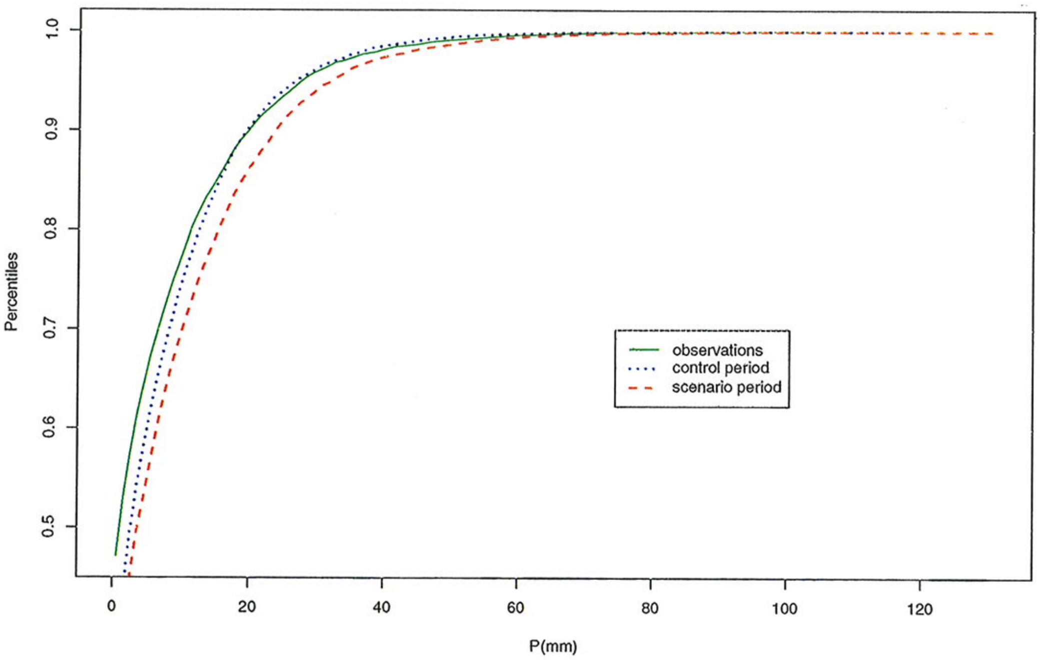
Meteorologisk institutt
met.no

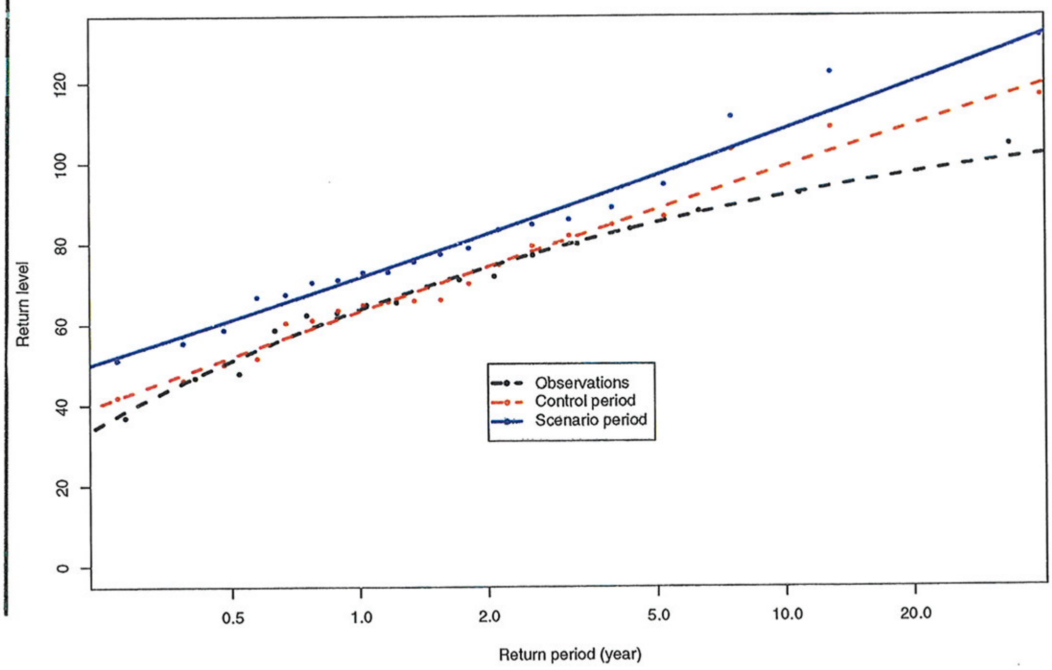
Report 16/02

KLIMA

Application of extreme values to study climate change

Alexandra Imbert





met.no REPORT

NORWEGIAN METEOROLOGICAL INSTITUTE
BOX 43 BLINDERN , N - 0313 OSLO, NORWAY

PHONE +47 22 96 30 00

ISSN 0805-9918

REPORT NO.
16/02 KLIMA

DATE
18.09.2002

TITLE:

Application of extreme values to study climate change

AUTHORS:

Alexandra Imbert*

* Present affiliation: INSA Engineering Course in Mathematical Modelling,
135 Avenue de Rangueil, 31077 – Toulouse Cedex 4, France

PROJECT CONTRACTORS:

Norwegian Meteorological Institute
The Research Council of Norway (Contract no. 120656/720 RegClim)
Norwegian Electricity Industry Association (Contract no. H1.00.5.0)

SUMMARY:
This report is a documentation of statistical analyses performed during a 10 weeks training period at the Climatology Division.

Frequencies and extremes of daily values of temperature and precipitation were analysed for more than 30 locations in Norway. The analyses comprised both observations, as well as downscaled climate model values for a control and a scenario period. General statistical tests were performed for changes in mean values and variances, as well as for changes in percentiles. Extreme values were studied both by General Extreme Value and Pareto distributions. The analyses were performed by use of the R-programming package, and an important task was to provide a "users manual" for supplementary analyses at met.no. Detailed descriptions and examples of use of the R-tools for extreme value analyses are included in the present report.

KEYWORDS:

Extremes, Temperature, Precipitation, Extreme value distributions

SIGNATURES:

Eirik J. Førland
Eirik J. Førland
Head of section for Climate Research

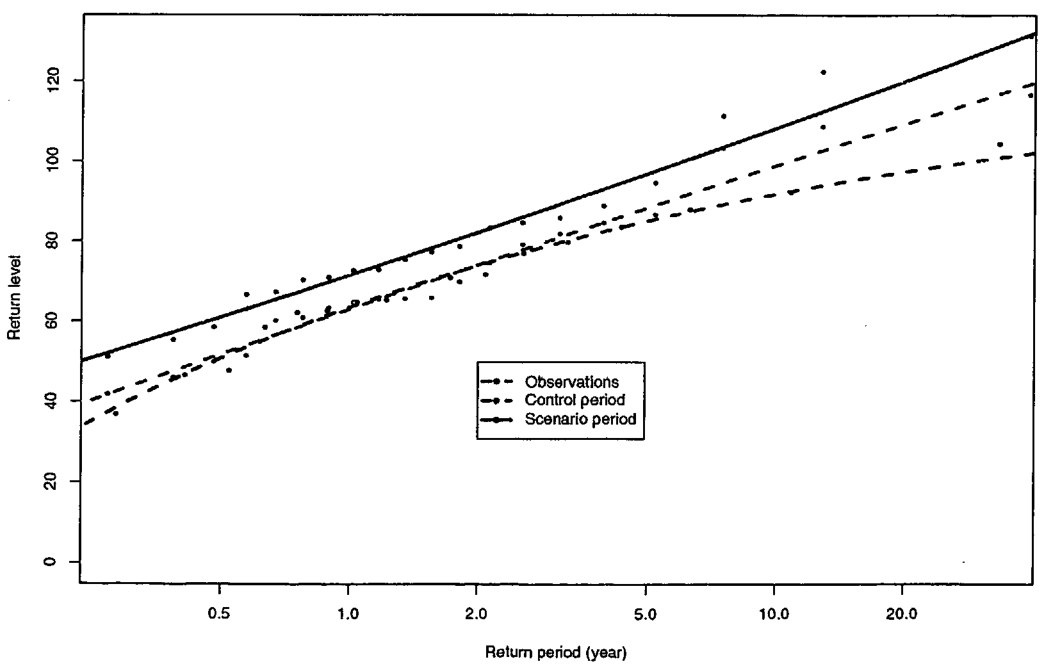
Bjørn Aune
Bjørn Aune
Head of Climatology Division

Project at the Norwegian Meteorological Institute
Climatology Research

headed by Eirik J. Førland, Section Leader

Application of extreme values to study climate change

Return Level Plot



Alexandra IMBERT

10th of June 2002 - 30th of August 2002

Contents

Acknowledgements

Lots of people have contributed to make my work at the Meteorological Institute a very learning and nice experience. In particular, I would like to mention Eirik J. Førland, Section Leader, Climate Research who supervised this project and made several helpful suggestions. I also do thank Rasmus E.Benestad who helped me so much to know more about R-package and with whom I worked a lot on GPD routines, P. Øyvind Nordli who gave me basics on Norway geography, Ole E. Tveito to welcome me during the first week, and to so many people like Torill or Inger, at the Climatology division, for being so kind and warmful. I am very grateful to all this people.

Introduction

Changes in severe weather due to climate change will have particular impacts on society and the natural environment. Hence the importance of understanding the mechanism of extreme weather events and, if possible, projecting future changes. The impacts of climate change will be particularly felt through changes in extreme events because they will stress or exceed our present day adaptions to climate variablity.
Changes in temperature and in the amount of rainfall have been used as an indicator of climate change. The type of extreme events considered are not only higher maximum temperatures and more hot days as well as fewer cold days, but also more intense precipitation events.

This ten week-project was headed by my supervisor, **Eirik J. Førland, Section Leader, Climate Research** at the Norwegian Meteorological Institute in Oslo (*met.no*).
The objectives were to compute analyses of the statistical properties of precipitation and temperature and to evaluate appropriate extreme value distributions in order to study the frequency and intensity of these extreme events in the 21st century. The analyses were performed on the R-programming package, the aim being to prepare a 'users manual' for supplementary analyses.
This study based on observations from met.no's Climate Database permits to adapt different models of extreme values and to compare the results.

After a description of the Meteorological Institute and of the dataset, it will be interesting to present the two extreme value distributions used during this project. The last part of the report will present the routines implemented as well as the results which will be illustrated on one selected station.

Part I

Presentation of the dataset

0.1 Data

The data are a combination of daily temperatures (T) in Celsius degrees and rainfall accumulation (P) in millimiter for 50 stations in Norway (cf. station map 0.2).
The dataset consists in three different types of data:
- **observations** from 1980 to 1999;
- interpolated values from **'Control time-slice'** for the period 1980-1999;
- interpolated values from **'Scenario time-slice'** for the period 2030-2049.

Each type of files is stored in a different directory.
Observations are present in **obs80-99-txt** directory, the control period files are in **present-txt** whereas the files for the scenario period are in **scenario-txt**.
The data are presented in columns, the first column being the number of the station, the second one the day then the month, the year and the two last columns being for precipitation (rr) and for temperature (temp).

0.2 Modifications in the dataset

The first step was to import the text files into R, which required to define a fix format for all of them.
First of all, it appeared that the two last columns were not systematically in the same order. In some files, temperatures were written in the first column and in others, precipitations came first. This led to implement a R-routine to check which of the two columns held precipitations.
This was made possible by implementing a R-routine called **check.data**.

Moreover, the separators between the different columns were not of the same kind, some of them being tabulations, others being white space or dots. As a consequence, the columns were not of the same width.
It has then been necessary to create several R-functions to write in the same way all the files from a same directory. This is summed up in table 1. This was important, not only to permit to import the files into R but also to be able, in a second step, to create a generic procedure which would treat all the files iteratively.
A preliminary work was then to replace all the tabulations by white spaces and to separate the values between two consecutive columns by the same number of white spaces.

When considering the files and in the view of computing statistical tests, other changes had to be made. Several notations which were used in the original files for the variable precipitation also had to be modified before the files could get into R.

Firstly, dots were used instead of '0.0' and then they had to be replaced by their true meaning. Indeed, if not associated with a number, dots are not meaningful and cannot be read by R.

In the same way, '0' actually represented a trace of precipitation (0.01 mm). As for the previous case, this modification was of a great importance as it affects the statistical results. It was once more realized by implementing two R-routines, the first one to determine the index of the precipitation column (**find.substr**) in order to make the changes in that column only (and not in the columns for temperatures) and the second one to make the substitution (**change.0**).

Finally, it can also be alluded to missing values indicated by '-'. They were changed in 'NA' which is a more explicite notation, again by writting a R-procedure.

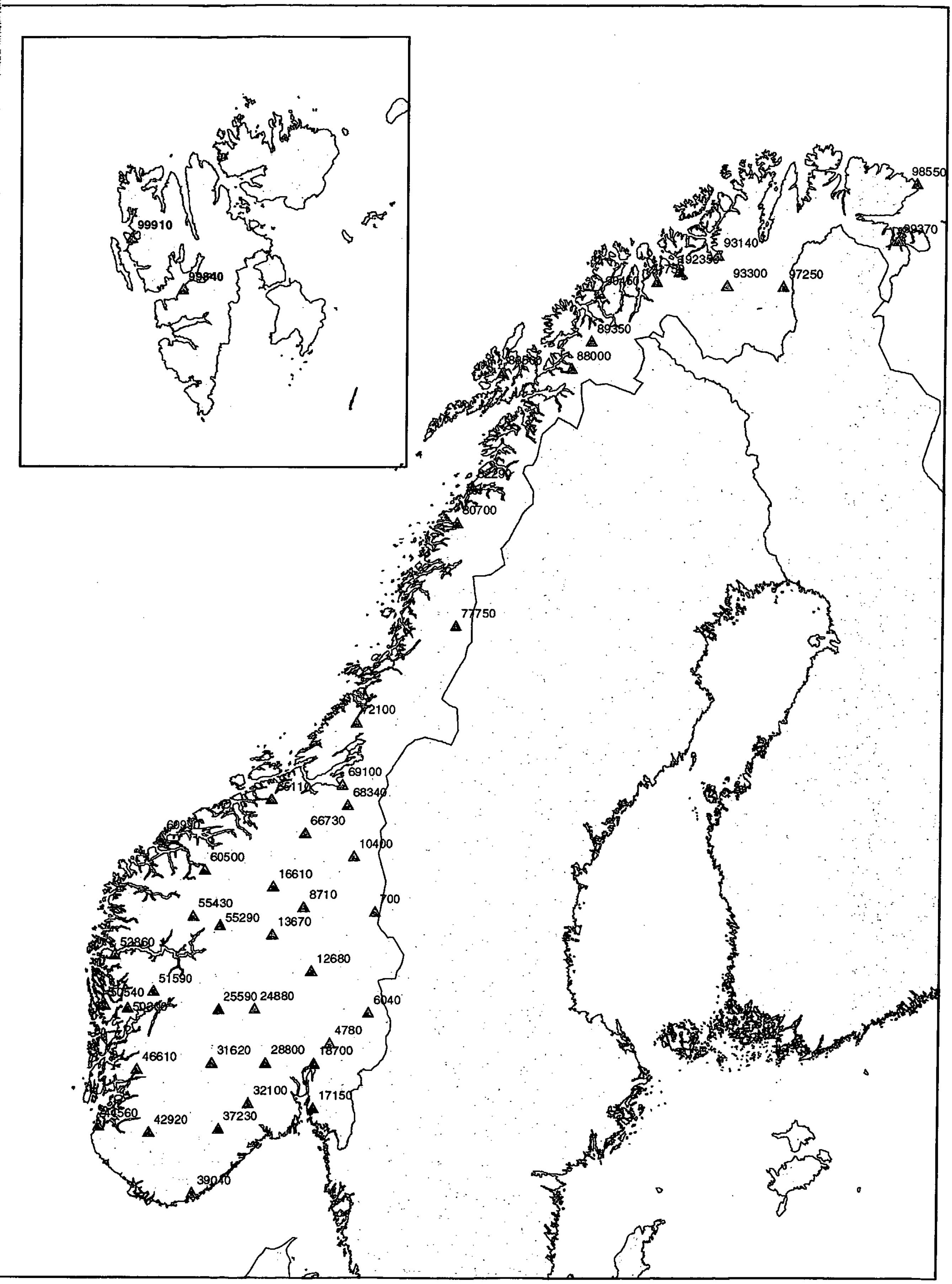
Object of the modification	**Before modification**	**After modification**	**R-routine implemente**
files of **obs80-99-txt** directory	columns not in the same order	P before T	check.data
P	.	0.0	fix.files
P	**0**	0.01	find.substr and change.0
P and T	-	NA	fix.files

Table 1: Changes realized in the files. (*Legend:* ***T*** *stands for Temperature,* ***P*** *for Precipitation*)

Example of the dataset presentation after the modifications

```
12680 01 12 1982 -7.8 NA
12680 02 12 1982 -7 0.0
12680 03 12 1982 -4 0.3
12680 04 12 1983 -13.3 0.01
```

Next page: (map 3) **Location of the meteorological stations in Norway used in this study**


99910
99840
98550
99370
93140
92350
93300
97250
89350
88000
80700
77750
72100
69100
68340
66730
10400
60500
16610
8710
700
55430
55290
13670
52860
12680
51590
25590
24880
6040
4780
46610
31620
28800
18700
32100
17150
42920
37230
39040

Part II

Extreme value theory

Chapter 1

GEV Distribution

Many practical problems encountered in climatololgy require to make inferences about the extremes of a probability distribution.
Extreme value analysis is the branch of probability and statistics that is used to make inferences about the size and frequency of extreme events. The basic paradigm use varies with application but generally has the following components:
- data gathering;
- identification of a suitable family of probability distributions, one of which is to be used to represent the distribution of the observed extremes;
- estimation of the parameters of the selected model;
- estimation of return levels for periods of fixed length. Return level values are thresholds which are exceeded, on average, once per return period.

On this study, it will be concentrated on two specific methods. One is the Generalized Extreme Value Distribution (GEV), the second one is the use of the Pareto Generalized Distribution (GPD) on daily maximum for temperature and precipitation as well as on daily temperature minimum.

The GEV method only takes into account a block of maximum, that-is-to say the maximum for each year.
Suppose that $X_1, X_2, ...X_n$ is a sequence of i.i.d random variables with distribution function F.
The behaviour of extremes is by considering the behaviour of the maximum order statistic:

$$M_n = max\{X_1, X_2, ...X_n\}.$$

Thus,

$$P\{M_n <= x\} = (F(x))^n.$$

The difficulty comes from the fact that F is unknown and the idea is then to look for models which approximate F^n.
As $F(x)^n \to 1$ when $n \to \infty$, the function F^n is degenerated.

As a consequence, in order to get a limite for the distribution M_n, it is better to consider $M_n^* = \frac{M_n - b_n}{a_n}$, where a_n et b_n are normalizing coefficients.

Definition : Two distributions F and F^* are said to be of the **same type** if it can be found two constants a et b such as $F^*(ax+b) = F(x) \quad \forall x$

The theorem of extreme types permits then to conclude that, for $a_n > 0$ and b_n,

if $P\left[\frac{M_n - b_n}{a_n} \leq x\right] \xrightarrow[n\to\infty]{} G(x)$,

then $G(x)$ which is the non-degenerated cumulative function, is of the same type as:

$$G(x) = exp-\left[1+\xi\frac{(x-\mu)}{\sigma}\right]^{-\frac{1}{\xi}}$$

defined above $\{x/1+\xi\frac{x-\mu}{\sigma}>0\}$
and with - ξ: **shape** parameter;
- $\sigma(>0)$: **scale** parameter;
- μ: **location** parameter.

Note : In fact, this expression is a generalization of three types of extreme values distributions called **Gumbel** (if $\xi \to 0$), **Fréchet** (if $\xi > 0$) and **Weibull** (if $\xi < 0$). They can respectively be written as:

I: $G(x) = exp\{-exp(-\frac{x-b}{a})\} \qquad -\infty < x < \infty;$

II: $G(x) = \begin{cases} 0 & x \leq b \\ exp(-\frac{x-b}{a})^{-\alpha} & x > b,\ \alpha > 0; \end{cases}$

III: $G(x) = \begin{cases} exp\{-(-\frac{x-b}{a})^{-\alpha}\} & x < b, \alpha > 0; \\ G(x) = 1 & x > b. \end{cases}$

An estimation of the three parameters μ, σ and ξ can be obtained using the maximum likelihood estimation.

Then, as R.L Smith develops it (ref. [4]):
- they are **regular** if $\xi > -0.5$, which is true in most of the cases;
- they exist if $-1 < \xi < -0.5$ but are **not regular**;
- they **do not exist** i $\xi > -1$.

For studying extremes in data, it is necessary to have as long records as possible. Ideally, these should be at least 100 years.
Observed records of precipitations and temperatures are quite short. Keeping only 20 maxima (one for each year of the observed period) does not really represent a wide range of data and it can be wondered about the significance of the results. An alternative to study extremes is then to consider all the values above a threshold rather than annual maxima, which is realized in the **Pareto method**.

