columns indicated  
#             by following headers  
#             HEADER  
#             "maand"    month for which deltas are valid (1,2,...,12)  
#             "P01"     1st percentile daily temperature  
40 #             "P05"     5th percentile daily temperature  
#             "P50"     50th percentile daily temperature  
#             "P95"     95th percentile daily temperature
```

¹³Note that the online tool is based on Fortran and may provide slightly different time series.

```

#           "P99"           99th percentile daily temperature
#
45 #           following column is optional in case deltas vary with region
#           (is needed in case <regio.tabel> is provided)
#           "regio"        region for which deltas are valid
#                           KNMI14 distinguishes ("NWN", "ZWN", "NON", "MON",
#                           "ZON", "NLD")
50 #
# regio.tabel  this (optional) argument provides a vector that relates the
#               stations to a particular region. Vector provides references to
#               region. Location (rownumber [i] in vector refers to column in
#               obs [i+1]
55 #
#               KNMI14 distinguishes following regions:
#               <NLD> Nederland           [DEFAULT]
#               <NWN> Noordwest Nederland
#               <ZWN> Zuidwest Nederland
60 #               <NON> Noordoost Nederland
#               <MON> Middenoost Nederland
#               <ZON> Zuidoost Nederland
#
#####
65 tm_trans_KNMI14 <- function(obs,
                             deltas,
                             regio.tabel=NA) {

70   version="v3.1"

# PREPARE DATA
# explore observations
ns      <- ncol(obs) - 1      # number of stations
75      # (= number of columns minus 1)
mm      <- (obs[,1]%/%100)%%100 # the month that a day belongs to
      # (1, 2, ..., 12)
nr      <- length(mm)        # total number of days (in reference file)
80      fut      <- obs; fut[,-1] = NA # future values (filled with NA)

# region information
# add column <regio> to deltas if not provided
if(!("regio" %in% colnames(deltas))) deltas$regio <- "NLD"
# arrange deltas
85 deltas <- deltas[c("regio", "maand", "P01", "P05", "P50", "P95", "P99")]

# TRANSFORMATION
# apply transformation per station <is> / time series
for(is in 1:ns) {
90   regio <- ifelse(is.na(regio.tabel), "NLD", regio.tabel[is]) # determine region

   for(im in 1:12) {
     # get change factors for specific station and month
     percentile.changes <- as.numeric(
95       deltas[which(deltas[,1]==regio & deltas[,2]==im),-1:-2] )

     # all days within in calendar month <im>
     days.im      <- which(mm == im)
     # select all obs in month <im> of station <is>
100    X          <- obs[days.im, is+1]
     Y          <- rep(NA, length(X))
     # observed percentiles

```



```

X.percentiles <- as.numeric(quantile(X,c(1,5,50,95,99)/100,na.rm=T))
# estimation of future percentiles
105 Y.percentiles <- X.percentiles + percentile.changes

# linear transformation: for intervals X<qq5, qq5<X<qq50, qq50<X<qq95,
# qq95<X
# ip = percentile id
110

# X < X.percentile[2] (5th percentile)
ip=2
x.ids      <- which(X < X.percentiles[ip]) # id's for all values smaller
# than second smallest

115 # estimate linear function for percentile range
a          <- (Y.percentiles[ip] - Y.percentiles[ip-1]) /
            (X.percentiles[ip] - X.percentiles[ip-1])
b          <- Y.percentiles[ip] - a * X.percentiles[ip]
# apply function to all temperatures below 5th percentile
120 Y[x.ids]  <- a * X[x.ids] + b

for(ip in 3:(length(X.percentiles)-1)) {
  # id's all values within analysed percentile range
  x.ids    <- which(X >= X.percentiles[ip-1] & X < X.percentiles[ip])
125 a      <- (Y.percentiles[ip] - Y.percentiles[ip-1]) /
            (X.percentiles[ip] - X.percentiles[ip-1])
  b      <- Y.percentiles[ip] - a * X.percentiles[ip]
  Y[x.ids] <- a * X[x.ids] + b
}

