Advanced Delta Change Report

Vera Vavan

13/01/2022

Introduction

Advanced delta change is a method which produces a climate change scenario based on the real observed data and simulated data. It can be performed on precipitation and temperature data, with different calculations. AdvDeltaChange package in R allows automatic calculations of all parameters and visualizes the climate change scenario. The calculations are performed on data from Klementinum.

Method and results

Firstly, the package needs to be downloaded from GitHub which is done by installing the devtools package if it is not already installed, and then running the code:

```
library(devtools)
#install_github("veravavan/AdvDeltaChange")
library(AdvDeltaChange)
```

The data used in this report comes from the data within the package. The source for observed data about precipitation and temperature from Klementinum is this website, and it is saved in the package as pr and tas. The simulated data about precipitation and temperature is from the second generation Canadian Earth System Model (CanESM2) downloaded through their website, and saved within the package as pr_sim and tas_sim. Other data that is not within the package can be used as well, however it must follow certain guidelines described within the function documentation. Besides the data, the method itself was constructed following the paper from Pelt et al. (2012).

If data from different locations need to be used, the first step would be extracting the simulated data from the .nc file by using the function rastr_extract() from the package and reading the observed data. However, certain system requirements are needed for large extractions, which is why this report will use already extracted data within the package.

To start using the method, the data needs to be saved within the environment using the following code:

```
data('pr')
data('pr_sim')
```

The data needs to be in a certain format so that the parameters for the method can be calculated. Firstly, the simulated precipitation values need to be summed into five day periods and the missing values need to be removed, which is done with the function frollsum():

```
pr_sim[, pr5 := 60*60*24*frollsum(pr, 5, align = 'c')]
pr_sim = pr_sim[!is.na(pr5)]
```

Then the control and scenario periods need to be defined, where the control represents the past and scenario the future. The control period used for this data is 1981-2010 and the scenario period is 2071-2100. When the data is restricted to these periods, the 60% and 90% quantiles need to be found for both periods in order to find the parameters:

```
pr_ctrl = pr_sim[year(DTM) %in% 1981:2010]
pr_scen = pr_sim[year(DTM) %in% 2071:2100]

ctrl_q = quantile(pr_ctrl$pr5, c(.6,.9))
scen_q = quantile(pr_scen$pr5, c(.6,.9))
ctrl_q
```

```
##      60%      90%
## 12.55315 28.90034
```

```
scen_q
```

```
##      60%      90%
## 13.33921 32.55411
```

The transformation can be performed automatically with the function pr_trasf(), but the parameters are not visible that way and they can be obtained by using the following functions.

If the method is performed without bias correction, this is enough to calculate the parameters with the functions pr_a_nb and pr_b_nb, but for bias correction, the quantiles for observed data within the control period needs to be found as well:

```
pr[, pr_o5 := frollsum(pr_o, 5)]
pr = pr[year(DTM) %in% 1981:2010]
obs_q = quantile(pr$pr_o5, c(.6,.9))
```

Now the a and b parameters can be found using the pr_b and pr_a functions, but the parameter b must be found first since it is needed for parameter a calculation:

```
b <- pr_b(scen_q, ctrl_q, obs_q)
a <- pr_a(scen_q, ctrl_q, obs_q, b)
a[[1]]
```

```
## [1] 0.9890466
```

```
b[[1]]
```

```
## [1] 1.047017
```