Chapter 2

Pareto distribution

Threshold methods present the advantage of taking into account more data than for maximum methods as all the values over the threshold count for the analysis.

$X_1, X_2, ..., X_n$ are n i.i.d random values distributed as X with distribution function F. Normalizing constants a_n et b_n can be found, so that

$$Pr\{(M_n - b_n)/a_n \leq x\} \to G(x)$$

with

$$G(x) = exp\left\{-\left[1 - \xi\left(\frac{x-\mu}{\sigma}\right)\right]^{-1/\xi}\right\}$$

For u high enough, the density of $X - u|X > u$ can be approximated by

$$H(x) = 1 - \left[1 + \frac{\xi x}{\widetilde{\sigma}}\right]^{-1/\xi}$$

defined on $\{y/\ y > 0 \text{ et } 1 + (\xi x)/\widetilde{\sigma} > 0\}$.
Therefore,

$$Pr(X > u + x|X > u) = \left[1 + \frac{\xi x}{\widetilde{\sigma}}\right]^{-1/\xi}$$

This is the Generalised Pareto Distribution (GPD).
It can be noticed that, unlike maximum methods, the GPD only has two parameters: $\widetilde{\sigma}$ and ξ.

<u>**Note**</u>:
The expression of GEV can be obtained from this expression by writting:

$$\widetilde{\sigma} = \sigma + \xi(u - \mu)$$

$$\begin{aligned} GPD(x;\sigma,\xi) &= 1 - (1 + x\xi/\sigma)^{-1/\xi} \qquad \xi \neq 0 \\ &= 1 - exp\{-x/\sigma\} \qquad \xi = 0 \end{aligned}$$

where $x > 0$
ξ is the **shape** parameter and $\sigma > 0$ is the **scale** parameter.
For $\xi \geq 0$, $0 \leq x < \infty$ and for $\xi < 0$, $0 \leq x < -\sigma/\xi$.

<u>Estimation of the parameters</u>

Extreme value theory to model in extremes is a tail modelling approach which has many potential advantages over existing descriptive approaches of extremes.
Once a model (i.e extreme value distribution) has been selected, the next step in the analysis is

to 'fit' the chosen extreme value distribution to the sample of extremes. Fitting means estimating the unknown parameters of the chosen extreme value distribution.

Several methods may be used for parameter estimation. The methods most often used for fitting are:

- the method based on Mean Excess Function;
- the method of moments;
- the method of probability weigthed moments;
- the method of maximum likelihood;

GEV distribution as it is presented in this report is based on the method of maximum likelihood whereas GPD model is determined from mean excess function.

Chapter 3

Return Levels

The last step in an extreme value analysis is usually to compute **return levels** for present periods (e.g. 10, 50, 100 years). These values are thresholds that, according to the fitted model, will be exceeded on average once every period.
Return values are simply the upper quantiles of the fitted extreme value distribution.
In general, the T-year return value for the annual maximum, say Y(T), is the solution of: $\int_{Y(T)}^{\infty} f(y)dy = 1/T$.

The quantiles correspond to the values of x_p defined by the relation $P[X \leq x_p] = p$. In extreme value theory, they correspond to the return level associated to the period $1/p$.

An estimation of the return period can easily be obtained by reversing the cumulative function G.
Thus,

$$x_p = \mu - \frac{\sigma}{\xi}\left[1 - \{-\log(1-p)\}^{-\xi}\right]$$

with $G(x_p) = 1 - p$.

If $\xi < 0$, the quantiles x_p have a upper limit and the curve is concave. On the contrary, if $\xi \geq 0$, the quantiles are no longer bounded.
Then, the shape parameter permits to determine wether or not the curve has a upper limit and so if there is a risk of reaching an extreme value never observed so far.

Part III

Applications and Results

Chapter 4

Presentation of the R-functions

4.1 fix.files.R

This code contains four functions which permit to **format** all the data files in the same way.

4.1.1 avail.files

This function gets the name of the files stored on the disc in a specified directory in the arguments of the routine. The routine filters the names by picking names that match two patterns. It returns the names in a vector of characters.
The **arguments** are:

- **direc**: specifies the directory where the files are stored;
- **pattern1**: suffixe of the file;
- **pattern2**: pattern written after the number of the station.

For instance, the command
> avail.files("present-txt",".txt","-present")

will give of the files contained in "present-txt" directory, as follows:

```
[1] "10400-present.txt" "12680-present.txt" "13670-present.txt"
 "16610-present.txt" "17150-present.txt" "18700-present.txt"
 "24880-present.txt" "25590-present.txt" "28800-present.txt"
 ...
```

4.1.2 fix.files

This routine aims at replacing all the missing values by NA and at separating all the columns by one white space.
The datasets produced are written in a file, having **.fix** as a suffixe. The arguments are:

- **files**: gives the pattern to search for in the names of the files;
- **direc**: names of the directory for the files which need to be fixed.

Example of call:
dumb <- fix.files()

4.1.3 check.columns

The reading of the observed data is made in this function, which also determines which of the two last columns contains temperatures. This is realized by calling the procedure **check.data.R**. There is only one argument to that function, which is the number of the station considered, written between quotation marks.
Example of call:
> *check.columns("99910")*

4.1.4 change.0

In that function without any arguments, all the 0 values present in the precipitation column (notation rr) are replaced by what they really stand for, that-is-to-say for a rainfall equal to 0.01, so that there cannot be a confusion with dry days refered as 0.0.
This implies to know the index of the **rr** column, which can be made by determining the position of 0.0, only present in this column.
The new files are identified by the suffixe **.ok**.
Example of call:
> *change.0()*

4.2 check.data.R

This function checks which of the two vectors contains rainfall (daily values) and which holds temperatures. The decision is based on statistical properties and five different tests. If the tests do not permit to conclude, the summaries of the two columns are printed on the screen, and the user is aked to make a choice.
The result is only store into **R** and not in a new file.
The two columns to identify are the unique arguments of this function.
Example of call:
> *check.data(st.obs$rr, st.obs$temp)*

4.3 find.substr.R

This routine searches the string for a matching pattern that may be embedded somewhere in the string. It is used as a support routine for **change.0** to determine the index of the column containing precipitation data.
The arguments are:

- **pattern**: string to be found;
- **in.string**: expression in which to search the pattern.

It returns a boolean indicating whether or not the search has been successful and if it is true, gives the position of all the occurences of the string.
Example of call:
> *find.substr("0.0",line)*

4.4 rain_study.R

This code computes the first type of statistical tests (i.e means, variances, percentiles).

4.4.1 mean.study

Annual, seasonal and monthly mean values are here calulated. The function has the following arguments:

- **data**: R-object corresponding to the analysed dataset;
- **rr**: if true, tests are made on precipitations. Otherwise, temperatures are taken into account.

The function returns a list containing the anlysed variable (rr or temp) as well as the different means.
Example of call:
> *mean.study(st.obs,rr=F)*

4.4.2 var.study

This function determines annual, seasonal and monthly variances.
Its structure being the same as the previous one, only an example of call is given here:
Example of call:
> *var.study(st.obs,rr=F)*

4.4.3 detail.means

This function realizes the same as **mean.study** but, instead of keeping one maximum on the whole period, it stores the means for each year. It is used as a support routine for **plot.means**.
Once more, the arguments are the same ones as for **mean.study**.
Example of call:
> *detail.means(st.obs,rr=F)*

4.4.4 student.comparison

This routine permits to performs a **Student test** (t-test) on a vector containing the difference of two means.
These arguments are:

- **data1**: dataset for the first mean
- **data2**: dataset for the second mean (H0: mean(data1)==mean(data2))
- **conf**: confidence level for the test

The routine returns a list containing the p-values for the test, when considering yearly means as well as seasonal means.
Example of call:
> *student.comparison(st.obs,st.ctr,conf=0.95)*

4.4.5 wilcoxon.comparison

This routine performs a Wilcoxon test on a vector containing the difference of two means.
These arguments are:

- **data1**: dataset for the first mean
- **data2**: dataset for the second mean (H0: mean(data1)==mean(data2))
- **rr**: if true, the study is on precipitations. If false, the study is on temperatures.
- **conf**: confidence level for the test
- **var**: if true, variance comparison is computed. If false mean comparison is computed.

As previously, a list containing the p-values for the test, when considering yearly means as well as seasonal means is returned.
Example of call:
> *wilcoxon.comparison(st.obs,st.ctr,rr=T, conf=0.99,var=F)*

4.4.6 fisher.comparison

Here is computed a Fisher test to test the equality of two variances from a normal distribution. These arguments are:

- **data1**: population (dataset) for the first variance
- **data2**: population (dataset) for the second variance (H0: variance(data1)==variance(data2))
- **conf**: confidence level for the test

The ouptut has the same structure as the previous routine.
Example of call:
> *fisher.comparison(st.sc,st.ctr,0.95)*

4.4.7 compare.plot

Here, plots are performed to compare observed annual mean values and modelled annual mean values. A barplot is produced with red bars for observations and lavander ones for the model. The arguments required are:

- **obs**: R-object for the observations
- **sim**: R-object for the controlled period
- **rr**: if true, the study is on precipitations. If false, the study is on temperatures.

Example of call:
> *compare.plot(st.obs,st.ctr,rr=F)*

4.4.8 plot.means

This routine realizes several plots of the observed and controlled mean values for each season and for each month all over the period.
The arguments (**obs**, **sim** and **rr**) have the same significance than for the previous routine.
Example of call:
> *plot.means(st.obs,st.ctr,rr=F)*

4.4.9 plot.freq

This routine plots the empirical cumulative distribution function (ecdf) for each season during the whole period of the study.
It has only two arguments which are:

- **data**: the R-object representing the dataset;
- **rr**:if true, the study is on precipitations. If false, the study is on temperatures.

Example of call:
> *plot.freq(st.obs,rr=F)*

4.4.10 percentile

This function calculates percentiles for a particular level. Its different arguments are:

- **data**: the R-objet representing the dataset;
- **percent**: quantile to be studied;
- **rr**: if true, the study is on precipitations. If false, the study is on temperatures.

Example of call:
> *percentile(st.obs,0.95,T)*

4.4.11 plot.percentile

This routine plots on the same graph the cumulative distribution function of the observed, modelled and scenario periods.
Example of call:
> *plot.percentile(st.obs,st.ctr,st.sc)*

4.5 extreme_value.R

This code uses GEV Distribution to study extreme values. It contains the following routines:

4.5.1 select.max.year

This function returns a list of the annual maximum for a dataset given in the arguments. This is computed for precipitation if the argument **rr** is true, for temperature in the alternative case.
Example of call:
> *select.max.year(st.obs)*

4.5.2 month.season.max

This routine returns a list of the seasonal and monthly precipitation (rr=T) or temperature (rr=F) maxima for each year of the period.
The name of the R-object for the dataset is specified in the arguemnts.
Example of call:
> *month.season.max(st.obs, T)*

4.5.3 temperature.min

Equivalent to the previous one but for minimum temperature, this routine can be called by the command:
Example of call:
> *temperature.min(st.obs)*

4.5.4 plot.gev

This function produces four diagnostic plots: probability plot, quantile plot, density plot and the return level plot.
Its arguments are:

- **data**: R-object for the dataset;
- **rr**: if true, the study is on precipitation. If false, the study is on temperature.

Example of call:
> *plot.gev(st.obs,rr=F)*

4.5.5 gev.return.period

GEV distribution is here used to calculate the return level values for 5, 10, 25 and 50 year return periods. This is realized for a vector a maxima or minima specified as an argument.
Example of call:
> *fgev(vect.max)*
where vect.max is a vector containing annual minimum for instance.

4.5.6 try

This function has been written with the aim at drawing a comparison between GEV Distribution using Coles estimators and GEV Distribution using **fgev** procedure.
The comparison is particularly for datasets containing yearly maximum values, as it the case for **oxford** data for instance.
Example of call:
> *data(oxford)*
> *plot.gev(st.obs,rr=F)*

4.6 estimator.R

This code aims at estimating the return levels and return intervals assuming a General Pareto Distribution (GPD).
It consists of twelve functions as detailled below.

4.6.1 mean.excess

Here is determined the Sample Mean Excess Function used to estimate GPD parameters. This is then a support routine for **gpd.mean.excess.**
Its arguments are:

- **u**: threshold for the dataset
- **x**: R-object for the dataset

4.6.2 gpd.mean.excess

This routine estimate the GPD parameters using the previous function and is used as a support routine for **gpd.fit.iter**. These arguments are:

- **u**: threshold for the dataset
- **x**: dataset
- **lplot**: if true (default case), the mean exceedance function is plotted against threshold.

Points of the cumulative distribution as well as ξ and σ parameters, the number of exceedances (**nexc**) are stored in a list.

4.6.3 gpd.moment

The method of moment is another way of estimating GPD parameters and this is what is realized in this function used as a support routine for **gpd.fit.iter**.
Its arguments are the threshold u and the R-object representing the data.

4.6.4 gpd

From ξ and σ (GPD parameters) given in arguments, this routine returns the value of the GPD distribution in a specified point. It is also used as a support routine for **gpd.fit.iter**.

4.6.5 gpd.fit.iter

This function, which is a support routine for gpd.fit.iter, estimates the GPD parameters using the moment estimator as a first guess. The different arguments are:

- **u**: threshold
- **x**: dataset

- **sig.lim**: top limit for the value of σ (25 by default)
- **xi.lim**: top limit for the value of ξ (3 by default)
- **lmoment**: if false (default case), parameters are determined using the mean excess function. Otherwise, this is realized using the method of moment.
- **accuracy**: 0.0001 by default, intervals between the steps of calculs.
- **N.steps**: number of steps (=500 by default).
- **f.guess**: first guess for GPD estimators (NULL by default).

The routine returns a list containing the density and the cumulative distribution, σ and ξ, the number of exeedances and the error between the first guess and the stabilized function.

4.6.6 gpd.fit

This routine calls gpd.fit.iter iteratively to estimate more accurately GPD parameters.
Example of call:
> gpd.fit(10,st.obs)

4.6.7 plot.gpd

This routine plots the GPD Distribution using parameters from mean excess function. It also permits to add the cumulative distribution function from GEV to the plot and to compute the return level plot. For its arguments, it only requires an evd object, determined from **gpd.fit.iter**.
Example of call:
> gpd.obs <- gpd.obs(10,obs)
> plot.gpd(gpd.obs)

4.6.8 add.coles

This routine has been implemented to compare GPD Coles results with results from **gpd.fit.iter**, using **rain.data** dataset as a verification.
The GPD distribution using Coles GPD parameters is then added to the plot provided with the previous R-procedure.
The dataset is given in argument.
Example of call:
> add.coles(rain.data)

4.6.9 pp.gpd

This routine produces the probability plot for GPD distribution. It is only required to specify the evd object in the argument.
Example of call:
> pp.gpd(gpd.obs)
where gpd.obs stands for the object returned by **gpd.fit.iter**.

4.6.10 rl.gpd

Here are given the return levels for GPD distribution. Example of call:
> rl(gpd.obs)
where gpd.obs is the GPD object returned by gpd.fit.

4.6.11 extreme.quantiles

This routine determines the return level values for different return periods. Example of call
>extreme.quantiles (gpd.obs)
where, once more, gpd.obs is the evd object (from gpd.fit.iter).

4.6.12 plot.u

This routine plots GPD parameters against threshold.
It has only one argument, which is the R-object containing the dataset.
Example of call:
> plot.u(st.obs$rr)

4.6.13 cmp.distr

This function compares two different distributions by plotting their histograms. It is used to appreciate how well GEV fits to GPD. These arguments are the two vectors for the distributions. For instance, if GEV and GPD distributions are to be compared, the corresponding call will be:
Example of call:
> cmp.distr(obs.nexc, vect.max)
where obs.nexc is the vector containing all the values above the threshold chosen to apply GPD model and vect.max is the vector containing the annual maxima (for GEV model).

4.7 select_analysis.R

This code aims at computing the whole statistical analysis iteratively for all the stations. It includes four routines presented here.

4.7.1 analysis

The routine **write.results** provides all the statistical tests presented previously for one specific station.
Its arguments are:

- **rr**: True (default case) if results on precipitations are neeeded, false otherwise
- **conf**: confidence level for the student or wilcoxon test used in the comparison test of the means and of variances (0.95 by default)
- **i.data**: number of the station to study
- **mean.test**: permits to select the kind of analysis wanted. If true, which is the default case, the first type of statistical tests is performed. In the other alternative, the study is on extreme values.

Example of call:
For instance, the command:
> analysis(T,0.95,12680,T)
will perform the mean, variance and percentiles tests for precipitation in Oslo (station=12680), using a 95% level of confidence in Wilcoxon test.

4.7.2 write.results

This calls iteratively the previous routine for all the stations in order to write the results in different tables.
This function has several parameters which all have a default value and which already have been detailed for the routine **analysis**.
Example of call:

Thus, the command
> *write.results()* will write in a table all the results relative to the first type of tests, for a confidence level equal to 0.95 and on precipitations data.

4.7.3 look.for.files

This routine permits to determine wether or not observations, model and scenario files can be found in the corresponding directories for the same station. It is used as a support routine for **analysis** in order to make comparisons between the three files.
None argument needs to be specified.

4.7.4 read.files

This routine imports into R the files corresponding to the observed, controlled and forecast periods for the same station.
It takes as an argument the number which identifies one station (**stat.nbr**).

Chapter 5

General statistical tests

Climate can be defined as an ensemble of many weather phenomena. Several statistical techniques can be used to describe climate, such as the distribution, the mean, the variance or the percentiles. Such an analysis was performed for each station, as presented in the chart of tasks 5.1.

5.1 Distribution for Temperature

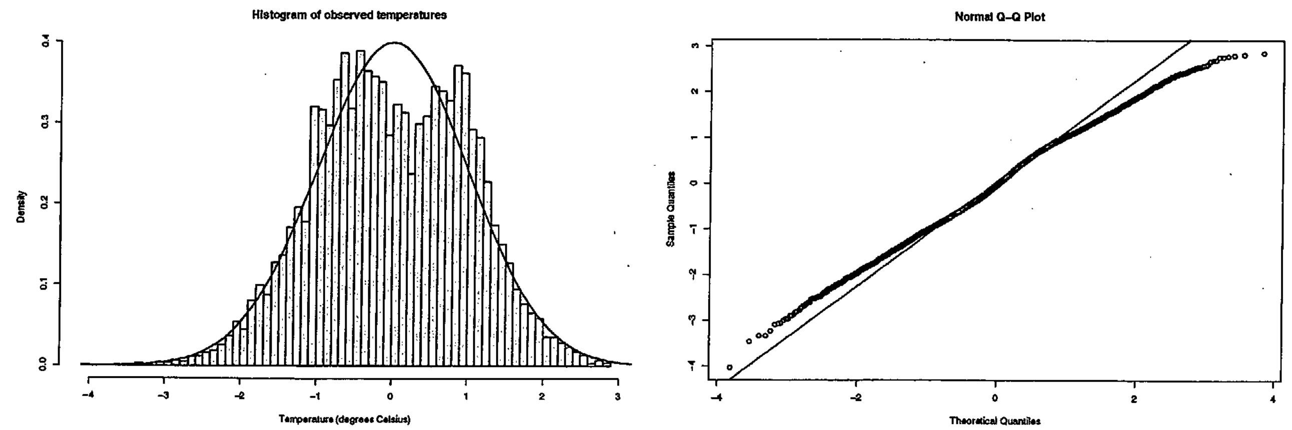


Figure 5.1: Comparison between normal and temperature densities (*left*) and quantile plot (*right*) for Bergen station (1980-1999).

Generally, temperature is considered as approximately normally distributed and this is what can be checked graphically. The first of the two plots presented in figure 5.1 compares the density of a normal distribution (mean=0, standard deviation=1) with the histogram of the temperature

	Month	Season	Annual
Mean value, standard deviation	T/P	T/P	T/P
Percentiles	T(jan, apr, jul, oct)	T/P	T/P
Extreme values	T(jan, apr, jul, oct)	T/P	T/P

Table 5.1: Summary of the different statistical tests realized.
Legend: ***T****=Temperature,* ***P****=Precipitation.*

observed in Bergen.

The second plot is a normal quantile-quantile plot of the observations. If the data are normally distributed, they should fit to the line which passes through the first and third quartiles and which appears in red on the figure. It can be observed some departures in the upper tail, suggesting that the normal hypothesis is approximative, especially in the extremes.
This will be a motivation to apply extreme value theory.

5.2 Mean values and variances

The mean (conventionally the monthly and annual mean) of weather-related arguments is often useful to describe climate and, consequently, annual and seasonal means were determined both for precipitation and temperature.

5.2.1 Precipitation

The purpose of this project being to describe climate change, a comparative study in mean values between the present period and the scenario will be presented here rather than a description of the seasonal differences in precipitation amounts.

Comparison between the observations and the model

The mean is an indicator which can be used to compare two different time series. Modelled one-day precipitation was then compared to observed precipitation at all the sites in order to appreciate the quality of the model.
For most of the sites, model results have higher mean values than the observations, especially in summer. During Autumn and Winter, it seems that precipitation is sligthly underestimated in the northern regions.