130 ip=length(X.percentiles)
# id's for all values larger than second largest
x.ids    <- which(X >= X.percentiles[ip-1])
135 a      <- (Y.percentiles[ip] - Y.percentiles[ip-1]) /
            (X.percentiles[ip] - X.percentiles[ip-1])
b      <- Y.percentiles[ip] - a * X.percentiles[ip]
Y[x.ids] <- a * X[x.ids] + b

140 fut[days.im, is+1] <- round(Y,1)      # round results and write to fut
} # END MONTHLY LOOP

} # END OF TRANSFORMATION LOOP

145 return(fut)
} # end tm_trans_KNMI14

```

Listing B.1: Temperature Transformation version 3.1

```

#####
# temperatuur_transformatie_KNMI14.R      March 11, 2015
#
# author: Alexander Bakker (KNMI)
5 #
# Function reads 'reference data' with ime series of daily of mean, minimum or
# maximum temperature [degrees Celsius] and 'change factors' from input files
# and applies them to function 'tm_trans_KNMI14' to obtain 'future time series'
# that match a certain climate
10 #
# Transformation developed for KNMI'14 climate change scenarios for the
# Netherlands:
#

```

```

15 # Bakker, A. (2015), Time series transformation tool version 3.1: Description of
# the program to generate time series consistent with the KNMI'14 climate
# scenarios, Technical Report TR-349, De Bilt, the Netherlands
#
#
# THIS CODE IS PROVIDED AS-IS WITH NO WARRANTY (NEITHER EXPLICIT
20 # NOT IMPLICIT). KNMI SHARES THIS CODE IN HOPES THAT IT IS USEFUL,
# BUT KNMI IS NOT LIABLE FOR THE BEHAVIOR OF THIS CODE IN YOUR OWN
# APPLICATION. YOU ARE FREE TO SHARE THIS CODE SO LONG AS THE
# AUTHOR(S) AND VERSION HISTORY REMAIN INTACT.
#
25 #
# arguments:
#
# ifile          Name of the input file (ASCII) that contains reference data
#                (all numerics) in which the columns provide time series for
30 #                specific stations. The first column should provide either
#                00000000 or a datestring YYYYMMDD: Rows starting with 00000000
#                are considered station info (station number, lat, lon etc.) and
#                are ignored. Rows starting with a datestring refer to a
#                specific day in the time series. Rows starting with "#" are
35 #                completely ignored and returned unchanged
#
# ofile          (DEFAULT="uitvoer.txt") Name of the output file to write the
#                transformed data to. Format is similar to ifile
#
40 # delta.file    [optional] Name of file that contains deltas (changes factors
#                for the transformation)
#                File should contain following compulsory columns identified
#                with compulsory headers
#                - HEADER -
45 #                "maand"      month for which deltas are valid (1,2,...,12)
#                "P01"        1st percentile daily temperature
#                "P05"        5th percentile daily temperature
#                "P50"        50th percentile daily temperature
#                "P95"        95th percentile daily temperature
50 #                "P99"        99th percentile daily temperature
#
#                following column is optional in case deltas vary with region
#                (is needed in case <regio.tabel> is provided)
#                "regio"      region for which deltas are valid
55 #                KNMI14 distinguishes ("NWN", "ZWN", "NON", "MON",
#                "ZON", "NLD")
#
#                (If delta.file is not provided, predefined deltas are derived
#                depending on <sc>, <p> and daily temperature characteristic
60 #                of interest [mean, min or max])
#
# sc            scenario          ["GL", "GH", "WL", "WH"]
# p            time horizon       [2030 (=DEFAULT), 2050, 2085]
# var         kind of daily temperature variable ["tg" = daily mean,
65 #                "tn" = daily minimum,
#                "tx" = daily maximum]
#
# regio.file    this (optional) argument provides the name of an ASCII file
#                that relates the stations to a particular region. First column
70 #                is station id and second column region.
#                KNMI14 distinguishes following regions:
#                <NLD> Nederland          [DEFAULT]
#                <NWN> Noordwest Nederland

```

```

75 #           <ZWN> Zuidwest Nederland
#           <NON> Noordoost Nederland
#           <MON> Middenoost Nederland
#           <ZON> Zuidoost Nederland
#
#####
80
temperatuur_transformatie_KNMI14 <- function(ifile,
                                           ofile="uitvoer.txt",
                                           delta.file=NA,
85                                           sc,
                                           p=2030,
                                           var,
                                           regio.file=NA) {