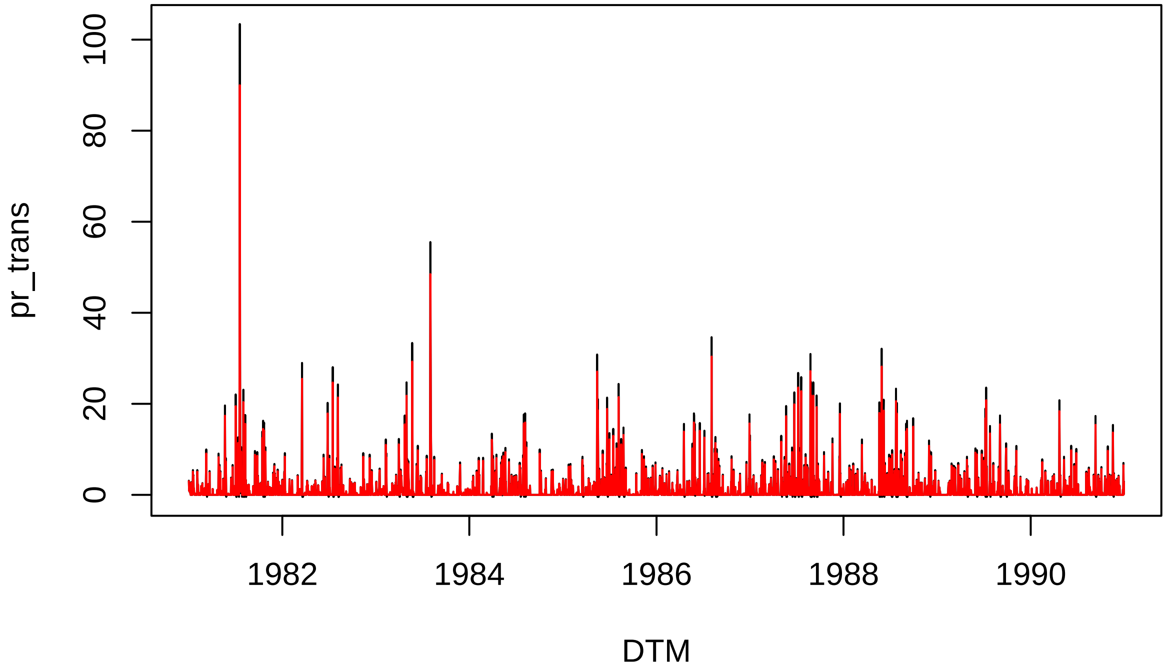
After finding these parameters and quantiles, the transformation can be easily performed by using the function pr_transf() from the package. The function can find the parameters automatically, but they can also be supplied:

```
pr_trans <- pr_transf(pr_ctrl, pr_scen, ctrl_q, scen_q, pr, obs_q)
head(pr_trans)
```

```
##      SOUID     DATE RR Q_RR pr_o        DTM pr_o5  pr_trans
## 1: 100082 19810101 30    0  3.0 1981-01-01   3.0 3.1244302
## 2: 100082 19810102  9    0  0.9 1981-01-02   3.9 0.8857433
## 3: 100082 19810103 11    0  1.1 1981-01-03   5.0 1.0928375
## 4: 100082 19810104  7    0  0.7 1981-01-04   5.7 0.6808192
## 5: 100082 19810105 15    0  1.5 1981-01-05   7.2 1.5121236
## 6: 100082 19810106 11    0  1.1 1981-01-06   5.3 1.0928375
```

However, to understand the data, a table is not enough and it can be visualized using the function pr_plot():

```
pr_plot(pr_trans)
```



```
## NULL
```

The black lines on the graph above represent the transformed data series, while the red lines represent the observed data supplied.

As the precipitation data is found, the next step is to perform the temperature transformation, which is much more simple as there are no parameters. Firstly, the data needs to be found and cleaned up, finding the same control and scenario periods, but no quantiles:

```
data("tas_sim")
data("tas")

tas_sim[, tas := tas-273.15]
```