To have a more precise idea of the significance of these differences, statistical tests can be used. In this regard, **Wicolxon-test** appears interesting as it permits to say in case of no-normality pattern, whether or not the difference in means is significant, the data being independant or not. The null hypothesis here considered is then:

$$H0: \quad \mu = 0$$

where μ is the difference in the two means considered.

The test returns the p-value. If it is smaller than 0.05 that means that $H0$ must be rejected for a 95%-confidence level, and that the difference in means is significant.
When considering tables 8, 9, 10, 11 and 12 (in Appendix), it appears that the difference is significant for most of the stations on an annual basis, when considering a confidence level of 0.95 for the test.
However, the comparison leads to very different conclusions according to the season taking into account. Thus, if in Winter the model presents quite a good fit to observations, differences are more relevant during Summer, especially in the Oslo area which is the most likely to be overestimated.

Comparison between the model and the scenario

The comparison between the model and the scenario reveals some changes in precipitation amounts and once more, the importance of the difference is dependant on the season considered. On a general aspect, the difference in mean values is mostly siginificative as the annual comparison suggests it. These changes are stronger during Summer and Autumn than in Winter and Spring for which values are very close to the control period means.

5.2.2 Temperature

The same kind of analysis can be computed for temperatures. However, the normal distribution hypothesis is now valid. This means that **Student-test** (t-test) (detailed above) as well as **Fisher-test** (cf. above) can both be applied in this case.
This test assesses whether or not the means of two groups are statistically different from each other. This analysis is appropriate supposing that the data are randomly drawn from a population and that their distribution is normal.

The hypotheses for a single sample t-test are:

- H0: $u = u0$ (null hypothesis)

- H1: $u \neq u0$ (alternative)

(where $u0$ and u denote the two means being compared)

Results of the student-test

If the p-value associated with the **t-test** is small (usually set at $p < 0.05$), there is evidence to reject the null hypothesis in favor of the alternative. In other words, there is evidence that the mean is significantly different from the hypothesized value.
If the p-value associated with the t-test is not small ($p > 0.05$), there is not enough evidence to reject the null hypothesis, and it must be concluded that the mean is not different from the hypothesized value.
P-values do not simply provide with a Yes or No answer, they also provide a sense of the strength of the evidence against the null hypothesis. The lower the p-value, the stronger the evidence.

Comparison between the observations and the model

The annual as well as the seasonal means for temperature are depicted in tables 13, 14, 15 , 16 and 17. Their values differ a lot from area to the other, but what is to notice is that the model shows a very good agreement with the observations.
However, means is not the only element to consider, variance also needs to be taken into account when comparing the model with the observations. Indeed, to determine if the two means are different, it is also important to study how well the model reproduces variability.
This can be determined by performing the **Fisher test** which compares variances of two samples from normal populations.
The null hypothesis is that the variances for the two periods are the same, the alternative is that their are different.
As it is suggested in tables 13, 14, 15 , 16 and 17, it indeed appears that the model very often shows lower variability than the observations. This is an important element which needs to be considered for further studies, especially when return level values will be focused on in the next chapter. This also leads to compare the scenario period with the control period rather than with the observations as they were produced by the same model.

Comparison between the model and the scenario

For all the stations, the 2030-2049 increase in mean temperature is statistically significant, as proved the p-values of the t-test returned. This reflects an increase in temperature, which appears to be true for all seasons.
As for variances, they prove to be quite close for the two periods, which is what was expected as the model to generate the two series is the same.

5.3 Percentiles

Trends in percentiles are another element to describe climate evolution. In that purpose, they have been determined both for precipitation and temperature by using the Cumulative Distribution Distribution Function (e.c.d.f).
E.c.d.f is a step function F_n with jump $1/n$ at each observation (possibly with multiple jumps at one place if there are ties).

5.3.1 Precipitation

It is interesting to calculate high percentiles for precipitation such as the 95, 99 or 99.1th percentiles, precipitation above the 95th percentile being defined as very wet days. This is what have been made and sum up in tables 19, 20, 21, 22.

The results show that the values of the percentiles are higher for the scenario period than for the control period at most stations. This reflects a positive trend in extreme precipitation events in the next years.

5.3.2 Temperature

Cold days and warm days are defined respectively by the 10th and the 90th percentiles. Looking at tables 24, 25, 26, 27, it appears that both the trend in cold days and the trend in warm days indicate warming. The results generally show indeed a warming in extremes as daily mean temperatures are higher both for warm and cold days at most stations.

Studying mean or percentile values gives a global view on climate aspects. However, they are not all the climate. Climatic changes migth occur if certain aspects of the distribution of extreme values change, while the mean does not. That is why a description of precipitation and temperature using the extreme value theory has then been led. This is the purpose of the next chapter.

Chapter 6

Results on return levels

The object of study are collections of annual and seasonal maxima for precipitation and of annual, seasonal as well as monthly minima and maxima for temperatures that are observed daily.

Extreme value theory can be applied equally well to model the lower tail of a distribution as the upper tail. Then, working with *minimum* temperatures is equivalent to working with *maxima* by negating the series and modelling the upper tail. This is how it has been proceeded during this project.

6.1 Comparison between Coles and R-functions

Splus-programs of Coles (1999) (ref. [1]) permit to estimate GEV and Pareto parameters mainly based on the maximum likelihood estimation.
During this project, it has been chosen to implement the same kind of functions using R-package. However, the procedures implemented were not exactly the same and it can appear interesting to compare the results.

6.1.1 GEV Distribution

The dataset used for the comparison of the GEV parameters can be made on the time series of annual maximum temperatures at Oxford, England, from 1901 to 1980 and stored in **oxford.data**.

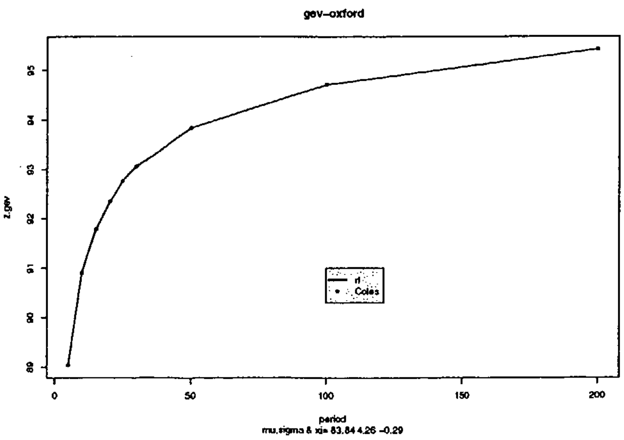


Figure 6.1: Coles and fgev results compared for oxford dataset

Splus function **gev.fit** (Coles) and **fgev** lead to the same estimators for μ, σ and ξ, which is reflected by the exact correspondance of the return level plots obtained respectively for Coles and

	Return Levels					GPD parameters	
	5 years	10 years	25 years	50 years	100 years	ξ	σ
gpd.fit.iter values	44.94	51.91	63.53	77.46	93.73	0.3095	6.0490
Coles values	48.85	56.25	67.61	77.58	88.9	0.1844	7.4402

Table 6.1: Coles and gpd.fit.iter results compared for rain.data dataset

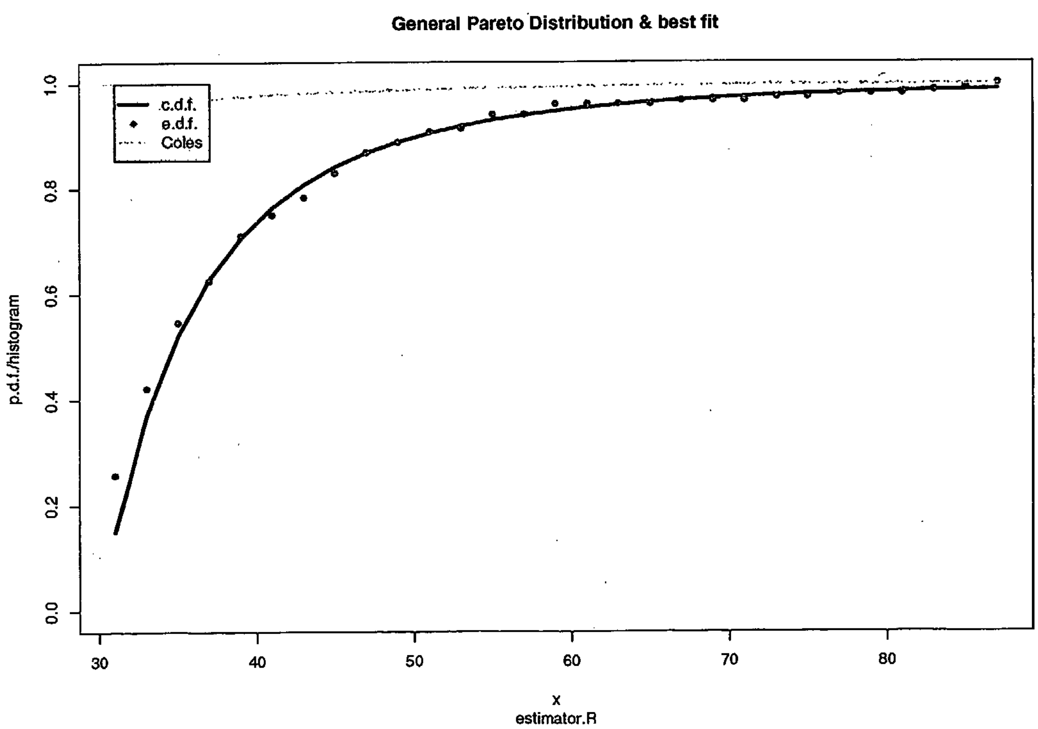
the R-function **rl** (figure 6.1).
This assesses the validity of the R-function.

6.1.2 Pareto Distribution

The comparison for Pareto study was based on **rain.data** dataset which is a vector of length 17531 and which contains daily rainfall accumulations in mm at a location in soutwest England.

Values of the parameters determined with the implemented R-routine gpd.fit.iter on one hand, and Coles parameters on the other hand are shown in table 6.1. The results suggest that the shape parameter ξ is positive and so regular in both cases. The scale parameters σ are quite similar.

The return levels obtained for the two methods are quite close and give a first evidence of the quality of the R model. A more suggestive illustration is given by the comparative graph 6.2. This graph suggests indeed that the Pareto distribution calculated from **gpd.fit.iter** parameters fits better than Coles the cumulative Pareto distribution. As a consequence, it seems justified to use the R-function **gpd.fit.iter** for further studies on GPD.



6.2 Comparison between GEV and GPD results

It can be interesting to draw a parallel between GEV and GPD methods, especially as far as return period values are concerned.

6.2.1 Results for Precipitation

Results with GEV

Table 28 gives the maximum precipitation return levels for 5, 10, 25 and 50 years return periods.

Conclusion

There is evidence for the most extreme levels of rainfall to occur more during the scenario period but this cannot be generalized to all stations and this is more relevant for short return periods than for long return periods. However, these results need to be taken with care as a larger dataset would be needed for a better conclusion.

Results with GPD

In methods like GPD, the main difficulty comes from estimating a suitable threshold.
Indeed, if the threshold value is too high, the inference will not be based on a large enough number of exceedances. On the other side, thresholds need to be high enough to only take into account extremes and to insure the validity of the asymptotic argument.

To help in making a choice, Coles proposes a method which is rather a method of threshold verification based on the likelihood analysis itself.
Maximum likelihood estimators are plotted against thresholds. Some stability should then be observed in the parameter estimates at a threshold where the asymptotics are valid. This is what Coles realizes with the SPlus function **.fitrange** which is computationally intensive and consequently very slow.
A similar function has been realized with R-package during this project. It is called **plot.u** and is part of **estimator.R** code. However, caution must again be taken in the selection on one particular threshold.

Thresholds can also be defined for each station relative to local climate. Thresholds can be set to a certain percentile of the distribution. Thus, considering the **90th percentile** of daily observed data permits to take a threshold adapted to each station and so this is what is realized here.
In that way, extreme events are defined by determining threshold values according to the data's own distribution.

The return levels then obtained and depicted in Appendix 29 have much lower values than for the GEV method. They also show an increase in rainfall amounts for all the return periods tested but the differences are not so significant anymore.
As above, these gaps in values may be accounted by the fact that the GEV method is not applied on a lare enough number of data.

6.2.2 Results for Temperature

Results with GEV

To apply the GEV method, the maximum and the absolute values of absolute minimum temperature in degrees Celsius were considered. This was again realized for the three different periods (observations, model and scenario).
Tables 30 and 31 present the return levels using GEV method for the observations, the model and the scenario periods.
The two first columns are interesting in the sense that they permit to appreciate the quality of the

model. It then appears quite clear that there is a systematic and significant difference between the two.
This suggests that it is better to compare the scenario period with the control period than with the observations as both the scenario and control data come from the same model. Doing this, differences do not appear any longer so important, even if they still exist.
Annual minimum daily mean temperature analysis reflects that there is a certain increase in return level values, which is a bit more obvious when considering short time return periods like 5 years.
Annual maximum daily mean temperature also experiment an increase in return levels at most stations but this is not always statistically significant.

Results with GPD

Return levels using GPD method are a bit different from results obtained with GEV in the sense that, for instance, the return level values for minimum daily mean temperature are generally lower with the Pareto distribution than with GEV.
It can also be noticed that the positive trend in maximum daily mean temperature does not appear as significant as with the first method tested. Thus, only 12 stations out of 37, that-is-to-say 32% of the stations, show an increase in temperature for a 50-year return period whereas there were 64% of them with GEV.

Chapter 7

Study of an example

Here is illustrated the use and the results of the different routines on one particular station. As an example, the statistical tests previously presented are applied to the station number 50540 which stands for Bergen-Florida. Precipitations are also chosen as a support of study.
Bergen is located at 60.23 degrees northern latitude and 5.20 degrees eastern longitude at the coast of Norway. This part of Norway is strongly influenced by maritime airmasses from the Atlantic Ocean.

7.1 General analysis

7.1.1 Mean values and variances

The dataset corresponding to Bergen gives observed precipitation from 1980 to 1999, which results in 7306 daily obervations.

It is first necessary to load the files containing the observations as well as the data for the control and the scenario periods.
In this view, it is needed to call the routine read.files and to specify the number of the station:

```
> read.st <- read.files(50540)
```

And then to extract the data:

```
> st.obs <- read.st$obs
> st.ctr <- read.st$ctr
> st.sc <- read.st$sc
```

where st.obs, st.ctr, st.sc are the R-objects respectively for the observations, the control period and the scenario period.
A first approach of Bergen station can be to produce result summaries of precipitation data for the three periods.

```
> summary(st.obs$rr)
Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
0.000 0.000 1.500 6.589 9.300 104.400 1097.000
```

```
> summary(st.ctr$rr)
Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
0.000 0.400 3.200 7.294 10.700 116.600 1.000
```

```
> summary(st.sc$rr)
Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
0.000 0.600 4.100 8.879 13.000 131.300 1.000
```

These first outputs yet reveal some differences between the three periods, the median and the maximum temperature being more important with the model than with the observations and the scenario period presenting even higher values for these indicators.
Another illustration of this is given in Figure 7.1 where observed annual mean values (*in red*) are compared to the model (*in lavander*) .

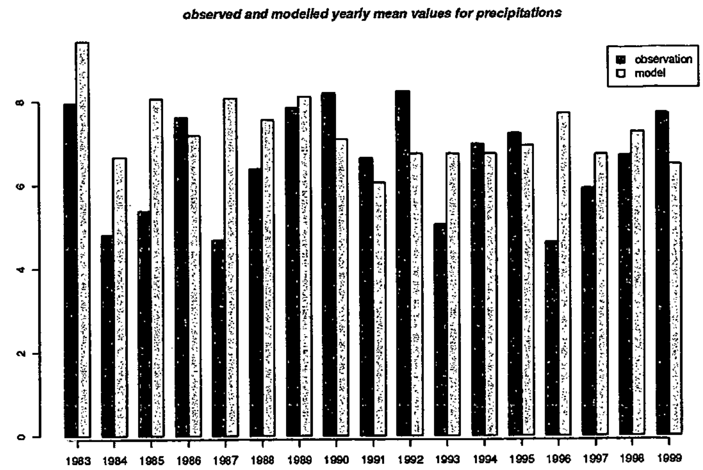


Figure 7.1: Comparison between observed and modelled annual mean values.

Annual as well as seasonal and monthly means can be obtained by the command:

```
> mean.study(st.obs,rr=T)

 $variable
[1] "precipitations"

 $annual
[1] 6.59

 $spring
[1] 4.94

 $summer
[1] 4.98

 $autumn
[1] 8.15

 $winter
[1] 8.34

 $month
m.month m.month m.month m.month m.month m.month m.month m.month m.month
[1,] 8.12 7.33 7.17 4.43 3.2 3.82 4.97 6.12 7.92
m.month m.month m.month
[1,] 9.03 7.45 9.49
```

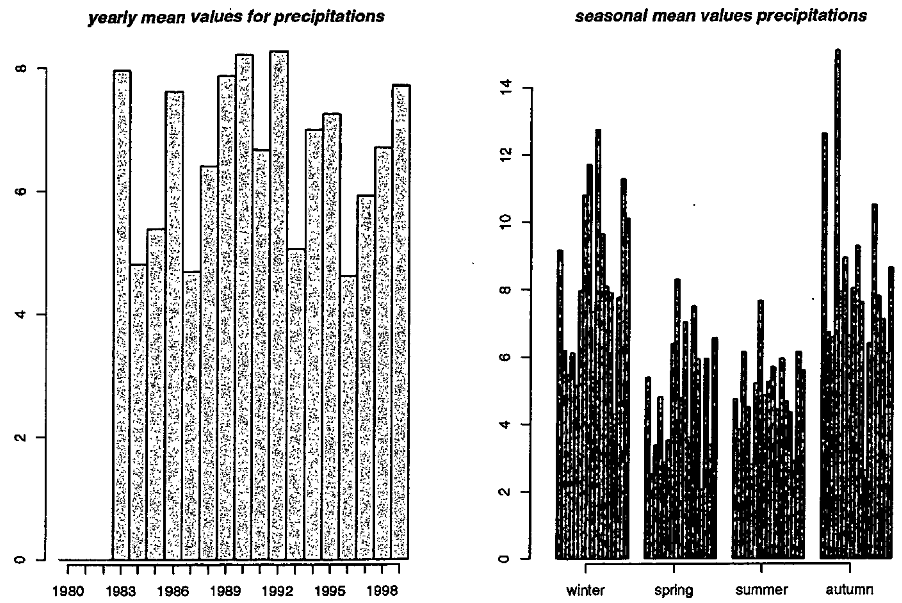


Figure 7.2: Precipitation annual mean values over the observation period (*left*) and precipitation mean value for each season over the same period (*right*)

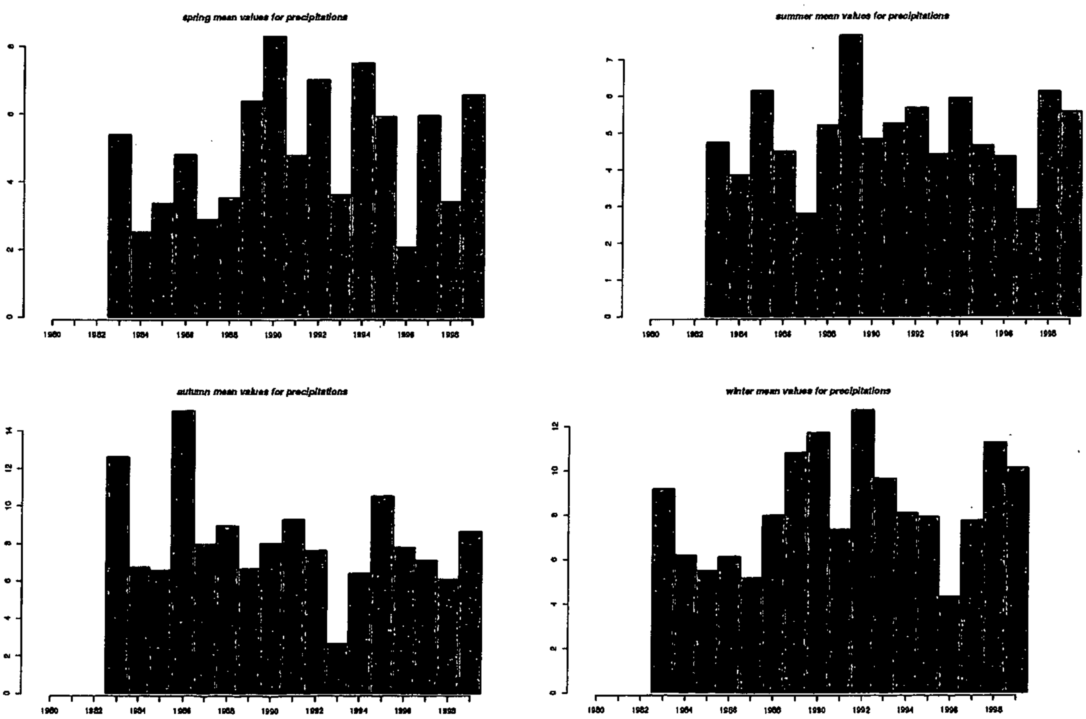


Figure 7.3: Seasonal mean values in Bergen *(upper left: Spring, upper right: Summer, left bottom: Autumn, right bottom: Winter)*

Figures 7.2 and 7.3 illustrate the evolution in precipitation over the period 1980-1999.
The same analysis as previously can be led with both the control period and the scenario period. The results are gathered in tables 7.1 and 7.2.
Then, a Wilcoxon test can be computed in order to appreciate the difference in mean values for the three periods. This is made possible by calling the command:

```
> wilcoxon.comparison(st.obs,st.ctr,rr=T,0.95,var=F)
```

which gives the output:

```
$confidence.level
[1] 0.95
$annual
[1] 0.1675
$summer
[1] 3e-04
$autumn
[1] 0.1582
$winter
[1] 0.4239
$spring
[1] 0.2565

$significant.diff
diff.year diff.summer diff.autumn diff.winter diff.spring
[1,] FALSE TRUE FALSE FALSE FALSE
```

$confidence.level is the level of confidence of the test. It is given in argument and here its value is 0.95.