# CONSTANTS AND FUNCTIONS #####
90 version="v3.1"

# READ REFERENCE DATA FROM ifile
# select lines with "#" from reference file and ignore them
# (only necessary for output file)
95 H.comments <- scan(ifile, character(0), sep = "\n")
H.comments <- H.comments[grep("#",H.comments)]

obs      <- read.table(ifile,header=F)           # read reference data
header   <- obs[which(obs[,1]==0),]             # header met stations
                                                # meta-data etc.
100 header[,1] <- "00000000"
names(obs) <- c("date",round(obs[1,-1],0))     # station names are read
                                                # from first line
obs      <- obs[which(obs[,1]!=0),]             # actual data
105
# READ CHANGE FACTORS (DELTAS)
if(!is.na(delta.file)) {
  deltas <- read.table(delta.file,header=T)     # deltas are provided in
                                                # file <delta.file>
110 } else {
  if(p=="2030") {
    deltas <- read.table(paste("deltas-KNMI14__",var,"-----2030.txt",
    sep=""),header=T) # 2030 decadal prediction if p=2030
  } else {

# str.ext is a function used to construct file.names if delta.file is not
# explicitly provided
str.ext <- function(var,ch,n) {
  paste(var,substr(paste(rep(ch,n),collapse=""),1,n-nchar(var)),sep="") }
115
  deltas <- read.table(paste("deltas-KNMI14__",var,"___",
                             str.ext( sc,"_",3),"_",
                             str.ext( p,"_",4),".txt",sep=""),header=T)
120
  }
}

125 # LINK STATIONS TO REGIONS
if(is.na(regio.file)) {
  regio.tabel <- NA
} else {
130 stationstabel <- read.table("stationstabel")
regio.tabel <- as.vector(stationstabel[match(names(obs)[-1],
                                             stationstabel[,1]),2])

```

```

}
135 # TRANSFORMATION
source("tm_trans_3.1_KNMI14.R")
fut <- tm_trans_KNMI14(obs=obs, deltas=deltas, regio.tabel=regio.tabel)

# OUTPUT #####
140 sink(ofile)

# comments
stat <-c("mean","min","max")[match(var,c("tg","tn","tx"))]
writeLines(paste("# Transformed daily",stat,"temperature according to KNMI'14
  transformation tool,"))
145 writeLines(paste("# version ",version,sep=""))
if(is.na(p)) {
  writeLines(paste("# Deltas are derived from ",sc,sep=""))
} else {
  if(p=="2030") {
150   writeLines("# Deltas are derived from the 2030 decadal prediction")
  } else {
    writeLines(paste("# Deltas are derived from ",sc," scenario",sep=""))
    writeLines(paste("# around the time horizon ",p,sep=""))
  }
155 }
writeLines("#")
writeLines("# Bakker A., 2014. Time series transformation tool: description of
  the program to")
writeLines("# generate time series consistent with the KNMI'14 climate
  scenarios, TR-349")
writeLines("#")
160 for(i in 1:length(H.comments)) writeLines(H.comments[i])
writeLines("#")

# header
write.table(format(header[1,],
165   width=10,
   justify="right"),
  row.names=F,
  col.names=F,
  quote=F)
write.table(format(header[-1,],
170   width=10,
   justify="right"),
  row.names=F,
  col.names=F,
  quote=F)
175 # transformed data
write.table(format(fut,
  width=10,
  digits=2,
180   justify="right"),
  row.names=F,
  col.names=F,
  quote=F)

185 sink()
} # end temperatuur_transformatie_KNMI14

```

Listing B.2: R-function to apply tool to specific dataset

```

# EXAMPLE FILE TRANSFORMATION TOOL KNMI'14

# clean environment and set directory
rm(list = ls())
graphics.off()

# PRECIPITATION TRANSFORMATION
source("temperatuur_transformatie_KNMI14.R")

ifile="temperatuur_ref" # input file
ofile="uitvoer.txt" # output file (DEFAULT="uitvoer.txt")
delta.file=NA # file containing deltas
# if delta.file is not provided (DEFAULT)
# KNMI'14 deltas are used