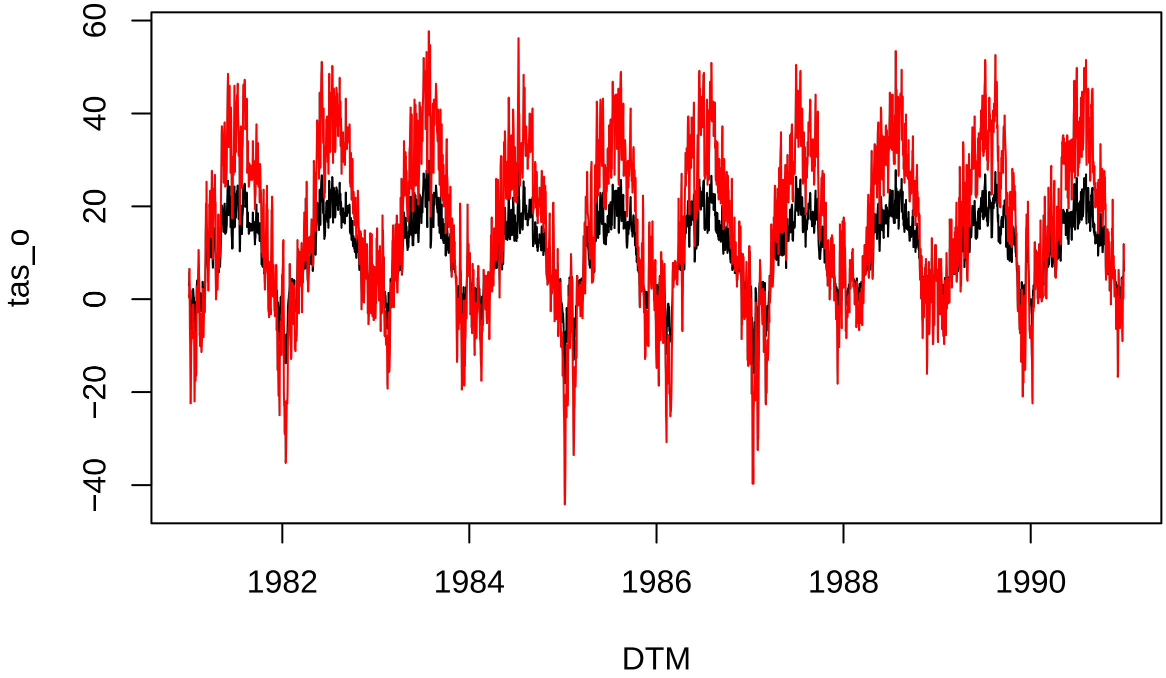
```
tas_ctrl = tas_sim[year(DTM) %in% 1981:2010]
tas_scen = tas_sim[year(DTM) %in% 2071:2100]

tas_trans <- tas_transf(tas, tas_scen, tas_ctrl)
head(tas_trans)
```

```
##     SOUID     DATE  TG Q_TG tas_o        DTM tas_mean_trans  tas_trans
## 1: 100080 17750101 -70    0  -7.0 1775-01-01      -1.770778 -20.722647
## 2: 100080 17750102 -22    0  -2.2 1775-01-02       3.029222 -10.494422
## 3: 100080 17750103 -10    0  -1.0 1775-01-03       4.229222  -7.937366
## 4: 100080 17750104   1    0   0.1 1775-01-04       5.329222  -5.593397
## 5: 100080 17750105  22    0   2.2 1775-01-05       7.429222  -1.118549
## 6: 100080 17750106  32    0   3.2 1775-01-06       8.429222   1.012331
```

And the visualization of the temperature data can be done in the same way as precipitation:

```
tas_plot(tas_trans)
```



```
## NULL
```

The black lines represent the observed true series, while the red one represents the transformed data.

Aside from the individual calculations, the whole method can be done using just two functions by just supplying the data:

```r
data_sim <- data('pr_sim')
data_obs <- data('pr')
pr_trans <- adc_pr(data_sim, data_obs)
head(pr_trans)
```

```
##     SOUID     DATE RR Q_RR pr_o        DTM pr_o5  pr_trans
## 1: 100082 19810101 30    0  3.0 1981-01-01   3.0 3.1244302
## 2: 100082 19810102  9    0  0.9 1981-01-02   3.9 0.8857433
## 3: 100082 19810103 11    0  1.1 1981-01-03   5.0 1.0928375
## 4: 100082 19810104  7    0  0.7 1981-01-04   5.7 0.6808192
## 5: 100082 19810105 15    0  1.5 1981-01-05   7.2 1.5121236
## 6: 100082 19810106 11    0  1.1 1981-01-06   5.3 1.0928375
```

```r
data_sim <- data('tas_sim')
data_obs <- data('tas')
tas_trans <- adc_tas(data_sim, data_obs)
head(tas_trans)
```

```
##     SOUID     DATE  TG Q_TG tas_o        DTM tas_mean_trans  tas_trans
## 1: 100080 17750101 -70    0  -7.0 1775-01-01      -1.770778 -20.722647
## 2: 100080 17750102 -22    0  -2.2 1775-01-02       3.029222 -10.494422
## 3: 100080 17750103 -10    0  -1.0 1775-01-03       4.229222  -7.937366
## 4: 100080 17750104   1    0   0.1 1775-01-04       5.329222  -5.593397
## 5: 100080 17750105  22    0   2.2 1775-01-05       7.429222  -1.118549
## 6: 100080 17750106  32    0   3.2 1775-01-06       8.429222   1.012331
```

As it can be seen, the results are identical as when performing individual steps, now the transformation just needs to be visualized.

Discussion and conclusion

The plots produced by using the functions from the AdvDeltaChange package show how well this method is able to transform the observed data with the simulated data. However, the precipitation transformation is much less extreme than the temperature transformations, as the temperature varies more and the impact on it is higher. Precipitation on the other hand varies less, but the extremes are even more extreme than the observed data.

Bibliography

Pelt, S. C. van, J. J. Beersma, T. A. Buishand, B. J. J. M. van den Hurk, and P. Kabat. 2012. "Future Changes in Extreme Precipitation in the Rhine Basin Based on Global and Regional Climate Model Simulations." *Hydrology and Earth System Sciences* 16 (12): 4517–30. https://doi.org/10.5194/hess-16-4517-2012.