$significant.diff is a boolean indicating if the difference in means is significant for the test (True) or not (False).

Conclusion
Bergen appears to be a station which experiments a lot of rain irrespective of the time of the year with a peak during Winter and Autumn. The model reveals to overestimate precipitation but this difference is only statistically significant in Summer, for a 95% level of confidence. For this period indeed, the Wilcoxon test returns a p.value lower than 0.05.
As for Winter, it is the only season for which observations are underestimated, but the difference is not significant for the Student test.

According to the scenario period, precipitation in Bergen should increase all over the year and this is significant for Spring, Summer and Winter.

To assess the validity of the model, it has been shown (section 5.2) that it was important to look at variances.
This can be realized by writting:

```
> var.study(st.obs,rr=T)
```

which leads to

```
$variable
[1] "precipitations"
$annual
[1] 112.96
```

	observations	control period	scenario period
means	6.59	7.29	8.88
variance	112.96	101.85	142.55

Table 7.1: Precipitation annual mean values and variances in Bergen (station=50540) for observations, control and scenario periods.

	Spring			**Summer**			**Autumn**			**Winter**		
	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
means	4.9	5.8	6.3	4.9	6.8	8.7	8.1	9.0	11.7	8.3	7.4	8.6
variance	77.2	82.6	81.2	78.2	72.4	115.0	152.9	143.6	203.0	133.6	103.7	156.3

Table 7.2: Precipitation seasonal mean values and variances in Bergen (station=50540) for observations, control and scenario periods.

```
$spring
[1] 77.27
$summer
[1] 78.28
$autumn
[1] 152.98
$winter
[1] 133.61
$month
v.month v.month v.month v.month v.month v.month v.month v.month v.month
[1,] 123.06 113.81 120.99 63.68 38.73 60.27 68.33 103.32 148.71
v.month v.month v.month
[1,] 182.73 125.76 160.33
```

This output has the same structure as the output from mean.study.
The variance study here is on observations but similar tests can be run for the other periods, as it is made in tables 7.1 and 7.2.

Conclusion

The comparison between variances for the observations and the control period suggests that the variability of the model is lower than for the observations. On the contrary, the scenario period introduces a high variability, even in Winter.
The conclusions related to further analysis will have to take into account these results in order not to interpretate too quickly possible gaps between values.

7.1.2 Percentiles

Climate changes can also be described by the evolution of percentile values. Such a comparison can be illustrated in a graph where the cumulative functions for the three periods are plotted. Thus, figure 7.4 shows that the control and the scenario periods are quite close for small percentiles but differ more from the 80th percentile. In extremes, precipitation amounts are larger for the scenario, which would suggest an increase in rainfall during the 21st century.

Numerical values can also be obtained by calling the routine percentile as below:

```
> percentile(st.obs,0.95,rr=T)
$annual 95%
28.06
$spring 95%
```

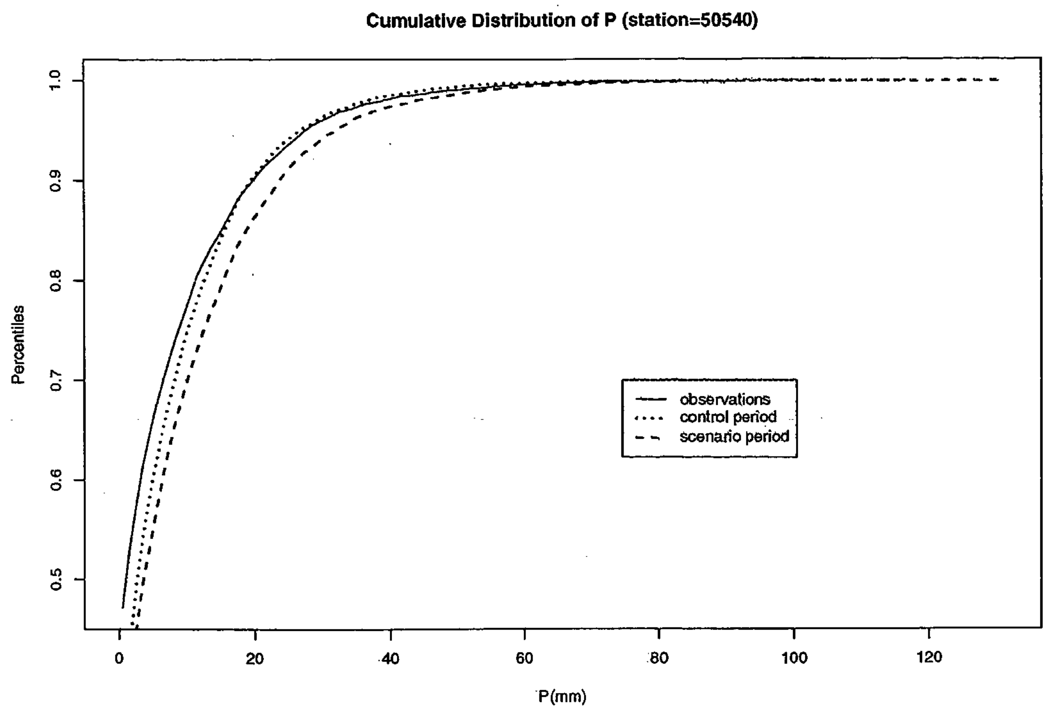


Figure 7.4: Comparison of the Cumulative functions for observations, control and scenario periods for Bergen station (annual values)

	95 percent.			99 percent.			99.1 percent.		
	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
annual	28.06	27.4	32.31	48.76	45.10	54.30	50.41	45.94	55.32
Spring	21.78	22.90	25.50	41.67	40.91	37.1	42.48	42.87	37.74
Summer	24.00	23.50	30	39.69	36.80	45.1	40.66	37.08	45.34
Autumn	32.70	32.81	40.90	56.77	49.30	63.10	57.91	50.47	64.27
Winter	30.71	51.97	32.33	27.81	45.60	55.00	52.96	46.78	57.19

Table 7.3: Annual and seasonal 95, 99 and 99.1 percentiles for precipitation in Bergen (station=50540)

21.785
$summer 95%
24
$autumn 95%
32.7
$winter 95%
30.71

Results for different percentiles and for the three periods are depicted in table 7.3. This table makes it clear that percentiles are higher for the scenario period than for the 1980-1999 period.

Conclusion

The determination of high percentiles such as the 95th, 99th or 99.1th percentiles goes along with the mean study. Once again indeed, it appears that precipitation amounts are quite important in Bergen, the two wettest seasons being Winter and Autumn, and that forecasts for the scenario period lead to even higher values of these percentiles. An increase in heavy daily rainfalls can then be expected in the future.

	μ			sigma			ξ		
	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
Estimators	63.5934	63.15672	71.593	16.585	16.097	15.321	-0.275	-0.029	0.0349
Standard Errors	4.466	4.146	3.878	3.160	3.040	2.847	0.166	0.194	0.174

Table 7.4: GEV estimators and standards errors for maximum precipitation in Bergen (station=50540)

7.2 GEV results

7.2.1 Estimation of the parameters

The function to estimate the GEV distribution is **fgev**. It is necessary to give only the argument corresponding to the vector containing the maximum data.
For example, if the annual maxima for the period 1980-1999 are in yea.obs$annual.year, the two commands:

```
> yea.obs <- select.max.year(st.obs,rr=T)
> fgev(yea.obs$annual.year)
```

permit to get the output:

```
Call: fgev(x = yea.obs$annual.year)
Deviance: 143.8466

Estimates
loc scale shape
63.5934 16.5851 -0.2753

Standard Errors
loc scale shape
4.4665 3.1601 0.1667

Optimization Information
Convergence: successful
Function Evaluations: 28
Gradient Evaluations: 17
```

Estimates contains the maximum likelihood estimates of respectively the location parameter, the scale parameter and the shape parameter.

Standard errors give the standard errors of the maximum likelihood parameters, again for the three parameters.

Deviance gives the deviance at the maximum likelihood estimates.
The estimated value of the shape parameter ξ is higher than -0.5, which corresponds to the case of a regular estimator (chapter.1). This is reflected by a concave extrapolation in the return level plot (Figure 7.7). Thus, the model has a finite upper end-point. Here, it must be noticed that even if the standard error for ξ is taken into account, the shape parameter remains greater than -0.5, which gives even more evidence to the concave curve.

Estimators for the different periods are depicted in table 7.4. Values for the observations and the control period are quite close except for the shape parameter which is lower for the observations than for the two other periods. This is reflected by a more concave extrapolation in figure 7.5 which shows the return level plots for the observations (*black line*), the adjusted values (*in red*) and the scenario (*in blue*). The points are the the maximum precipitation for each year and the curves are the return level plot lines.

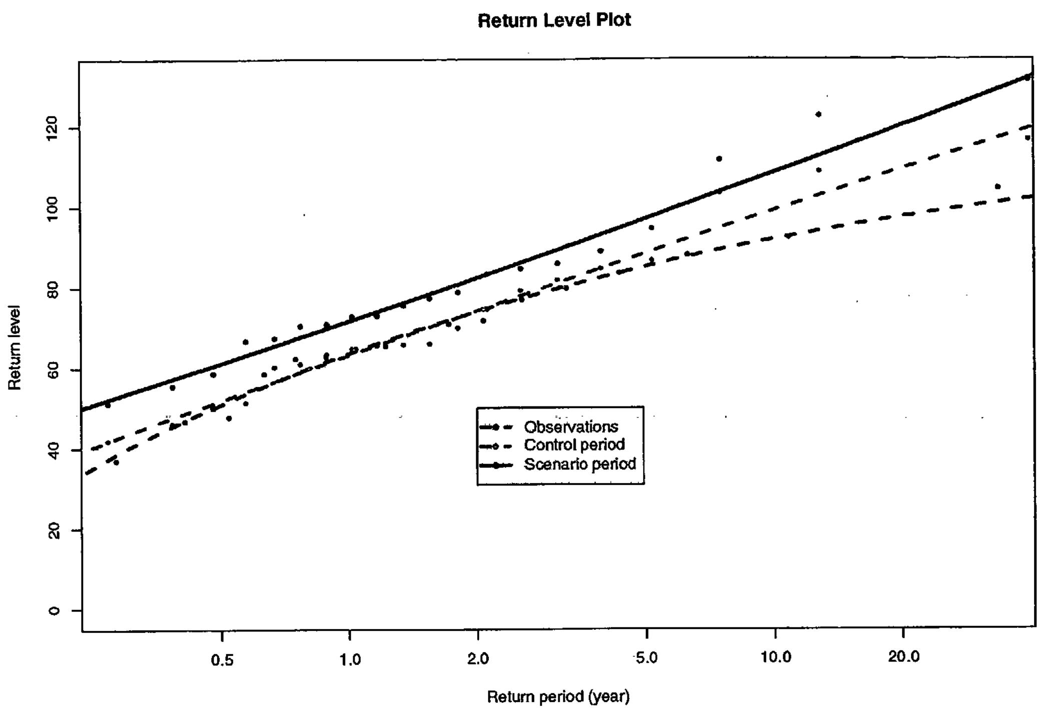


Figure 7.5: Comparative Return level plots using GEV distribution for precipitation observations, control period and scenario in Bergen

7.2.2 Return levels

To illustrate the results and to assess the quality of the fitted model, the function **fgev** also provides four graphical diagnostics.

Using for instance,
> M1 <- fgev(yea.obs$annual.year)
to store the application of **fgev** in the object 'M1'.

Then, the command:
> plot(M1)
permits to get a **P-P plot**, a **Q-Q plot**, a **density plot** and a **return level plot** (Figure 7.6).

Both the probability plot and the quantile plot suggest that the quality of the GEV is quite good. The return level plot includes 95% confidence intervals and shows how the fitted model extrapolates from the sample information. Again, this extrapolation suggests that the fit is quite reasonable.
The last figure compares the fitted GEV density function with the data.

If the study is more on return periods, as it is the case in this section, the return level plot can also be visualized directly by:
> rl(M1)

On the graph, the curve within the return level plot is z_t plotted against y_t on a logarithmic scale, using maximum likelihood estimates of μ, σ and ξ .

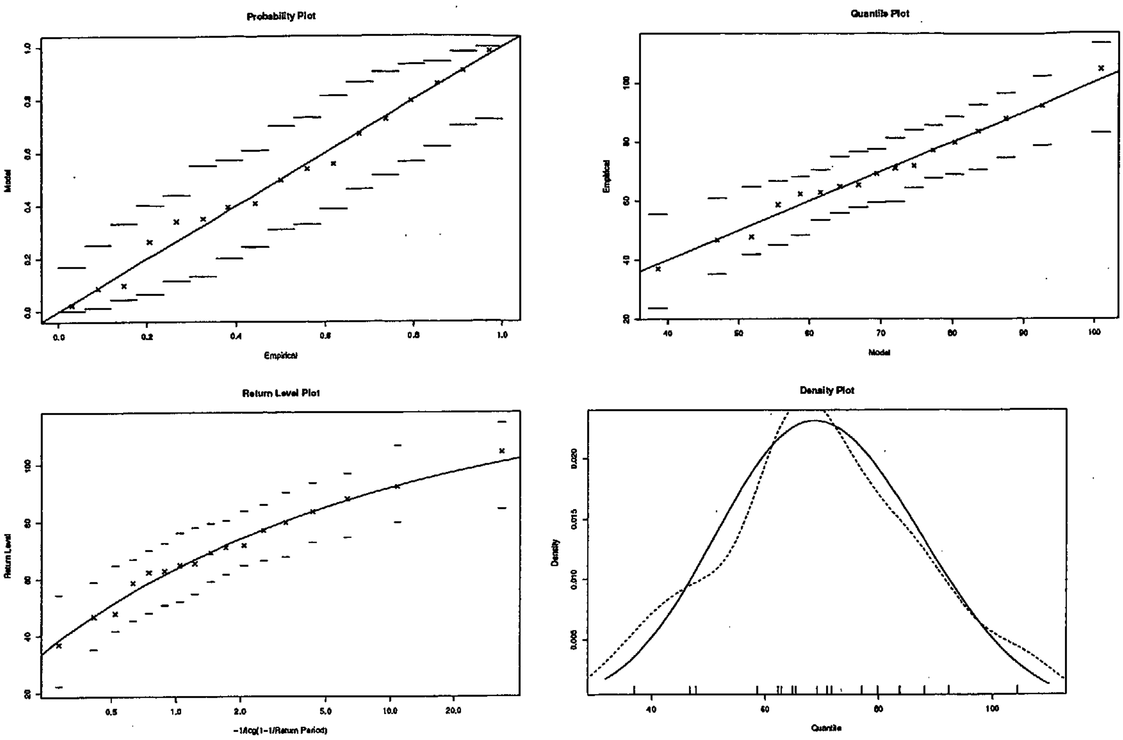


Figure 7.6: Diagnostic plots for GEV fit to precipitation observations in Bergen

The points on the plot are

$$(-1/log(p_i), z_i), \qquad i = 1, ..., m$$

where $p_1, ..., p_m$ are plotting ppoints defined by points, and $z_1, ..., z_m$ are the data used in the fitted model, sorted into ascending order. For a good fit the points should lie "close" to the curve defined by $(z_t, log(y_t))$, which is the case in Figure 7.7.
On this graph, the horizontal lines (resp.red, blue and green) corresponding to returns periods (resp. 5, 10 and 25 years) have been drawn using the results of the function **gev.return.period**. The knots being on the curve, this graph permits to assess the validity of the return level values obtained.

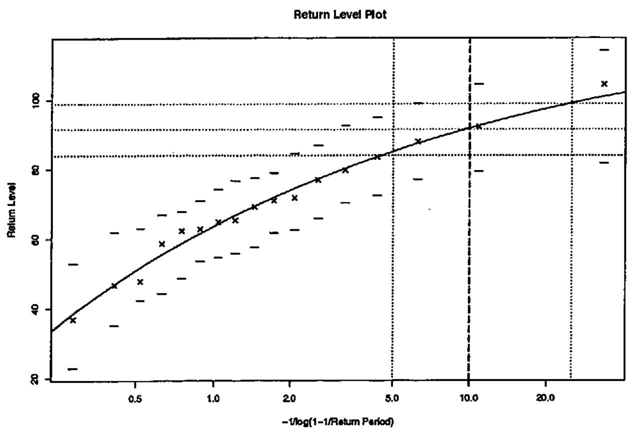


Figure 7.7: Return level plot using GEV distribution for precipitation observations in Bergen

5 years			10 years			25 years			50 years		
obs	ctr	sc	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
83.97	86.78	95.19	91.41	98.22	107.47	98.86	112.33	123.45	103.26	122.54	135.65

Table 7.5: Maximum precipitation return levels (in mm) using GEV distribution for Bergen station.

5, 10, 25 and 50 years return levels can also be obtained, directly by the commands:

```
> yea.obs <- select.max.year(st.obs,rr=T)
> gev.return.period(yea.obs$annual.year)
$r.val5
[1] 83.97
$r.val10
[1] 91.41
$r.val25
[1] 98.86
$r.val50
[1] 103.26
```

Conclusion

What first can be evidenced from the comparison of the return levels shown in table 7.5 is that the model does not have a very good agreement to observations as it gives higher precipitation amounts. That is why it seems more accurate to draw conclusions from the comparison between the control and the scenario time-slices than between observation and scenario.
Looking at the results and at figure 7.5, it then appears that extremes in maximum precipitation will occur more often in the future than it has been so far.

Another approach of extreme values is to consider the Generalized Pareto Distribution (GPD) as detailed in chapter 2.

7.3 GPD results

7.3.1 Estimation of the parameters

As explained in part 6.1.2, the 90th percentile for observations is chosen as the threshold u.

```
> obs90 <- percentile(st.obs,0.90,rr=T)
> u <- obs90$annual
> u
90%
20.3
```

The parameter estimation is returned in the routine extreme.quantiles which is called by the command:

```
> extreme.quantiles(gpd.obs)
```

where gpd.obs is the object obtained by:

```
gpd.obs <- gpd.fit.iter(u,st.obs$rr)
```

The first part of the output gives then the estimation of the scale parameter σ and of the shape parameter ξ:

```
[1] "sigma= 8.94840455805428 xi= 0.145186436830011 nbr exceedances 1715"
[1] "r.values 37.22" "r.values 45.88" "r.values 57.42" "r.values 68.96"
[5] "r.values 80.5"
```

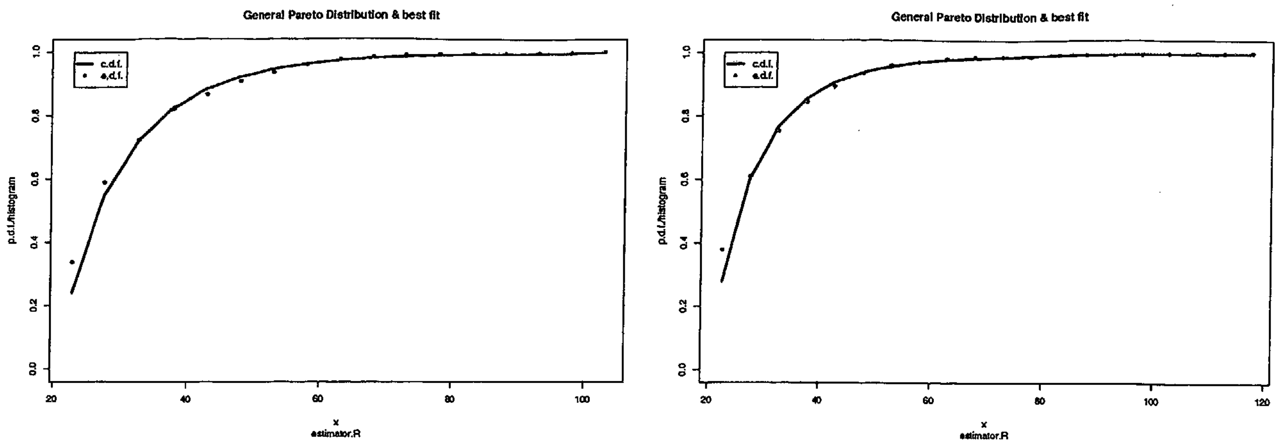


Figure 7.8: GPD cumulative distribution function (e.d.f) for maximum precipitations in Bergen using observations (*left*) and control period (*right*).

sigma			xi			nbr exceedances		
obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
8.948	7.321	8.646	0.145	0.204	0.189	1715	699	997

Table 7.6: GPD parameters (for precipitation and for Bergen station=50540) with observations (*left*) and control period (*right*)

The fit of the GPD distribution suggests that the estimation of the GPD parameters using the function gpd.fit.iter is quite adequate.
The estimators for each time-slice are given in table 7.6.