# deltas are derived from KNMI'14 deltas if delta.file is not specified
# following arguments are used
sc="WH" # scenario ("GL", "GH", "WL", "WH")
p=2085 # time horizon (2030, 2050, 2085)
var="tg" # temperature characteristic ("tg" = mean,
# "tn" = min,
# "tx" = max)
regio.file="stationstabel" # table that links stations to region

temperatuur_transformatie_KNMI14(ifile=ifile,
ofile=ofile,
delta.file=delta.file,
sc=sc,
p=p,
var=var,
regio.file=regio.file)

```

Listing B.3: Example how to call R-function

C. R-scripts for transformation program Global Radiation¹⁴

```
#####  
#  
# rds_trans_KNMI14.R    March 11, 2015  
#  
5 # author: Alexander Bakker (KNMI)  
#  
# Function 'transforms' a specific reference-dataset with time series of daily  
# global radiation sums [kJ/m2] to a dataset representative for a future climate  
# scenario.  
10 #  
# Transformation developed for KNMI'14 climate change scenarios for the  
# Netherlands:  
#  
# Bakker, A. (2015), Time series transformation tool version 3.1: Description of  
15 # the program # to generate time series consistent with the KNMI'14 climate  
# scenarios, Technical Report TR-349, De Bilt, the Netherlands  
#  
# Version History:  
#   v3.0 - October, 2014  
20 #   v3.1 - March 11, 2015 [adjustment factor to estimate clear sky radiation]  
#  
# THIS CODE IS PROVIDED AS-IS WITH NO WARRANTY (NEITHER EXPLICIT  
# NOT IMPLICIT). KNMI SHARES THIS CODE IN HOPES THAT IT IS USEFUL,  
# BUT KNMI IS NOT LIABLE FOR THE BEHAVIOR OF THIS CODE IN YOUR OWN  
25 # APPLICATION. YOU ARE FREE TO SHARE THIS CODE SO LONG AS THE  
# AUTHOR(S) AND VERSION HISTORY REMAIN INTACT.  
#  
#  
# arguments:  
30 #  
# obs          data.frame or matrix:  
#               first column provides datestring YYYYMMDD  
#               other columns provide global radiation [kJ/m2] time series  
#               (each column represents specific station)  
35 #  
# deltas       data.frame or matrix that contains deltas (=change factors for  
#               the transformation) should contain following columns indicated  
#               by following headers  
#               HEADER  
40 #               "ave"          relative change [%] in mean shortwave surface  
#                               radiation  
#  
# lat          this argument provides a vector with latitudes belonging to
```

¹⁴Note that the online tool is based on Fortran and may provide slightly different time series.

```

#           stations. location (row.number [i]) in vector refers to column
#           in obs [i+1]
#
#####

rsds_trans_KNMI14 <- function(obs,
                             deltas,
                             lat) {

  version="v3.1"

# PREPARE DATA
# explore observations
  ns      <- ncol(obs) - 1           # number of stations
                                           # (= number of columns minus 1)
  mm      <- (obs[,1]%/%100)%%100    # the month that a day belongs to
                                           # (1, 2, ..., 12)
  nr      <- length(mm)             # total number of days
                                           # (in reference file)

  fut     <- obs; fut[,-1] = NA      # future values (filled with NA)

# FUNCTIONS USED FOR TRANSFORMATION

# transformation function
# X will not exceed Xmax
  tf <- function(a,X,Xmax) apply(cbind(a*X,Xmax),1,min)

# iterative estimation of coefficient a in function <tf>
  bounded.scaling <- function(X,Xmax,delta) {
    f <- function(a) mean(tf(a,X,Xmax)) - (1+delta)*mean(X)
    rc <- uniroot(f, lower=0, upper=4, tol = 0.01)
    a <- rc$root
    return(a)
  }

# "Angot" or "Top Of Atmosphere" radiation
  Angot <- function(datestring_YYYYMMDD,lat) {
    Gsc <- 0.0820 # solar constant [MJ/m2/min]
    phi <- pi*lat/180
    J <- daynumber(datestring_YYYYMMDD)
    dr <- 1 + 0.033*cos(2*pi*J/365)
    delta <- 0.409*sin(2*pi*J/365-1.39)
    omega <- acos(-tan(phi)*tan(delta))
    Ra <- (24*60/pi) * Gsc * dr*(omega *sin(phi)*sin(delta) +
                                sin(omega)*cos(phi)*cos(delta))

    return(Ra)
  }

# Derive daynumber within year (1-366)
  daynumber <- function(datestring_YYYYMMDD) {
    dpm <- c(0,31,59,90,120,151,181,212,243,273,304,334)
    id <- floor( datestring_YYYYMMDD %% 100)
    im <- floor((datestring_YYYYMMDD %% 10000) / 100)
    iy <- floor( datestring_YYYYMMDD / 10000) %% 4
    dnr <- dpm[im] + id + (iy==0 & im >2)
    return(dnr)
  }
}

```

```

105 # TRANSFORMATION
# apply transformation per station / time series <is> and calendar month <im>
for(is in 1:ns) {
  for(im in 1:12) {

    days.im      <- which(mm == im)    # all days within calendar month <im>
110    X          <- obs[days.im,is+1] # all obs in month <im> & station <is>

    # clear sky radiation for all days in month <im>
    # version v3.0 applied accidentally factor 0.7 (rather than 0.75)
115    Rx          <- 0.75 * 1000 * Angot(obs[days.im,1], lat[is])

    # determine coefficient a for transformation function
    # relative change of average in month <im>
    delta <- deltas[im,2]/100
    if(delta > 0) {
120      a <- bounded.scaling(X, Rx, delta)
    } else {
      a <- 1 + delta
    }
    Y <- tf(a, X, Rx)                    # transform
125    fut[days.im,is+1] <- round(Y,1)
  }
} # END TRANSFORMATION LOOP

return(fut)
130 } # end rsds_trans_KNMI14

```

Listing C.1: Radiation Transformation version 3.1

```

#####
#
# straling_transformatie_KNMI14.R      March 11, 2015
#
5 # author: Alexander Bakker (KNMI)
#
# Function reads 'reference data' with time series of daily global radiation
# sums [kJ/m2] and 'change factors' from input files and applies them to
# function 'rsds_trans_KNMI14' to obtain 'future time series' that match a
10 # certain climate
#
# Transformation developed for KNMI'14 climate change scenarios for the
# Netherlands:
#
15 # Bakker, A. (2015), Time series transformation tool version 3.1: Description of
# the program to generate time series consistent with the KNMI'14 climate
# scenarios, Technical Report TR-349, De Bilt, the Netherlands
#
#
20 # THIS CODE IS PROVIDED AS-IS WITH NO WARRANTY (NEITHER EXPLICIT
# NOT IMPLICIT). KNMI SHARES THIS CODE IN HOPES THAT IT IS USEFUL,
# BUT KNMI IS NOT LIABLE FOR THE BEHAVIOR OF THIS CODE IN YOUR OWN
# APPLICATION. YOU ARE FREE TO SHARE THIS CODE SO LONG AS THE
# AUTHOR(S) AND VERSION HISTORY REMAIN INTACT.
25 #
#
# arguments:
#
# ifile          Name of the input file (ASCII) that contains reference data

```



```

30 # (all numerics) in which the columns provide time series for
# specific stations. The first column should provide either
# 00000000 or a datestring YYYYMMDD: Rows starting with 00000000
# are considered station info (station number, lat, lon
# etc.) and are ignored. Rows starting with a datestring refer to
35 # a specific day in the time series. Rows starting with "#" are
# completely ignored and returned unchanged
#
# NOTE !!!
# FIFTH row indicated by 00000000 is interpreted to contain
40 # LATITUDES of station
#
# ofile (DEFAULT="uitvoer.txt") Name of the output file to write the
# transformed data to. Format is similar to ifile
45 #
# delta.file [optional] Name of file that contains deltas (changes factors
# for the transformation). File should contain following
# compulsory column identified with compulsory headers
# - HEADER -
50 # "ave" relative change [%] in mean shortwave surface
# radiation
#
# sc scenario ["GL", "GH", "WL", "WH"]
# p time horizon [2030 (=DEFAULT), 2050, 2085]
55 #
#####

straling_transformatie_KNMI14 <- function(ifile,
60 ofile="uitvoer.txt",
delta.file,
sc,
p=NA) {