7.3.2 Return levels

Extreme.quantiles(gpd.obs) routine also provides us with the return levels for 5, 10, 25 and 50 years return periods respectively.

The complete output is indeed:

```
[1] "sigma= 8.94840455805428 xi= 0.145186436830011 nbr exceedances 1715"
[1] "r.values 37.22" "r.values 45.88" "r.values 57.42" "r.values 68.96"
[5] "r.values 80.5"
$r.p5
[1] 37.22
$r.p10
[1] 45.88
$r.p25
[1] 57.42 $r.p50
[1] 68.96
$r.p100
[1] 80.5
```

5 years			10 years			25 years			50 years			100 years		
obs	ctr	sc	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
37.2	36.1	37.9	45.8	42.8	45.5	57.4	56.1	60.7	68.9	66.2	72.1	80.5	79.5	87.3

Table 7.7: Return level values corresponding to 5, 10, 25, 50 and 100 years return period using GPD distribution (for precipitation and for Bergen station=50540)

Conclusion

As with GEV, results with GPD show higher return levels for the scenario period than for the present period as far as maximum precipitation are concerned. This trend is even more obvious when considering long-time periods.

However, the values obtained with the two methods are quite different. GEV results were based on only twenty values, which is quite poor and this can account for the gaps in return levels. A solution to this could be to keep several independant maximum per year and to compare the GEV and GPD distributions. Performing this method, which is close to the r-lagest method, it then appears that the more maxima are kept, the better the fit is. Thus, for 10 maxima, the return level plots are quite similar for the two models as figure 7.9 illustrates it. This is part of a new project which is going to be led by Rasmus E.Benestad.

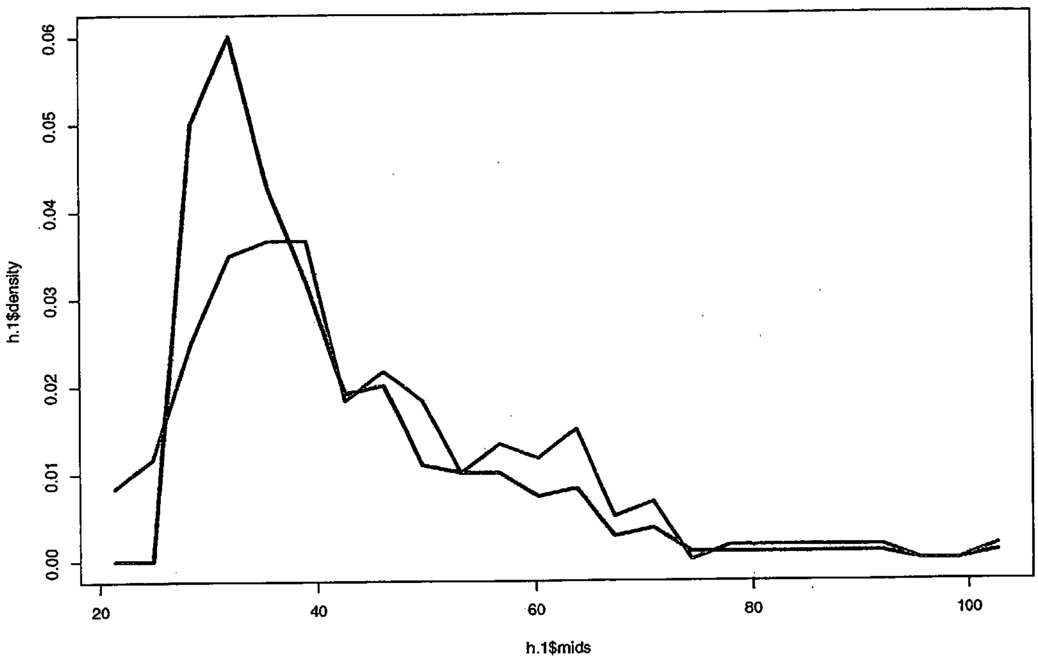


Figure 7.9: Comparison between GPD (threshold=95th percentile) and GEV (keeping 10 maxima per year) distributions for observations in Bergen

Conclusion

During this project, extreme value theory has been applied to temperature and precipitation in order to know if climate change indicators could be evidenced.

The comparison between the 1980-1999 period and the scenario time-slice 2030-2049 as well as return level values have often shown a significant increase in rainfall and temperature for the last period. However, these conclusions are just a first guess and it has been shown that GEV results were not based on a large enough number of data to really be relied on. To consolidate them, it would be necessary to extend the daily dataset or to broaden the daily dataset to other climatic elements like atmospheric pressure, humidity, sunshine and cloudiness. This is ongoing research at *met.no*.

This project has also been the occcasion to know R-programming language. Thanks to Rasmus E. Benestad, the routines implemented for GPD distribution were added to R-package.

Working at the Norwegian Meteorological Institute was really a great experience as it has enabled me to discover the meteorological field through statistical tests and extreme value theory. I have been able to apply and to develop the skills learnt at INSA.
What is more, studying climate change was something really interesting, all the more so as this summer has seen several extreme events all around the world.
It has then been a learning experience, not only from an educative point of view but also on a personal level and this experience does make me feel like working on such statistical applications.

Bibliography

[1] S. Coles, Statistical Modelling of Extreme Values (May 2000).

[2] S. Coles, website notes **http://www.stats.bris.ac.uk/masgc/software**.

[3] V. Choulakian et M. A. Stephens *'Gooddness-of-Fit Tests for the Generalized Pareto Distribution'*, Technometrics (Nov. 2001)

[4] R.L Smith,Maximum likelihood estimation in a class of non-regular cases, Biometrika, 72 (1985)

[5] Barndorff-Nielsen, O. E., and Cox, D. R. (1994). Inference and Asymptotics. Chapman Hall, London.

[6] Gary Patrik and Farrokh Guiahi, May 17-20 1998 CAS Spring Meeting, Marco Island, Florida. An Extremely Important Application of Extreme Value Theory to Reinsurance Pricing.

[7] H.Alexandersson, E.J. Førland, J. Helminen, N.Sjöblom and O.E. Tveito, 2001. Extreme value analysis in the Nordic countries, Norwegian Meteorological Institute Report 03/01 KLIMA.

Appendix

station	observations		control period		scenario period		p.value of the difference	
	mean	variance	mean	variance	mean	variance	mean.obs/ctr	mean.ctr/sc
12680	1.86	18.5	2.2	13.65	2.3	14.69	**0.0038**	0.1143
13670	1.51	13.34	1.86	9.31	1.94	9.36	**0**	**0.0375**
16610	1.3	8.57	1.51	4.12	1.61	4.63	**0**	**0.0211**
17150	2.31	24.81	2.98	34.87	3.14	36.52	**0**	0.3834
18700	2.11	21.91	2.76	26.73	2.83	26.21	**0**	0.5648
24880	1.42	11.5	1.82	11.02	1.83	10.64	**0**	0.5117
25590	2	16.26	2.36	11.81	2.46	13.1	**2e-04**	0.2766
28800	2.16	23.4	1.85	8	2.8	24.18	**0.0012**	**0**
31620	2.29	17.65	2.81	18.51	2.93	22	**1e-04**	0.1918
37230	2.69	36.66	3.39	36.16	3.51	43.57	**2e-04**	0.3983
39040	3.53	57.83	4.35	60.71	4.67	77.99	**8e-04**	0.1738
42920	5.15	81.76	6.03	78.84	6.71	102.33	**0.0095**	**0.0402**
44560	3.35	35.34	3.87	34.85	4.65	50.02	**0.0095**	**2e-04**
46610	6.39	118.65	7.41	97.11	8.61	132.96	**0.0132**	**0.0047**
50300	9.19	201.52	10.58	196.45	12.53	269.75	**0.0067**	**0.002**
50540	6.59	112.96	7.29	101.85	8.88	142.55	0.1675	**2e-04**
51590	3.76	43.71	4.23	27.95	4.92	37.31	**0.0167**	**0.0051**
60500	2.83	35.07	3.22	15.63	3.5	17.75	**0.0375**	0.1046
60990	3.78	38.3	4.67	38.78	5.36	49.24	**2e-04**	**0.0103**
65110	4.28	56.38	4.74	43.84	5.18	49.8	0.1186	**0.046**
66730	2.25	21.12	2.61	10.29	2.88	12.93	**0.0015**	**0.0039**
68340	2.58	21.1	2.89	13.34	3.21	17.59	**0.0227**	**0.0041**
69100	2.4	19.56	2.78	12.37	3.07	15.69	**0.0095**	**0.0245**
77750	2.75	40.23	2.93	14.23	3.15	16.24	0.3506	0.0965
80700	5.72	109.83	5.75	77.7	6.18	82.12	0.968	0.1826
82290	2.83	26.91	2.79	17.03	3.02	18.54	1	0.1441
86500	3.75	42.97	3.36	35.26	3.67	39.61	0.0638	0.2012
89350	1.81	12.52	1.74	6.55	1.85	7.47	0.698	0.211
90450	2.88	22.69	2.77	16.86	2.96	20.56	0.4945	0.2012
91750	1.88	14.58	1.91	7.32	1.95	7.93	0.7584	0.8287
92350	1.25	7.75	1.32	3.65	1.33	3.78	0.445	0.841
93140	1.07	6.03	1.12	2.8	1.16	3.17	0.3013	0.6017
93300	1.28	8.25	1.41	5.57	1.48	6.48	0.1918	0.2423
98550	1.75	11.56	1.79	7.31	2.05	10.11	0.3547	**0.0012**

Table 8: Precipitations annual mean values, variances and comparison between observations and control period and between scenario and control periods (means) (Legend: ***bold values indicate that the difference is significant***)

station	observations		control period		scenario period		p.value of the difference	
	mean	variance	mean	variance	mean	variance	mean.obs/ctr	mean.ctr/sc
12680	1.39	9.73	1.57	5.91	1.81	7.36	0.4783	0.1143
13670	1.02	5.51	1.17	2.6	1.33	3.14	0.1143	0.1441
16610	1.07	3.62	1.22	1.52	1.34	1.62	0.1022	0.1941
17150	2.03	18.41	2.18	13.16	2.61	18.82	0.2012	0.0583
18700	1.6	12.14	1.8	8.61	2.11	11.26	0.2084	0.0962
24880	0.87	3.18	1.03	2.64	1.18	3.28	**0.0245**	0.2084
25590	1.94	12.81	2.1	7.97	2.29	9.12	0.5291	0.2134
28800	1.5	10.41	1.19	2.2	2.04	10.02	0.0965	**0**
31620	2.25	14.75	2.72	15.89	3.01	19.27	**0.0245**	0.2423
37230	2.31	25.3	2.47	15.95	2.85	20.14	0.4135	0.2648
39040	3.68	52.79	3.88	38.55	4.5	47.35	0.4291	0.1344
42920	6.96	111.99	7.26	97.86	7.82	113.85	0.4612	0.445
44560	3.72	33.03	3.5	27.72	4.01	39.07	0.5648	0.1738
46610	8.71	161.8	8.86	131.57	9.62	163.33	0.6783	0.445
50300	11.66	246.08	11.79	232.83	12.89	301.77	0.6783	0.5468
50540	8.34	133.61	7.45	103.76	8.64	156.36	0.4239	0.1207
51590	5.11	60.73	5.16	37.98	5.53	44.97	0.698	0.5648
60500	3.92	49.84	4.51	24.02	4.72	26.04	0.1918	0.6588
60990	4.55	44.5	5.62	43.4	6.13	61.98	**0.0375**	0.3408
65110	5.2	69.92	5.71	56.11	6.22	67.12	0.3928	0.2888
66730	2.18	18.57	2.85	10.99	3.23	13.33	**0.0235**	0.1595
68340	2.2	15.5	2.43	7.3	2.63	8.3	0.1918	0.2888
69100	2.38	17.52	2.68	10.11	2.86	11.86	0.1653	0.3141
77750	3.85	67.5	3.98	22.85	3.97	22.72	0.5315	0.9467
80700	6.83	135.33	6.71	97.88	6.9	99.49	0.9254	0.718
82290	3	25.08	2.93	17.36	3.05	19.33	0.718	0.5648
86500	5.18	57.9	4.79	64.99	5.07	71.18	0.6331	0.862
89350	2.21	14.81	2.03	7.53	2.15	9.04	0.2423	0.6205
90450	3.52	26.66	3.31	21.88	3.59	27.81	0.4291	0.6205
91750	2.22	17.25	2.1	6.9	2.15	7.14	0.4945	0.6588
92350	1.33	8.95	1.33	2.61	1.34	2.49	0.698	0.7764
93140	1.17	5.42	1.15	2.15	1.11	1.78	0.7764	0.8604
93300	1.09	4.14	1.09	1.74	1.08	1.5	0.6205	1
98550	2.01	10.05	1.72	3.28	2	4.39	0.0962	**0.009**

Table 9: Precipitations Winter mean values, variances and comparison between observations and control period and between scenario and control periods (means) (Legend: *bold values indicate that the difference is significant*)

station	observations		control period		scenario period		p.value of the difference	
	mean	variance	mean	variance	mean	variance	mean.obs/ctr	mean.ctr/sc
12680	2.47	25.92	3.23	25.9	3.22	24.48	**0.0016**	0.6849
13670	2.41	24.91	3.12	22.68	3.16	19.85	**0.0014**	0.6783
16610	2.03	17.15	2.4	10.27	2.6	11.11	**0.002**	0.1105
17150	2.49	29.29	4.22	75.54	4.3	67.66	**0**	0.7557
18700	2.63	32.24	4.25	60.31	4.15	50.29	**0**	0.9893
24880	2.27	22.2	3.13	25.76	2.93	22.32	**3e-04**	0.7381
25590	2.28	21.03	2.94	19.33	2.95	19.94	**0.002**	0.6395
28800	2.79	34.59	3.05	18.83	3.93	43.63	0.1081	**7e-04**
31620	2.59	23.86	3.4	27.24	3.42	33.08	**0.0017**	0.698
37230	2.82	40.38	4.24	54.96	4.19	67.44	**2e-04**	0.9467
39040	3.04	59.33	4.6	83.76	4.93	120.95	**0.002**	0.445
42920	3.4	42.9	4.99	50.16	5.84	82.14	**1e-04**	**0.0326**
44560	2.91	36.15	4.07	32.61	5.21	48.17	**0**	**2e-04**
46610	4.38	69.13	6.32	57.61	7.74	86.97	**1e-04**	**0.0043**
50300	6.58	114.9	9.43	132.32	11.65	184.88	**0**	**0.0043**
50540	4.98	78.28	6.84	72.42	8.76	115.06	**3e-04**	**0.001**
51590	2.65	23.77	3.62	17.48	4.37	24.29	**2e-04**	**0.0061**
60500	1.82	13.12	2.36	8.89	2.64	10.62	**1e-04**	**0.0239**
60990	2.75	24.46	3.67	24.69	4.36	32.35	**2e-04**	**0.0056**
65110	3.33	33.34	4.17	32.06	4.69	36.8	**0.0048**	**0.0326**
66730	3.01	32.98	3.45	17.84	3.93	22.4	**0.0305**	**0.0211**
68340	3.52	32.57	4.32	26.7	4.92	36.22	**0.0013**	**0.0195**
69100	2.81	26.96	3.78	20.69	4.22	27.05	**0**	**0.0439**
77750	2.09	16.09	2.58	10	2.74	11.93	**0.009**	0.3834
80700	4.29	68.37	5.13	51.35	5.24	56.35	0.0675	0.9254
82290	2.47	24.56	2.8	16.4	2.92	19.53	0.2616	0.7557
86500	1.96	16.86	2.04	10.86	1.96	9.61	0.4993	0.6395
89350	1.72	12.38	1.93	8.25	1.85	8.58	0.1653	0.6588
90450	2.3	17.39	2.5	14.22	2.31	14.94	0.1344	0.4135
91750	1.73	14.05	2.06	10.08	1.94	10.82	**0.0195**	0.3941
92350	1.33	9.44	1.58	6.94	1.5	7.05	0.046	0.4135
93140	1.18	9.45	1.37	5.38	1.39	6.43	0.0559	0.8831
93300	1.77	16.48	2.15	14.58	2.26	17.34	0.0859	0.6456
98550	1.73	19.67	2.18	17.63	2.37	23.22	0.0559	0.6017

Table 10: Precipitations summer mean values, variances and comparison between observations and control period and between scenario and control periods (means)(*Legend: bold values indicate that the difference is significant*)

station	observations		control period		scenario period		p.value of the difference	
	mean	variance	mean	variance	mean	variance	mean.obs/ctr	mean.ctr/sc
12680	1.28	9.35	1.23	3.02	1.21	3.87	0.9566	0.841
13670	1.06	8	1.23	2.67	1.23	3.38	0.0056	0.841
16610	0.82	3.74	0.98	1.02	0.93	1.18	**0.0106**	0.5428
17150	1.69	16.06	1.93	12.63	1.83	13.31	0.1417	0.3547
18700	1.53	11.72	1.75	8.15	1.68	9.34	0.1417	0.4818
24880	0.91	5.59	1.12	3.46	1.09	3.65	**9e-04**	0.8392
25590	1.55	12.59	1.8	5.73	1.73	5.81	0.0559	0.841
28800	1.64	13.58	1.14	1.77	1.87	10.73	**0.0012**	**0**
31620	1.67	10.73	1.91	8.28	1.83	7.69	0.084	0.7994
37230	1.9	18.91	2.33	19.54	2.22	17.69	**0.043**	0.6205
39040	2.51	34.13	2.86	28.91	2.82	29.98	0.2423	0.7994
42920	3.47	50.38	3.77	28.75	3.72	30.71	0.3547	0.9042
44560	2.18	18.22	2.67	18.26	2.83	19.94	0.1493	0.4612
46610	4.3	64.48	5.28	51.01	5.45	54.7	0.1075	0.8392
50300	6.61	135.12	8.3	135.68	8.78	142.7	0.0763	0.6205
50540	4.94	77.27	5.89	82.62	6.35	81.21	0.2565	0.3834
51590	2.58	29.34	2.99	14.4	3.11	15.4	0.2534	0.766
60500	2.16	26.43	2.56	9.51	2.45	9.23	**0.008**	0.6783
60990	2.71	24.13	3.53	24.42	3.44	19.25	**0.0112**	0.9893
65110	3.04	39.46	3.7	23.35	3.43	19.8	**0.0154**	0.3834
66730	1.63	13.03	1.89	3.95	1.78	4.17	0.0652	0.5978
68340	1.83	12.76	2.17	6.13	2.09	6.01	**0.0283**	0.718
69100	1.7	11.93	2.17	6.84	2.1	6.38	**0.0087**	0.6783
77750	1.74	21.88	1.96	5.66	2	6.33	**0.0225**	0.7972
80700	4.07	53.43	4.22	35.86	4.4	37.24	0.5831	0.8817
82290	1.94	14.72	2.05	8.78	2.13	8.49	0.4945	0.8831
86500	2.96	29.64	2.77	17.84	3.09	23.12	0.3819	0.6588
89350	1.09	6.65	1.02	1.77	1.1	1.9	0.8831	0.57
90450	1.96	13.58	1.92	6.71	2.14	7.92	0.8831	0.3983
91750	1.17	6.67	1.19	1.92	1.25	2.4	0.3013	0.6205
92350	0.77	3.49	0.86	0.98	0.88	1.22	**0.0294**	0.6949
93140	0.64	2.35	0.74	0.76	0.74	0.89	**0.0468**	0.8604
93300	0.88	3.99	0.99	1.32	0.98	1.47	**0.043**	0.8201
98550	1.22	4.92	1.2	2.35	1.27	3.04	0.5792	0.2976