# CONSTANTS AND FUNCTIONS #####
65 version="v3.1"

# READ REFERENCE DATA FROM ifile
# select lines with "#" from reference file and ignore them
70 # (only necessary for output file)
H.comments <- scan(ifile, character(0), sep = "\n")
H.comments <- H.comments[grep("#",H.comments)]

obs <- read.table(ifile,header=F) # read reference data
75 header <- obs[which(obs[,1]==0),] # header met stations
# meta-data etc.

header[,1] <- "00000000"
names(obs) <- c("date",round(obs[1,-1],0)) # station names are read
# from first line
80 obs <- obs[which(obs[,1]!=0),] # actual data

lat <- as.numeric(header[5,-1])

# READ CHANGE FACTORS (DELTAS)
85 if(!is.na(delta.file)) {
deltas <- read.table(delta.file,header=T) # deltas are provided in
# file <delta.file>
} else {
if(p=="2030") {

```

```

90     # 2030 decadal prediction if p=2030
    deltas <- read.table(      "deltas-KNMI14__rsds_____2030.txt",header=T)
  } else {

    # str.ext is a function used to construct file.names if delta.file is not
95    # explicitly provided
    str.ext <- function(var,ch,n) {
      paste(var,substr(paste(rep(ch,n),collapse=""),1,n-nchar(var)),sep="") }

    deltas <- read.table(paste("deltas-KNMI14__rsds_",
100      str.ext( sc,"_",3),"_",
      str.ext( p,"_",4),".txt",sep=""),header=T)

  }
}

105 # TRANSFORMATION
source("rsds_trans_3.1_KNMI14.R")
fut <- rsds_trans_KNMI14(obs=obs, deltas=deltas, lat=lat)

110 # OUTPUT #####
sink(ofile)

# comments
writeLines("# Transformed daily mean global radiation [kJ/m2] according to KNMI
'14 transformation tool,")
115 writeLines(paste("# version ",version,sep=""))
if(is.na(p)) {
  writeLines(paste("# Deltas are derived from ",sc,sep=""))
} else {
  if(p=="2030") {
120    writeLines("# Deltas are derived from the 2030 decadal prediction")
  } else {
    writeLines(paste("# Deltas are derived from ",sc," scenario",sep=""))
    writeLines(paste("# around the time horizon ",p,sep=""))
  }
}
125 writeLines("#")
writeLines("# Bakker A., 2014. Time series transformation tool: description of
the program to")
writeLines("# generate time series consistent with the KNMI '14 climate
scenarios, TR-349")
writeLines("#")
130 for(i in 1:length(H.comments)) writeLines(H.comments[i])
writeLines("#")

# header
write.table(format(header[1,],
135      width=10,
      justify="right"),
      row.names=F,
      col.names=F,
      quote=F)
write.table(format(header[-1,],
140      width=10,
      justify="right"),
      row.names=F,
      col.names=F,
      quote=F)
145 # transformed data

```

```

write.table(format(fut,
                  width=10,
                  digits=2,
150                  justify="right"),
            row.names=F,
            col.names=F,
            quote=F)

155 sink()

} # end straling_transformatie_KNMI14

```

Listing C.2: R-function to apply tool to specific dataset

```

# EXAMPLE FILE TRANSFORMATION TOOL KNMI'14

# clean environment and set directory
rm(list = ls())
5 graphics.off()

# PRECIPITATION TRANSFORMATION
source("straling_transformatie_KNMI14.R")

10 ifile="straling_ref"      # input file
   ofile="uitvoer.txt"     # output file (DEFAULT="uitvoer.txt")
   delta.file=NA           # file containing deltas
                           # if delta.file is not provided (DEFAULT)
                           # KNMI'14 deltas are used

15 # deltas are derived from KNMI'14 deltas if delta.file is not specified
   # following arguments are used
sc="WH"                   # scenario ("GL", "GH", "WL", "WH")
p=2085                    # time horizon (2030, 2050, 2085)

20 straling_transformatie_KNMI14(ifile=ifile,
                                ofile=ofile,
                                delta.file=delta.file,
                                sc=sc,
25                                p=p)

```

Listing C.3: Example how to call R-function