Table 11: Precipitations spring mean values, variances and comparison between observations and control period and between scenario and control periods (means) (*Legend: bold values indicate that the difference is significant*)

station	observations		control period		scenario period		p.value of the difference	
	mean	variance	mean	variance	mean	variance	mean.obs/ctr	mean.ctr/sc
12680	2.26	27.46	2.79	17.03	2.96	20.36	**0.0477**	0.445
13670	1.55	13.55	1.92	6.83	2.05	8.68	**0.0245**	0.3273
16610	1.3	8.89	1.41	2.53	1.58	3.08	0.2853	0.0565
17150	3.04	34.48	3.58	34.53	3.82	42.5	0.0718	0.445
18700	2.68	30.27	3.25	25.46	3.37	30.08	**0.0326**	0.6783
24880	1.63	13.6	2.02	9.4	2.11	11.06	**0.0326**	0.5468
25590	2.24	18.25	2.61	13.42	2.84	16.62	0.0911	0.2888
28800	2.71	33.52	2.02	6.79	3.37	29.32	**1e-04**	**0**
31620	2.67	20.94	3.2	21.33	3.45	26.26	**0.043**	0.372
37230	3.75	60.26	4.53	50.24	4.78	64.91	**0.043**	0.5117
39040	4.89	82.21	6.07	86.24	6.42	107.18	**0.0112**	0.6949
42920	6.69	108.41	8.09	126.8	9.45	164.38	0.0596	**0.035**
44560	4.62	50.87	5.24	57.39	6.57	85.28	0.1207	**0.0056**
46610	8.25	163.63	9.17	137.52	11.65	206.07	0.2211	**0.0061**
50300	11.99	284.8	12.79	272.44	16.78	417.07	0.445	**0.0035**
50540	8.15	152.98	9	143.67	11.76	203.01	0.1582	**0.0026**
51590	4.75	56.11	5.15	38.36	6.65	57.72	0.2534	**0.0039**
60500	3.45	48.34	3.45	17.25	4.18	21.34	0.7584	**0.0161**
60990	5.13	55.89	5.86	58.08	7.51	73.58	0.1344	**0.0043**
65110	5.53	77.9	5.37	61.2	6.36	69.81	0.798	**0.0283**
66730	2.2	19.01	2.26	6.99	2.6	9.32	0.2637	0.0524
68340	2.8	22.07	2.62	10.38	3.19	15.33	0.968	**0.0043**
69100	2.73	21.09	2.51	10.39	3.09	15.19	0.9254	**0.0041**
77750	3.36	53.14	3.2	16.22	3.88	21.34	0.9447	**0.0112**
80700	7.72	173.55	6.93	120.82	8.2	126.89	0.4135	0.0787
82290	3.91	41.36	3.36	24.7	3.97	25.15	0.1738	0.0583
86500	4.95	60.82	3.87	43	4.55	48.6	**0.0392**	0.0763
89350	2.25	15.45	1.99	7.97	2.31	9.49	0.3273	0.1207
90450	3.75	30.99	3.35	23.26	3.81	29.41	0.3273	0.2534
91750	2.42	19.72	2.29	9.68	2.46	10.6	0.698	0.2315
92350	1.56	8.85	1.49	3.78	1.61	4.06	0.5978	0.1762
93140	1.28	6.61	1.23	2.71	1.39	3.29	0.4903	0.0524
93300	1.4	7.89	1.43	3.79	1.62	4.56	0.9467	0.043
98550	2.04	11.16	2.04	5.41	2.55	8.85	0.5831	**7e-04**

Table 12: Precipitations autumn mean values, variances and comparison between observations and control period and between scenario and control periods (means) (*Legend: bold values indicate that the difference is significant*)

station	**observations**		**control period**		**scenario period**		**p.value of the difference**			
	mean	variance	mean	variance	mean	variance	mean obs/ctr	var obs/ctr	mean ctr/sc	var ctr/sc
12680	3.84	79.75	3.68	62.14	4.25	58.59	0.6609	**6e-04**	**1e-04**	0.7649
13670	0.86	64.71	0.69	49.59	3.72	53.68	0.41	**0**	**0.0127**	0.1622
16610	0.24	60.3	-0.1	42.24	0.37	40.9	**0.0792**	**0.0013**	**8e-04**	0.6585
17150	6.1	65.08	6.14	45.94	6.93	41.31	0.8821	**5e-04**	**0.0073**	0.1817
18700	6.09	67.31	6	51.21	6.59	47.83	0.7163	**7e-04**	**0.0045**	0.3326
24880	3.45	100.54	3.23	75.91	3.85	71.31	0.4524	**0.0011**	**0.0049**	0.7922
25590	1.34	63.59	1.21	46.52	1.83	43.59	0.5508	**2e-04**	**0.0016**	0.5818
28800	3.81	77.56	0.83	56	4.37	52.31	**0**	**0**	**0.0014**	0.1507
31620	0.9	57.4	0.86	43.22	1.46	40.43	0.5315	**3e-04**	**5e-04**	0.5323
37230	5.49	60.36	5.36	45.66	5.97	41.73	0.6293	**0.0018**	**0.0082**	0.401
39040	7.1	49.71	7.41	27.45	8.25	24.67	0.2047	**2e-04**	**6e-04**	0.1524
42920	3.47	56.02	2.87	35.36	3.47	32.14	**0.0116**	**0.0013**	**3e-04**	0.2332
44560	7.68	32.98	7.64	23.24	8.2	21.04	0.8374	**3e-04**	**0.0014**	0.1831
46610	6.58	47.83	6.32	36.54	6.86	33.85	0.1657	**4e-04**	**4e-04**	0.2785
50300	4.45	41.29	4.42	30.54	5	28.25	0.8485	**1e-04**	**0.0042**	0.4022
50540	7.88	31.73	8.07	25.83	8.82	24.04	0.255	**0**	**0.0774**	0.5592
51590	5.29	60.52	4.99	44.5	5.58	41.53	0.1715	**3e-04**	**1e-04**	0.411
60500	7.05	36.44	7.07	24.48	7.51	23.68	0.8341	**2e-04**	**2e-04**	0.7334
60990	7.1	23.08	7.74	15.29	8.3	14.84	**1e-04**	**0**	**0.0133**	0.7791
65110	5.33	42.14	5.46	30.42	5.99	29.19	0.2601	**2e-04**	**0**	0.8127
66730	2.7	54	2.92	36.12	3.45	34.98	0.1692	**2e-04**	**2e-04**	0.8952
68340	4.44	54.14	4.55	38.22	5.1	36.86	0.5542	**5e-04**	**2e-04**	0.7701
69100	5.74	52.55	5.96	38.32	6.5	36.56	0.2453	**6e-04**	**2e-04**	0.6611
77750	1.99	85.36	1.83	65.82	2.59	61.84	0.5311	**1e-04**	**1e-04**	0.8483
80700	5.3	37.85	5.4	28.93	6.04	27.65	0.5627	**0**	**6e-04**	0.4635
82290	4.88	40.26	5.56	32.26	6.51	30.52	**6e-04**	**0**	**0.0188**	0.3704
86500	4.43	36.85	4.51	32.23	5.28	30.16	0.6733	**0**	**0.0756**	0.2998
89350	1.22	93.72	1.27	76.07	2.28	68.74	0.8593	**0**	**6e-04**	0.3551
90450	2.77	44.35	3.46	38.46	4.38	35.5	2e-04	**0**	**0.01190**	0.4473
91750	1.55	76.54	1.95	60.72	2.89	55.76	0.0683	**0**	**5e-04**	0.3476
92350	2.99	47.05	3.16	40.12	3.93	37.27	0.3211	**0**	**0.0029**	0.3341
93140	1.64	78.82	1.84	66.05	2.78	60.87	0.3771	**0**	**6e-04**	0.4309
93300	-2.13	112.53	-2.02	89.95	-0.98	82.59	0.7339	**0**	**1e-04**	0.2838
98550	1.6	34.85	2.06	41.74	4.13	38.75	**0.0288**	**0**	0.361	0.2843

Table 13: Temperatures: Annual mean values, variances and comparison between observations and control period and between scenario and control periods (means and variances) (*Legend: bold p-values indicate that the difference is significant.*)

station	observations		control period		scenario period		p.value of the difference			
	mean	variance	mean	variance	mean	variance	mean obs/ctr	var obs/ctr	mean ctr/sc	var ctr/sc
12680	-6.35	36.45	-6.05	11.26	-5.25	9.24	0.6583	**0.0328**	**2e-04**	0.566
13670	-7.45	26.27	-7.25	9.28	-4.57	13.25	0.6874	**0**	**0.008**	0.1279
16610	-7.39	27.31	-7.23	9.1	-6.6	8.76	0.7484	**0.0469**	**0.0016**	0.5351
17150	-2.55	35.01	-1.47	19.99	-0.17	17.23	0.1415	**0.0193**	0.3371	0.1686
18700	-2.96	26.85	-2.45	12.44	-1.57	10.21	0.4071	**0.0309**	0.1486	0.1253
24880	-7.98	56.76	-7.58	19.02	-6.59	16.62	0.6122	**0.041**	**0.0104**	0.2817
25590	-6.45	37.96	-6.16	14.19	-5.26	11.4	0.6382	**0.0161**	**6e-04**	**0.4409**
28800	-5.77	44.94	-8.09	14.67	-4.05	17.14	**0.0026**	**0**	**0.0012**	0.6379
31620	-6.51	28.45	-6.25	12.45	-5.39	10.19	0.7057	**0.0219**	**0.0043**	0.2934
37230	-2.58	34.86	-2.21	18.8	-1.16	15.74	0.6071	**0.0373**	**0.032**	**0.0907**
39040	-0.24	27.94	1.94	16.48	3.31	15.64	**0.0014**	**0.0079**	**0.0117**	0.3711
42920	-4	38.03	-3.54	18.47	-2.42	16.64	0.4777	**0.0204**	**2e-04**	0.2169
44560	1.88	17.41	2.36	10.98	3.33	10.44	0.3158	**0.0139**	**0.03**	0.3872
46610	-0.8	23.49	-0.72	11.47	0.17	10.19	0.8744	**0.0145**	**4e-04**	0.3761
50300	-1.89	21.17	-1.71	10.41	-0.8	9.6	0.6952	**0.0107**	**0.012**	0.5741
50540	2.26	13.33	2.44	8.07	3.47	7.72	0.6439	**0.0039**	0.2369	0.676
51590	-2.85	41.61	-2.9	16.44	-1.97	14.88	0.9432	**0.0258**	**0**	0.8913
60500	1.25	18.66	1.37	5.42	1.92	4.79	0.7622	**0.0251**	**8e-04**	0.699
60990	2.6	9.7	3.76	4.4	4.41	4.18	**0.0013**	**0.0062**	0.176	0.8863
65110	-0.97	20.81	-0.59	6.53	0.06	6.14	0.4348	**0.0222**	**0**	0.3714
66730	-4.59	27.19	-4.26	7.54	-3.67	6.82	0.5511	**0.0445**	**1e-04**	0.4186
68340	-2.79	29.55	-2.41	8.99	-1.74	8.08	0.4699	**0.0369**	**0**	0.1678
69100	-1.58	31.71	-1.2	9.7	-0.49	8.67	0.495	**0.0316**	**0**	0.1151
77750	-7.01	69.77	-7.43	23.17	-6.42	20.1	0.6263	**0.0172**	**0**	0.7948
80700	-0.3	20.04	-0.3	7.43	0.36	6.44	0.9925	**0.0091**	**3e-04**	0.5771
82290	-1.14	20.92	-0.43	11.29	0.64	9.29	0.14	**0.0019**	0.0638	0.71
86500	-1.22	17.67	-1.46	9.1	-0.52	7.6	0.6197	0.0013	0.087	0.0427
89350	-8.42	68.07	-8.58	21.9	-7.13	16.35	0.8166	**4e-04**	**2e-04**	**0.0091**
90450	-3.5	19.1	-3.02	8.31	-1.89	6.37	0.2265	**1e-04**	**0.0063**	0.0556
91750	-7.02	46	-6.7	13.95	-5.46	11.03	0.5928	**3e-04**	**0.0017**	**0.0365**
92350	-3.61	22.47	-3.6	8.72	-2.63	6.98	0.9911	**3e-04**	0.1358	0.1295
93140	-7.12	45.58	-7.14	15.34	-5.85	12.62	0.9742	**3e-04**	**0.0135**	0.136
93300	-12.79	71.56	-12.65	19.88	-11.21	15.98	0.9934	**2e-04**	**9e-04**	**0.0319**
98550	-4.32	14.85	-4.81	7.01	-2.53	6.34	0.2256	**0**	0.0684	0.3772

Table 14: Temperatures: Winter mean values, variances and comparison between observations and control period and between scenario and control periods (means and variances) (<u>Legend</u>: *bold p-values indicate that the difference is significant.*)

station	observations		control period		scenario period		p.value of the difference			
	mean	variance	mean	variance	mean	variance	mean obs/ctr	var obs/ctr	mean ctr/sc	var ctr/sc
12680	13.98	9.7	13.58	1.84	13.8	1.49	0.1807	0.0696	**0**	0.9296
13670	10.25	10.79	9.81	2.03	13.08	1.85	0.0645	**0**	**0**	0.1634
16610	9.2	11.58	8.28	2.42	8.5	2.32	**9e-04**	0.0294	**0**	**0.0187**
17150	15.28	7.97	14.35	1.87	14.65	1.36	**0.0027**	0.0617	**0**	0.1581
18700	15.69	9.35	14.96	1.69	15.18	1.33	**0.0114**	0.1043	**0**	0.2375
24880	14.56	9.47	13.92	2.07	14.17	1.64	**0.0074**	**0.0448**	**0**	0.5999
25590	10.39	9.45	9.93	1.89	10.2	1.52	**0.0468**	**0.0317**	**0**	0.4718
28800	13.7	9.23	10.06	1.63	13.24	1.44	**0**	**0**	**0**	0.4435
31620	9.88	9.92	9.33	1.89	9.59	1.56	**0.0309**	**0.0323**	**0**	0.2639
37230	14.4	8.74	13.74	1.92	13.98	1.53	**0.0105**	0.0746	**0**	0.0885
9040	15.11	7.13	13.57	1.83	13.99	1.44	**0**	**0.0044**	**0**	0.551
42920	11.95	8.93	8.8	16.2	8.98	15.92	**0**	0.0976	0.6744	0.1092
44560	13.86	7.37	13.26	1.29	13.43	1.17	**0.0146**	0.0567	**0**	0.3549
6610	14.34	7.97	13.8	1.16	13.98	0.94	**0.0284**	0.068	**0**	0.2017
50300	11.65	9.41	11.24	1.36	11.46	1.09	0.0937	**0.0394**	**0**	0.2904
50540	14.12	7.98	14.16	2.38	14.59	1.91	0.861	**0.018**	**0**	0.6917
51590	13.76	8.92	13.17	1.26	13.39	0.97	**0.0137**	**0.0295**	**0**	0.4041
60500	13.55	7.4	13.21	0.94	13.38	0.83	0.0979	**0.0177**	**0**	0.0501
60990	12.29	6.24	12.26	2.18	12.65	2.06	0.8976	**0.0027**	**0**	0.4668
65110	12.72	9.49	12.44	1.76	12.68	1.49	0.3366	**0.0372**	**1e-04**	**0.0781**
66730	10.93	11.32	10.47	1.93	10.72	1.57	0.0935	**0.0438**	**0**	**0.0288**
68340	12.49	13.3	12.2	2.57	12.47	2.04	0.341	0.0531	**0**	**0.0164**
69100	13.58	10.57	13.58	2.2	13.83	1.72	0.9795	0.0566	**0**	**0.0175**
77750	11.89	12.03	11.74	2.64	12.1	1.74	0.5734	**0.0088**	**0.0023**	**0.0039**
80700	11.97	11.6	11.98	2.69	12.37	2.07	0.9747	**0.0082**	**0.0032**	**0.0416**
82290	11.89	10.54	12.23	3.03	12.98	2.34	0.2656	**3e-04**	**0.0029**	0.1047
86500	11.43	9.88	11.52	2.83	12.02	2.7	0.7489	**0.0043**	**4e-04**	0.2568
89350	11.84	14.42	12.07	3.53	12.66	3.07	0.4428	**0.0032**	**0.0032**	0.0874
90450	10.63	12.76	11.43	4.03	12.07	3.67	0.0074	**0.0048**	**0.0082**	0.192
91750	11.4	14.07	11.72	3.67	12.29	3.5	0.2551	**0.0028**	**5e-04**	0.2623
92350	10.97	10.48	11.21	3.19	11.7	3.04	0.2988	**0.0031**	**0.001**	0.4635
93140	11.83	14.3	12.08	4.05	12.65	3.95	0.2958	**0.0032**	**1e-04**	0.417
93300	9.76	16.75	9.92	3.9	10.51	3.47	0.4867	**0.0035**	**1e-04**	0.336
98550	8.22	8.27	10	7.55	11.76	5.69	**0**	**0**	0.816	0.4926

Table 15: Temperatures: Summer mean values, variances and comparison between observations and control period and between scenario and control periods (means and variances) (*Legend: bold p-values indicate that the difference is significant.*)

station	observations		control period		scenario period		p.value of the difference			
	mean	variance	mean	variance	mean	variance	mean obs/ctr	var obs/ctr	mean ctr/sc	var ctr/sc
12680	3.54	35.02	3.43	20.88	3.87	21.91	0.7716	0.107	**5e-04**	0.8113
13670	-0.06	32.07	-0.19	18.8	2.61	24.66	0.5984	**0**	**1e-04**	0.292
16610	-1.15	31.64	-1.36	18.11	-0.86	20.48	0.3893	**0.049**	**0**	0.4577
17150	5.2	29.73	5.43	18.42	6.03	19.27	0.4758	0.0627	**0.0018**	0.4798
18700	5.48	31.28	5.51	19.82	5.98	20.79	0.9148	0.0987	**0.0023**	0.825
24880	4.06	34.11	3.87	20.3	4.35	21.56	0.525	0.1055	**5e-04**	0.7674
25590	0	31.35	-0.19	17.79	0.36	19.6	0.4744	0.0517	**1e-04**	0.6452
28800	3.55	31.79	0.25	18.19	4.01	20.43	**0**	**0**	**5e-04**	0.4126
31620	-0.78	27.96	-0.98	16.53	-0.37	17.86	0.4886	0.0512	**0.0016**	0.285
37230	4.26	28.05	4.28	17.59	4.75	18.35	0.9435	0.1307	**0.0118**	0.8023
39040	5.76	22.58	6.28	11.03	6.97	12.1	0.0937	**0.0297**	**0.0046**	0.8602
42920	2.17	24.89	2.03	13.38	2.48	14.36	0.598	0.0734	**6e-04**	0.3182
44560	6.34	15.94	6.31	8.34	6.77	8.8	0.8731	**0.0229**	**0.0114**	**0.0346**
46610	5.9	20.7	5.56	12.8	6.06	14.48	0.0807	**0.0273**	**8e-04**	0.2203
50300	3.13	19.99	3.08	11.98	3.68	13.12	0.8426	**0.0165**	**0.0061**	0.2004
50540	6.72	16.75	7.07	12.11	7.89	13.58	0.1027	**0.0025**	0.1189	0.1902
51590	4.86	23.2	4.48	14.24	5.06	16.49	0.0761	**0.0182**	**2e-04**	0.6946
60500	6.14	19.38	6.1	9.4	6.58	11.15	0.8395	**0.025**	**0**	0.7128
60990	5.42	11.59	6.01	5.7	6.64	6.57	**0.0016**	**0.0016**	**2e-04**	0.8984
65110	4.35	20.11	4.09	12.55	4.66	13.9	0.2656	**0.0153**	**0.0023**	0.7447
66730	1.84	26.93	2.41	8.93	2.95	10.77	**0.0309**	**0.0046**	**0**	0.1815
68340	3.42	26.88	3.43	15.62	3.99	17.61	0.9673	**0.0278**	**0**	0.9695
69100	4.91	23.46	5.12	14	5.64	15.34	0.3983	**0.0273**	**0**	0.8677
77750	1.17	34.95	1.09	24.19	1.81	26.36	0.7823	**0.0132**	**0**	0.5223
80700	3.77	20.04	3.97	13.42	4.7	13.84	0.4117	**0.0054**	**1e-04**	0.7144
82290	3.25	19.43	4.16	16.08	5.1	16.96	**9e-04**	**9e-04**	**5e-04**	0.5964
86500	2.52	15.1	2.67	12.75	3.43	11.97	0.5939	**0.0016**	0.0747	0.4212
89350	0.47	35.76	0.48	29.99	1.36	28.14	0.9779	**0.0022**	**0.001**	0.699
90450	1.04	18.73	1.65	14.97	2.55	14.09	0.0396	**4e-04**	**0.0015**	0.8012
91750	0.52	31.31	0.79	24.08	1.62	22.83	0.4262	**0.002**	**0.0033**	0.7814
92350	1.15	20.04	1.37	15.91	2.1	15.22	0.4417	**0.0015**	**0.0133**	0.7008
93140	-0.12	29.63	0.23	25.24	1.07	23.74	0.2652	**0.0012**	**0.0097**	0.5234
93300	-3.85	44.47	-3.68	35.07	-2.78	32.62	0.6161	**8e-04**	**0.0011**	0.3191
98550	-0.26	13.21	-0.33	10.38	1.36	10.14	0.7851	**0**	0.072	0.8372

Table 16: Temperatures: Spring mean values, variances and comparison between observations and control period and between scenario and control periods (means and variances) (*Legend: bold p-values indicate that the difference is significant.*)

station	observations		control period		scenario period		p.value of the difference			
	mean	variance	mean	variance	mean	variance	mean obs/ctr	var obs/ctr	mean ctr/sc	var ctr/sc
12680	3.52	32.92	3.76	21.87	4.57	20.16	0.4939	**0**	0.0838	0.2551
13670	0.49	31.89	0.41	21.28	3.75	17.5	0.762	**0**	**0.0045**	0.4727
16610	0.15	30.51	-0.08	16.63	0.43	15.84	0.3904	**0.002**	**3e-04**	0.9888
17150	6.3	28.05	6.25	17.59	7.21	16.53	0.8668	**0**	**0.0201**	0.2441
18700	5.98	27.64	5.99	18.93	6.75	18.22	0.9609	**0**	0.0058	0.4796
24880	2.92	48.27	2.68	30.45	3.45	29.55	0.4383	**1e-04**	**2e-04**	0.9028
25590	1.27	31.5	1.23	19.79	2.01	19.33	0.9072	**1e-04**	**5e-04**	0.4942
28800	3.57	35.25	1.09	24.5	4.29	20.61	**0**	**0**	0.0873	**0.0242**
31620	1.32	25.2	1.34	16.19	2.03	15.51	0.5127	**0**	**0.0024**	0.7232
37230	5.73	24.18	5.63	15.49	6.32	14.69	0.6779	**0**	**0.0071**	0.6125
39040	7.62	21.45	7.85	11.08	8.72	10.39	0.2979	**0**	**0.005**	0.2176
42920	4.24	23.97	4.21	14.66	4.84	13.77	0.8816	**1e-04**	**0.0013**	0.5657
44560	8.54	16.95	8.63	10.16	9.27	9.53	0.677	**0**	**0.001**	0.7663
46610	6.75	24.34	6.64	14.52	7.21	13.61	0.6571	**1e-04**	**2e-04**	0.7817
50300	4.9	20.97	5.05	12.19	5.66	11.6	0.5058	**1e-04**	**4e-04**	0.8527
50540	8.34	16.92	8.59	10.69	9.34	9.91	0.2426	**0**	**0.0158**	0.2465
51590	5.23	30.72	5.21	16.68	5.86	15.39	0.9464	**1e-04**	**0**	0.4884
60500	7.36	24.14	7.61	10.82	8.14	11.06	0.3641	**9e-04**	**0**	0.4471
60990	8	13.98	8.92	8.21	9.49	8.34	**1e-04**	**1e-04**	**1e-04**	0.8763
65110	5.43	23.12	5.91	13.45	6.54	13.3	**0.0722**	**4e-04**	**0.0037**	0.8751
66730	2.62	29.5	3.07	17.11	3.8	16.79	0.1172	**4e-04**	**0.0038**	0.5929
68340	4.66	29.3	4.98	17.27	5.67	17.02	0.2308	**2e-04**	**6e-04**	0.4784
69100	5.89	29.09	6.35	17.26	7.03	16.92	0.0946	**2e-04**	**0.0015**	0.4639
77750	1.75	45.63	1.92	28.86	2.85	26.83	0.5411	**1e-04**	**0.0103**	0.2381
80700	5.66	21.43	5.94	14.07	6.73	13.57	0.2146	**0**	**0.0032**	0.5137
82290	5.43	21.92	6.27	15.79	7.31	14.53	8e-04	**0**	**0.0139**	0.1686
86500	4.9	19.95	5.31	15.5	6.19	15.01	0.1059	**0**	**0.0221**	0.6403
89350	0.81	50.96	1.1	34.46	2.23	30.44	0.341	**0**	**8e-04**	0.2742
90450	2.82	22.81	3.77	17.76	4.8	15.84	1e-04	**0**	**0.0022**	0.2189
91750	1.31	42.98	2	29.63	3.08	26.07	0.0188	**0**	**2e-04**	0.2435
92350	3.32	24.71	3.68	18.71	4.55	16.74	0.1277	**0**	**0.001**	0.2415
93140	1.78	41.59	2.17	31.22	3.26	28.15	0.1685	**0**	**1e-04**	0.2512
93300	-2.16	62.19	-1.7	42.57	-0.45	38.14	0.1534	**0**	**0**	0.2661
98550	2.65	19.71	3.39	24.27	5.91	19.43	0.0064	**0**	0.7032	0.0634

Table 17: Temperatures: Autumn mean values, variances and comparison between observations and control period and between scenario and control periods (means and variances) (*Legend: bold values indicate that the difference is significant.*)

station	95 percentile			99 percentile			99.1 percentile		
	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
12680	10	9.5	9.7	20.53	17.3	18	21	17.6	18.82
13670	8.5	7.21	7.6	17.8	13.71	14.2	18.35	14.7	14.62
16610	6.5	4.8	5.1	13.99	9.8	10.3	14.53	10.02	10.72
17150	13	13.6	14.6	24.19	26.6	28	25	27.7	29.19
18700	11.48	12.31	12.6	22.5	23.5	23.7	23.55	24.02	24.47
24880	7.7	8	7.71	16	15.6	15.6	17.03	16.1	16.4
25590	10.4	9.3	9.7	18.4	15.41	16.8	18.85	16.12	17.2
28800	11.5	6.9	12.2	23.8	12.9	23.1	24.2	13.22	23.84
31620	11	11.7	12.4	20.09	19.8	21.8	20.59	20.1	22.2
37230	14.4	16	16.4	28.5	28.5	31.6	29.7	29.6	33.27
39040	18.3	20.7	21.6	35.69	36.5	40.71	36.8	37.02	41.74
42920	24.5	24.51	27.21	41.06	40.5	47.31	42.17	41.22	49.53
44560	15.3	15.41	18.3	25.8	27.8	33.2	26.3	28.3	33.64
46610	28.7	28	31.5	49.39	44.9	52.9	50.43	45.52	53.42
50300	38.27	39.9	44.61	61.99	61.3	75.1	63.93	63.74	76.92
50540	28.06	27.4	32.31	48.77	45.1	54.3	50.41	45.94	55.32
51590	17.2	15.4	17.3	29.99	23.2	27.1	31.2	23.6	27.8
60500	14.57	11.4	12	27.89	17.9	19.3	28.53	18.12	19.52
60990	15.4	16.3	18.2	29.2	28	32.6	30.05	28.57	33.2
65110	18.4	17.2	19.01	34.5	30.6	32.51	35.61	31.5	34.4
66730	11.3	8.5	9.7	21.6	14.6	16.3	22.5	15	16.82
68340	11.8	9.6	10.7	20.53	17.3	19.4	21.15	17.7	19.8
69100	11.1	9.7	10.6	21.3	15.8	17.5	21.91	16.2	18.14
77750	13.3	10.6	11.2	29.56	17	18.4	32.25	17.4	18.72
80700	24	22.01	23.61	46.4	40.9	41.01	47.33	42.39	42.72
82290	12.7	10.5	11	23.39	19	19.3	24.15	19.7	20
86500	17.2	13.6	14.8	30.8	27.4	29.4	31.97	28.02	30.7
89350	8.9	7	7.2	16.2	11.5	12.8	16.93	11.9	13.32
90450	13	10.9	11.4	20.99	18.7	21.5	22	18.97	21.77
91750	9.5	7.2	7.2	18.5	12	12.9	18.81	12.22	13.4
92350	6.6	4.8	4.9	12.89	8.1	8.7	13.2	8.52	9.12
93140	5.6	4.1	4.3	12.5	7.5	8	12.9	7.8	8.3
93300	6.3	5.1	5.4	14	11	11.7	14.7	11.4	12.44
98550	7.97	6.3	7.5	16.09	12.3	14	16.55	12.82	14.62

Table 1: Annual 95, 99 and 99.1 percentiles for precipitations (obs=observations, ctr=control period, sc=scenario)

station	95 percentile			99 percentile			99.1 percentile		
	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
12680	7.9	4.8	5.1	14.0	7.8	8.9	15.0	8.0	8.9
13670	6.4	4.6	4.8	14.3	7.9	8.6	15.4	8.1	8.6
16610	4.8	3.0	2.8	8.0	4.8	4.4	8.2	4.8	4.4
17150	9.8	9.4	9.0	19.2	16.1	16.8	20.1	16.2	17.2
18700	8.8	7.8	7.9	16.6	13.1	13.8	16.9	13.6	14.0
24880	5.0	4.7	4.6	12.0	9.3	8.8	12.6	9.5	9.5
25590	7.8	6.6	6.8	16.4	10.70	10.40	17.0	10.8	10.8
28800	9.0	3.9	8.4	17.3	6.6	15.7	17.6	6.6	15.9
31620	8.0	7.7	7.9	14.4	12.8	12.3	14.7	13.4	12.5
37230	10.5	10.4	10.7	21.7	20.6	21.1	22.5	21.7	21.5
39040	13.3	14	13.6	28.5	26.5	24.2	29.0	26.8	24.8
42920	17.8	14.7	15.3	31.1	24.0	24.0	31.9	25.3	24.8
44560	11.7	11	11.8	18.4	18.3	19.5	18.6	18.7	19.5
46610	20.31	20.1	21.1	34.9	31.3	31.2	35.9	32.0	31.9
50300	28.6	31.3	34	52.2	51.7	50.7	54.5	52.4	51.2
50540	21.7	22.9	25.5	41.6	40.9	37.1	42.4	42.8	37.7
51590	12.8	10.7	11.4	23.1	16.7	16.3	24.5	16.9	16.3
60500	11.6	9	8.9	26.5	13.7	13.4	27.1	13.7	13.4
60990	12.3	12.3	12.4	20.8	21.1	18.4	21.7	21.7	18.7
65110	13.4	13	12.6	22.87	22.5	19.7	23	22.6	20.1
66730	8	5.8	5.7	18.7	8.1	9	19.3	8.1	9
68340	8.5	7.2	6.8	15.9	10.7	10.6	17.2	11.1	10.7
69100	7.7	7.5	7.2	17.7	12.6	10.9	18	13.0	11.0
77750	8.3	7.0	7	19.7	9.9	11.1	20.6	10.1	11.2
80700	18.3	16.3	16.2	33.1	24.0	26.4	33.9	24.5	27.2
82290	9.4	7.5	7.6	18.1	11.2	12.7	18.4	11.8	12.9
86500	13.6	10.8	11.9	26.5	20.1	22.6	26.6	21.6	23.1
89350	5.8	3.8	3.8	11.3	5.8	5.7	11.5	5.8	6.0
91750	6.1	4.1	4.3	11.8	5.8	6.4	11.9	6.1	6.5
92350	4.3	2.8	2.9	8.3	4.4	4.5	8.5	4.5	4.5
93140	3.6	2.5	2.6	7.6	4	4.4	8	4	4.4
93300	4.5	3.3	3.3	9.0	5.4	5.9	9.2	5.5	6.2
98550	5.6	4.2	4.6	10.2	7.1	8.5	10.4	7.4	8.5

Table 19: 95, 99 and 99.1 percentiles for precipitations in Spring (obs=observations, ctr=control period, sc=scenario)

station	95 percentile			99 percentile			99.1 percentile		
	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
12680	12.5	11.6	12.1	23.2	18.6	20.8	23.6	19.2	21.5
13670	8.5	7.5	7.8	17.7	11.2	13.0	18.6	11.3	13.3
16610	6.7	4.5	4.9	13.8	7	7.8	13.9	7.0	8
17150	15.3	16.2	16.7	27.5	26.6	29.6	27.8	26.6	29.9
18700	13.8	14.7	15.1	26.1	22.1	24.2	26.8	22.5	25.3
24880	9.3	8.6	8.5	17.7	13.6	14.4	18.4	13.6	15.1
25590	11.7	10.4	11.1	18.6	15.0	18.8	19.6	15.9	19
28800	14.8	7.4	15.0	26.5	11.4	22.6	27.2	11.5	23.7
31620	12.2	13.3	14.1	20.2	20.0	22.7	20.8	20.1	23.3
37230	19.6	19.8	21.1	35.1	31.3	36.9	36.0	31.4	37.5
39040	23.7	26.205	29.0	39.9	41.1	43.9	40.9	42.0	44.9
42920	30	32.6	35.7	46.7	49.7	59.4	47.0	51.0	59.8
44560	18.2	20.8	26.2	28.7	35.2	40.4	29.8	35.5	43.2
46610	35.4	33.2	41.2	54.1	50.8	62.5	56.4	51.0	65.6
50300	46.8	47.4	61.1	72.4	69.4	89.6	73.8	70.4	92.0
50540	32.7	32.8	40.9	56.7	49.3	63.1	57.9	50.4	64.2
51590	20.6	17.8	23.2	32.7	25.8	32.3	33.4	26.2	33.2
60500	17.9	12.1	13.7	31.2	17.9	19.8	33.6	18.2	20.2
60990	18.6	19.6	23.9	34.6	33.9	38.1	36.1	33.9	38.7
65110	22.9	19.5	22.7	40.6	37.1	39.0	41.4	37.6	40.0
66730	10.9	7.7	8.6	20.4	11.7	12.3	21.2	12	12.4
68340	12.4	9.2	10.8	18.9	14.6	17.8	20.6	15.1	18.1
69100	12.3	9.0	10.6	21.2	14.4	17.1	22.4	14.6	17.9
77750	15.6	11.6	13.1	40.5	17.1	19.2	41.8	17.2	19.5
80700	31.7	29.2	29.4	63.5	51.0	48.9	65.5	52.0	49.9
82290	15.5	12.7	13.6	31.3	24.1	23.1	32.5	24.4	23.5
86500	21.2	15.3	17.9	35.4	29.2	33.9	36.6	29.5	34.3
89350	10.1	7.8	8.1	18.5	12.9	14.6	18.7	13.0	15.1
90450	15.7	13.2	13.5	24.4	21.6	24.8	25.0	22.5	25.8
91750	11.8	8.3	8.3	21.0	13.5	14.4	21.5	13.8	14.7
92350	7.8	5.5	5.6	13	8.1	9.0	13.2	8.1	9.1
93140	6.5	4.4	5.3	12.7	7.5	8.0	13.1	7.5	8.1
93300	6.6	5.2	5.8	13.3	9.9	9.7	14.1	10.0	10.0
98550	9.1	6.8	8.4	15.4	11.2	12.7	16.4	11.2	13.2

Table 20: 95, 99 and 99.1 percentiles for precipitations in Autumn (obs=observations, ctr=control period, sc=scenario)

station	95 percentile			99 percentile			99.1 percentile		
	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
12680	8.1	6.8	7.8	14.4	10.7	12.0	14.9	10.8	12.5
13670	5.7	4.7	5.1	11.6	7.0	7.8	12	7.1	7.8
16610	5.1	3.6	3.7	7.6	5.3	5.2	8	5.3	5.3
17150	11.1	10.1	11.5	20.0	16.1	20.9	20.2	16.1	20.9
18700	8.6	8	9.6	15.6	12.8	14.9	16.0	12.9	15
24880	4.9	4.6	5.1	8.4	7.4	7.8	8.6	7.4	8.3
25590	9.0	8.4	8.8	15.8	12.2	13.2	16.3	12.6	13.5
28800	8.5	4.0	9.1	14.0	6.9	13.4	14.3	6.9	13.6
31620	9.8	11	12.4	18.3	17.1	20.7	18.8	17.3	20.9
37230	11.9	11.1	12.5	23.8	18	19.3	24.5	18.3	20.7
39040	19	17.3	18.8	32.0	28.2	30.9	33.5	28.9	33.9
42920	28.1	28.2	29.9	46.4	43.6	47.2	46.5	45.9	49.1
44560	15.8	14.6	16.1	25	24.8	27.1	25.5	24.8	28.3
46610	34.2	33.3	34.8	54.3	47.5	52.9	55.5	50.3	53.3
50300	45.3	43.4	47.3	66.7	68.5	74.4	68.5	68.7	78.2
50540	30.7	27.8	32.3	51.9	45.6	55.0	52.9	46.7	57.1
51590	20.6	18	18.9	36.2	26.5	27.4	36.3	26.6	28.8
60500	19.8	14.9	15.2	30.1	19.9	20.9	31.3	20.5	21.5
60990	16.3	18.1	19.2	32.1	28.3	33.1	33.5	29.4	34.3
65110	21.7	20.3	22.7	37.8	32.5	35.7	39.2	33.9	36.2
66730	11	9.1	10.6	20.6	14.2	15.3	20.7	14.4	15.5
68340	9.5	7.7	8.3	16.6	11.5	11.6	17.0	11.6	11.7
69100	9.8	9.3	9.6	18.6	13.9	15.2	19.1	13.9	15.2
77750	17.9	13.0	13.6	42.6	20.9	22.0	44.9	21.3	22.4
80700	25.6	25.5	26.1	53.0	47.2	45.3	54.2	48.0	45.6
82290	11.9	10.6	10.6	22.9	18.4	19.8	23.2	19.7	20.2
86500	20.6	19.5	20.4	36.9	38.7	40.8	37.9	39.1	41.4
89350	9.8	7.9	8.0	17.4	11.4	14.3	17.8	11.8	14.3
90450	14.8	12.7	13.1	22.4	21.0	25.0	22.5	21.5	25.3
91750	10.8	7.5	7.2	18.6	11.6	12	19.7	11.6	12.3
92350	6.9	4.6	4.4	12.3	7	7.0	13.0	7	7.1
93140	5.5	4.2	3.7	11.1	6.2	5.6	12.2	6.5	5.9
93300	5	3.6	3.5	9.3	6.2	5.5	9.3	6.4	5.6
98550	7.8	5.3	6.2	14.9	8.3	8.7	15.2	8.5	9.1

Table 21: 95, 99 and 99.1 percentiles for precipitations in Winter (obs=observations, ctr=control period, sc=scenario)

station	95 percentile			99 percentile			99.1 percentile		
	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
12680	13.2	13.3	13.1	24.5	24.5	23.3	25.3	25.0	24.2
13670	12.1	12.5	12.4	23.2	23.4	19.5	23.5	23.9	20.6
16610	10.6	8.9	9.1	20.3	13.9	16.1	20.7	15.3	17.1
17150	14.6	21.4	19.6	25.3	39.2	37.6	25.6	41.5	39.0
18700	13.5	19.7	18.0	27.1	33.7	33.5	29.4	34.0	33.6
24880	11	12.5	12.9	22.3	24.9	23.5	22.7	25.8	23.8
25590	12	12.0	12.3	21.1	19.6	20.4	21.4	20.2	22.4
28800	14.5	11.4	17.3	29.0	20.7	34.1	29.3	22.7	35.2
31620	13	14.6	14.9	22.9	23.8	27.6	23.5	24.6	28.0
37230	15.0	20.3	20.0	29.5	32.4	42.3	31.2	34.1	43.2
39040	17.2	24.1	24.6	34.9	44.0	47.5	35.4	46.5	47.7
42920	17.1	19.9	22.1	30.5	34.2	40.3	31.9	34.6	42.8
44560	14.8	15.5	19.2	26.4	25.7	32.5	27.1	26.7	33.1
46610	21.1	22.0	26.3	38.3	32.3	41.3	39.6	32.8	41.8
50540	24	23.5	30	39.6	36.8	45.1	40.6	37.0	45.3
51590	12.0	12.1	14.5	23.3	17.9	21.5	23.8	18.0	22.2
60500	9.1	7.9	8.9	16.6	14.9	15.8	17.1	15.3	16.1
60990	12.6	12.7	15.4	23.3	22.9	27.3	23.8	23.5	27.6
65110	15.3	14.7	17.1	27.2	26.2	28.2	27.4	26.8	28.7
66730	13.9	11.4	13.1	26.5	21.5	23.8	27.4	21.9	23.9
68340	15.5	14.7	17.1	24.9	22.8	25.9	25.3	23.5	26.6
69100	12.9	13.3	14.1	24.3	21.0	22.8	24.8	21.2	24.0
77750	10.4	9.0	9.6	19.2	13.9	15.4	19.6	14.0	15.7
80700	19.4	20.2	19.4	41.2	33.6	35.6	42.2	34.0	36.2
82290	11.6	10.8	11.6	23.9	19.7	20.5	24.4	20.0	21.1
86500	10.7	8.3	8.3	19.3	15.7	14.8	19.6	15.9	15.6
89350	8.9	7.6	7.6	16.4	12.1	13.6	16.8	12.9	14.3
90450	11.3	10.0	10.2	20.2	17.8	18.3	20.6	18.3	18.6
91750	9.1	8.1	8.1	18.8	14	15.2	19.7	14.2	16.0
92350	7.2	6	6.4	15.1	13.0	12.7	15.4	13.3	13.0
93140	6.3	5.3	5.5	16.9	11.0	13.1	17.2	11.3	13.4
93300	8.8	8.9	9.4	19.7	18.8	19.5	21.3	20.0	21.1
98550	10	9.4	10.3	23.2	17.9	21.8	23.4	18.3	22.8

Table 22: 95, 99 and 99.1 percentiles for precipitations in Summer (obs=observations, ctr=control period, sc=scenario)

station	0.1			1			5			95			99		
	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
12680	-29.09	-20.01	-18.39	-18.15	-12.8	-11.1	-11.6	-8.7	-7.6	16.8	14.8	14.8	19.9	16.1	15.9
13670	-27.97	-21.1	-21.48	-16.7	-13.8	-12.7	-12.3	-9.8	-7.7	13.2	11	14.4	16.6	12.4	15.4
16610	-28.83	-18.76	-20.78	-17.3	-13.7	-13.2	-12.5	-9.9	-9.1	12.3	9.7	9.9	15.7	11.3	11.1
17150	-23.52	-19.67	-17.74	-15	-10.6	-8	-7.9	-5.4	-3.9	17.8	15.5	15.6	20.6	17	16.7
18700	-20.9	-17.45	-14.28	-13.5	-9.7	-7.5	-7.5	-5.4	-4.3	18.58	16.1	16.2	21.59	17.3	17.2
24880	-33.55	-30.22	-25.27	-22.2	-15.8	-14.7	-14.8	-11	-9.7	17.2	15.2	15.3	20.29	16.5	16.4
25590	-31.66	-21.61	-23.14	-18.3	-14.5	-12.3	-12.97	-9.4	-8.3	13	11.1	11.2	16.1	12.4	12.4
28800	-30.2	-24.47	-21.16	-19.2	-16.1	-11.8	-11.97	-11.3	-7.5	16.4	11.1	14.3	19.29	12.4	15.3
31620	-28.44	-19.87	-19.1	-17.1	-13.5	-11.6	-11.7	-9.41	-8.3	12.6	10.4	10.6	15.7	11.9	11.7
37230	-24.73	-19.96	-16.96	-13.99	-10.9	-8.5	-7.88	-6.1	-4.8	17	14.9	15	19.8	16.3	16.1
39040	-23.17	-11.5	-12.08	-11	-6	-4.4	-4.7	-1.8	-0.5	17.5	14.7	14.9	20.1	16.1	15.9
42920	-29.19	-17.47	-17.99	-17.5	-12	-10.4	-9.9	-7.5	-6.1	14.7	11.9	12.1	17.5	13.4	13.1
44560	-15.42	-9.16	-8.68	-6.8	-4.1	-3	-2	-0.7	0.3	16.3	14.1	14.3	19.8	15.3	15.3
46610	-20.45	-11.8	-11.91	-10.8	-7.5	-6.2	-5.3	-3.9	-2.51	17	14.6	14.7	20	15.8	15.8
50300	-20.25	-13.18	-13.08	-11.5	-8.4	-7.4	-6.4	-4.61	-3.5	14.6	12.1	12.2	17.7	13.3	13.4
50540	-12.46	-7.85	-7.67	-4.8	-3.6	-2.5	-1.3	0	1.1	16.7	15.4	15.7	20.1	17.3	17.1
51590	-29.26	-17.13	-17.51	-17.09	-11.4	-10	-8.97	-6.4	-5.1	16.5	14	14.2	19.6	15.3	15.2
60500	-12.82	-8.02	-6.88	-7.06	-3.7	-2.8	-3	-0.6	0.1	16.1	14	14.2	19.2	14.9	15
60990	-9.05	-3.87	-3.98	-3	-0.7	-0.1	-0.7	1.7	2.4	14.67	13.6	14	17.4	14.3	14.6
65110	-18.2	-11.17	-10.66	-10.4	-5.9	-5.1	-5.4	-2.8	-2	15.6	13.4	13.7	18.8	14.9	14.7
66730	-23.18	-17.88	-17.98	-14.8	-10.1	-9.3	-9.8	-6.3	-5.41	14.3	11.5	11.7	17.1	13.1	12.9
68340	-23.16	-15.57	-16.64	-14.5	-9	-8	-7.83	-4.9	-4	16.1	13.4	13.6	19.4	15.3	14.9
69100	-22.08	-13.38	-13.89	-13.6	-7.9	-7	-6.9	-3.8	-2.8	16.7	14.7	14.8	20.2	16.5	16.1
77750	-31.84	-25.21	-23.48	-24.5	-17.1	-15.9	-15.8	-12.01	-10.9	15.1	13	13.2	18.56	14.8	14.3
80700	-13.51	-8.8	-7.52	-9.1	-5.6	-4.5	-5.1	-3.3	-2.4	15.2	13.2	13.5	18.69	14.8	14.6
82290	-15.09	-10.91	-9.28	-9.9	-7.1	-5.3	-5.8	-4.1	-2.6	14.7	13.6	14.2	18.6	15	15.4
86500	-15.13	-9.68	-7.83	-9	-7.1	-5.4	-5.31	-4.5	-3.2	14	12.7	13.3	17.4	14.5	14.7
89350	-32.54	-22.23	-21.42	-24.5	-17.3	-14.6	-17.18	-13.2	-11.1	15.3	13.6	14	18.89	15.3	15.5
90450	-15.73	-11.22	-8.98	-11.4	-8.4	-6.7	-7.9	-6.2	-4.5	13.7	13	13.5	17.7	14.9	15.4
91750	-28.35	-18.96	-16.96	-19.3	-13.9	-11.9	-14.1	-10.5	-8.8	14.7	13.2	13.71	18.5	14.9	15.4
92350	-18.43	-13.18	-12.07	-12.1	-9.4	-7.9	-8.4	-6.8	-5.5	13.8	12.6	13.1	17	13.9	14.4
93140	-29.2	-19.67	-17.85	-20	-14.9	-13	-14.1	-11	-9.4	15	13.7	14.2	19	15.2	15.8
93300	-39.41	-27.03	-24.48	-29.5	-22	-19.7	-21.77	-16.9	-14.9	13.4	11.5	11.9	17.29	13	13.5
98550	-18.69	-15.1	-11.53	-11.3	-9.9	-7.3	-8	-7.3	-4.9	10.6	12.5	13.8	13.3	14.2	15.4

Table 23: Annual 0.1, 1, 5, 95 and 99 percentiles for temperature (obs=observations, ctr=control period, sc=scenario)

station	0.1			1			5			95			99		
	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
12680	-18.1	-9.4	-10.0	-11.5	-6.6	-6.7	-6.3	-3.7	-3.5	13.5	10.1	10.7	16.4	12	1 2.1
13670	-17.8	-13.6	-14.3	-12.2	-9	-10.4	-9.1	-6.8	-5.9	9.8	6.1	9.8	12.7	8.4	11.4
16610	-18.4	-14.7	-12.6	-13.1	-10.2	-10.7	-10.1	-7.6	-8	8.7	4.9	6.1	11.8	6.9	8
17150	-16.2	-9.5	-9.0	-7.8	-5.4	-5	-3	-1.7	-1.4	14.3	11.4	12.1	18.0	13.2	14.1
18700	-14.9	-7.7	-7.6	-6.6	-4.8	-4.5	-2.8	-1.7	-1.3	15.3	11.9	12.5	18.7	14	14.2
24880	-18.8	-11.6	-10.4	-11.9	-6.9	-6.9	-6.4	-3.8	-3.5	13.3	10.2	10.9	16.0	12.2	12.7
25590	-19.9	-13.6	-13.5	-14.1	-9.7	-10.2	-9.7	-7	-6.8	8.8	5.8	6.9	11.7	8	8.6
28800	-18.5	-14.3	-11.4	-10.8	-9.9	-7.1	-5.9	-6.7	-3.7	12.9	6.3	10.5	15.92	8.4	12.2
31620	-17.2	-12.3	-12.9	-12.5	-9.7	-10.2	-9.6	-7.2	-7.1	8.1	5.1	6	11.4	7.1	7.8
37230	-15.9	-10.6	-10.8	-8.9	-6.4	-6.2	-4.5	-2.9	-2.7	13.1	10.2	10.7	16.1	12.3	13
39040	-10.3	-6.8	-6.4	-4.8	-2.4	-2.3	-1.5	0.7	0.8	13.9	11	11.7	16.7	12.7	14.4
42920	-14.1	-10.2	-10.6	-10.2	-6.9	-6.7	-6.3	-3.8	-3.9	10.8	7.6	8.2	14.29	9.8	10.2
44560	-4.8	-3.9	-3.9	-2.2	-0.3	-0.3	0.4	1.8	2	13.5	10.7	11.2	17.5	12.8	13.2
46610	-7.4	-4.5	-4.3	-4.4	-2.3	-1.9	-1.1	0.1	0	13.9	11	12	17.0	12.8	14
50300	-9.3	-8.5	-7.6	-7.3	-5.3	-5	-4.2	-2.4	-2.1	11	8.5	9.3	14.9	10.6	11
50540	-4.1	-4.76	-3.9	-1.4	-0.6	-0.7	0.8	2	2.1	14.3	12.8	13.8	17.7	15.3	15.8
51590	-10.5	-7.1	-7.0	-7.2	-4.5	-4.4	-2.9	-1.7	-1.6	13.2	9.9	11.3	15.8	11.7	13
60500	-7.6	-3.3	-4.1	-3.82	-1.4	-1	-0.8	1.3	1.2	13.6	10.4	11.6	16.8	12	12.9
60990	-4.4	-3.0	-1.7	-1.6	0	0.3	0	2.3	2.5	11.2	9.8	10.6	14.8	11.1	11.5
65110	-8.6	-6.8	-5.2	-5.9	-4	-3.6	-2.6	-1.2	-1.1	12	9.6	10.5	16.4	11.7	12.2
66730	-14.2	-7.1	-6.4	-9.8	-3.3	-3.5	-6.3	-1.6	-0.9	10.5	7.6	8.9	14.7	9.7	10.2
68340	-12.8	-9.0	-8.1	-8.1	-5.8	-5.5	-4.4	-2.2	-2.4	12.4	9.9	10.8	17.0	12	12.4
69100	-12	-7.2	-6.3	-6.4	-3.9	-3.4	-2.4	-0.4	-0.6	13.2	11.1	11.8	17.8	13.1	13.4
77750	-21.9	-14.2	-14.2	-14.96	-11.2	-10.1	-9.4	-8.2	-7.1	10.1	7.8	9.7	14.6	11.6	11.9
80700	-9.9	-6.8	-5.6	-6.2	-4.6	-3.3	-3.4	-2.3	-1.3	11.7	9.6	10.9	15.0	12.8	12.9
82290	-10.2	-8.0	-7.5	-7.2	-5.8	-4.3	-4.1	-3	-1.7	10.8	10.2	11.9	14.7	12.6	13.8
86500	-9.7	-6.6	-5.3	-6.22	-4.9	-3.2	-4.1	-2.9	-1.8	8.6	8.1	9.1	12.1	10.9	11.1
89350	-20.8	-16.5	-12.1	-16.6	-12.6	-10.2	-10.3	-9.5	-7.7	9.2	8.1	9.7	12.9	11.8	12.4
90450	-12.1	-9.7	-6.6	-9.0	-7.4	-5.2	-6.3	-4.8	-3.5	8	7.7	8.9	11.5	10.8	11.3
91750	-21.8	-15.4	-12.7	-14.4	-10.9	-9	-9.3	-7.4	-6.5	8.6	7.7	9	12.0	11.2	11.9
92350	-15.0	-12.5	-9.1	-9.6	-8.4	-6.6	-6.7	-5.3	-4.5	7.9	7.2	8.4	10.5	9.9	10.7
93140	-20.3	-18.1	-12.5	-15.2	-12.6	-10.2	-9.7	-8.6	-7.6	7.6	7.5	8.6	9.9	10.1	11.1
93300	-29.4	-24.6	-20.4	-23.1	-19	-16	-15.6	-13.9	-12.6	5	4.9	6	7.9	7.7	8.3
98550	-14.1	-12.2	-8.5	-8.7	-7.9	-5.7	-6.3	-5.7	-3.9	5.3	4.5	6.7	7.6	6.4	8.2

Table 24: 0.1, 1, 5, 95 and 99 percentiles for temperature in Spring (obs=observations, ctr=control period, sc=scenario)

station	0.1			1			5			95			99		
	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
12680	5.9	9.6	9.9	7.7	10.9	11.4	9.2	11.6	12	19.6	15.9	15.7	21.5	16.7	17.2
13670	1.79	5.8	9.52	3.34	7	10.4	5.4	7.6	11	16.2	12.3	15.3	18.36	13.3	16.2
16610	0.2	6.02	6.3	2.2	6.3	6.4	4	6.5	6.5	15.3	11.1	11	17.3	12.4	12.2
17150	5.29	10.92	11.6	9.5	11.7	12.1	10.9	12.4	12.8	20.3	16.8	16.6	22	18	17.8
18700	5.87	11.72	12.12	9.5	12.4	12.8	11.1	13.1	13.4	21.2	17.3	17	22.7	18.2	18
24880	5.98	10.25	10.9	8.34	11.1	11.8	9.8	11.9	12.3	19.8	16.3	16.2	22.2	17.2	17.4
25590	2.41	6.34	6.75	3.94	7.2	7.5	5.9	7.8	8.2	15.8	12.3	12.1	17.96	13.5	13.4
28800	4.55	6.2	9.94	7.4	7.4	10.9	9	7.9	11.5	19	12.3	15.2	20.76	13.1	16.2
31620	1.53	5.72	6	3.5	6.3	6.7	5.4	7.1	7.5	15.4	11.7	11.5	17.24	12.8	12.7
37230	5.74	10.25	10.62	8.34	11	11.4	10	11.7	12	19.5	16.2	16	21	17.3	16.9
39040	6.57	10	9.93	9.4	10.6	10.9	11	11.4	11.8	19.9	15.9	15.8	21.6	16.8	16.9
42920	3.68	1.14	1.04	6.2	1.7	1.9	7.4	2.29	2.4	17.3	13.2	12.9	19.1	14.5	14.2
44560	7.2	10.7	10.64	8.8	10.9	11.1	10.1	11.3	11.5	19.1	15	15	21.7	16.2	16.2
46610	6.01	11.32	11.4	8.74	11.7	11.9	10.3	12.2	12.5	19.5	15.6	15.5	21.64	17.2	16.8
50300	3.4	8.14	8.37	5.6	8.9	9.1	7.4	9.5	9.8	17.3	13.2	13.2	19.39	15	14.5
50540	6.62	10.12	10.62	8.74	10.9	11.4	10.3	11.8	12.3	19.7	17.1	16.8	22	18.5	18.5
51590	5.47	10.5	10.5	7.5	11	11.3	9.4	11.5	12	19.2	15.1	15.1	21.4	16.5	16.2
60500	5.46	10.72	11	8.08	11.3	11.4	9.5	11.6	11.8	18.4	14.8	14.8	21.2	15.6	15.4
60990	5.52	7.3	8.02	7.24	8.7	9.2	8.4	9.6	9.9	16.8	14.2	14.4	19.3	15	14.9
65110	4.35	8.4	8.85	6.53	9.4	9.7	8.2	10.2	10.4	18.3	14.7	14.5	20.88	15.9	15.4
66730	2.02	6.22	6.75	3.83	7.4	7.6	5.9	8.3	8.4	16.9	12.9	12.8	18.7	14.2	13.8
68340	2.58	7.42	7.94	5.1	8.6	8.9	7	9.6	9.9	19.1	15.1	14.7	21.19	16.6	15.7
69100	4.87	9.22	9.74	7.3	10.3	10.5	8.9	11.2	11.4	19.6	16.3	15.9	22.48	17.7	16.9
77750	3.42	7.3	8.14	4.9	8.3	8.8	6.4	9.2	9.9	17.9	14.5	14.2	20.2	16.1	15.1
80700	3.36	7	8.02	5.04	7.9	8.6	6.9	9.1	9.6	18.3	14.6	14.5	20.76	16	15.4
82290	4.22	5.6	7.45	5.5	7.4	8.5	7.1	9	10.1	18	14.9	15.3	20.56	16	16.2
86500	3.11	6.62	7.42	5.1	7.7	8.2	6.8	8 .6	9	17	14.2	14.5	20.06	15.6	15.8
89350	1.72	6.14	7.2	3.54	7.4	8.1	5.8	8.8	9.6	18.5	15.1	15.3	20.9	16.6	16.8
90450	0.92	5.02	6.1	2.98	6.2	7.3	5.2	7.8	8.6	17.3	14.6	15.2	19.5	16.3	16.7
91750	1.02	6	6.72	3.31	6.9	7.6	5.8	8.2	8.9	18.06	14.7	15.2	21.09	16.4	16.5
92350	1.32	5.7	6.42	3.6	6.5	7.2	5.8	7.8	8.4	16.6	13.7	14.2	19.12	15.1	15.4
93140	1.44	6.02	6.72	3.64	6.9	7.8	6	8.4	8.9	18.7	15	15.6	21.6	16.6	17
93300	-1.5	3.24	4.15	1	4.7	5.6	3.4	6.4	7	16.8	12.8	13.3	19.69	14.4	14.6
98550	-0.11	1.29	2.82	1.84	3.4	5.7	3.6	4.9	7.3	13	13.9	15.2	15.9	14.9	16.3

Table 25: 0.1, 1, 5, 95 and 99 percentiles for temperature in Summer (obs=observations, ctr=control period, sc=scenario)

station	0.1			1			5			95			99		
	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
12680	-16.1	-11.7	-10.0	-10.7	-7.5	-5.3	-7.4	-4.4	-2.8	11.7	10.2	11	14.5	11.8	12.7
13670	-17.7	-15.6	-12.3	-13.5	-9.6	-6.1	-9.6	-7.1	-2.9	8.8	7	10.4	11.2	8.4	12.1
16610	-19.2	-16.4	-14.3	-13.8	-9.8	-9.6	-9.7	-6.9	-6.4	8.4	4.9	5.4	11.1	5.6	6.1
17150	-13.5	-7.6	-5.1	-7.2	-3.9	-1.6	-2.8	-0.8	0.3	13.9	12.1	13	16.0	13.6	14.4
18700	-10.4	-6.3	-4.9	-6.4	-3.1	-1.5	-2.9	-0.9	0.1	13.8	12.1	12.9	16.08	13.6	14.3
24880	-20.4	-16.6	-15.0	-16.0	-9.5	-7.8	-10.1	-6.3	-5.4	12.4	10	10.9	14.6	11.2	12.3
25590	-20.0	-15.9	-15.9	-14.2	-10.1	-8.9	-9.4	-6.3	-5.3	9	7.4	8.4	11.5	8.8	9.7
28800	-18.1	-17.1	-7.9	-12.4	-11.4	-5	-7.3	-7.4	-3.2	11.8	7.7	10.6	14.3	8.9	12
31620	-17.6	-13.1	-11.1	-12.6	-8.3	-6.8	-7.7	-5.7	-4.7	8.7	7	7.8	10.9	8.5	9.1
37230	-11.5	-6.5	-4.5	-6.4	-3.4	-1.7	-2.7	-1.2	-0.1	12.9	11.2	11.8	14.8	12.6	13.2
39040	-11.2	-3.9	-2.1	-5.0	-0.8	0.6	-0.4	1.7	2.8	14.2	12.3	13.1	15.8	13.2	14.2
42920	-14.3	-8.6	-6.4	-10.6	-5.4	-3.9	-4.5	-2.6	-1.6	11.3	9.3	9.9	13.09	10.8	11.5
44560	-7.8	-2.1	0.9	-2.3	0.3	1.7	1	2.8	3.8	14.5	13	13.5	16.8	14.2	15.5
46610	-10.6	-5.0	-3.2	-5.9	-2.7	-1.1	-1.9	0	1	13.7	11.8	12.4	16.4	13.1	14
50300	-11.5	-7.4	-4.1	-6.9	-3.5	-2	-3.2	-1.1	0	11.8	9.9	10.6	14.1	11.5	12.3
50540	-6.9	-3.71	-0.47	-1.9	0.5	2	0.9	2.8	3.9	14.5	13.2	13.9	16.9	14.4	15.3
51590	-16.3	-10.42	-6.8	-10.4	-5.5	-3.4	-5.3	-2.1	-0.9	13.1	10.7	11.4	15.4	12.1	12.8
60500	-8.2	-3.8	-2.6	-4.0	0.1	0.6	-1	2.3	3	15.1	12.5	13.4	18.0	13.9	14.9
60990	-4.2	0.6	1.9	-0.8	2.8	3.4	1.6	4.3	4.9	13.8	13.1	13.8	16.2	13.9	14.8
65110	-12.3	-5.7	-4.0	-7.0	-2.8	-1.8	-3	0	0.5	12.8	11.2	11.9	16.0	12.3	13.3
66730	-16.2	-13.7	-8.8	-11.5	-6.7	-6	-7.3	-3.7	-2.9	11.2	9	9.9	13.5	10.5	11.9
68340	-19.5	-8.7	-7.2	-9.3	-4.4	-3.5	-4.8	-1.7	-0.7	12.9	10.9	11.8	16.1	12.3	13.1
69100	-14.8	-6.8	-5	-8.1	-3.2	-2.3	-3.8	-0.3	0.4	14	12.2	13	17.4	13.6	14.2
77750	-28.5	-17.8	-15.6	-18.0	-12.7	-11.1	-11.2	-7.8	-6.5	10.5	8.7	9.6	13.2	9.9	10.8
80700	-8.7	-5.18	-4.18	-5.5	-2.6	-1.4	-2.3	-0.5	0.4	12.9	11.1	12.1	15.8	12.1	13.3
82290	-10.0	-6.2	-3.2	-5.9	-3.4	-1.4	-2.8	-0.8	0.6	12.4	11.6	12.6	15.4	12.5	13.5
86500	-8.7	-4.8	-2.8	-5.2	-3.4	-1.8	-2.5	-1.3	-0.2	11.8	11.1	12.1	13.8	12.2	13.3
89350	-26.1	-14.6	-13.3	-19.6	-12.5	-9.4	-12.9	-9.3	-7.2	10.3	8.8	9.8	13	10	11.1
90450	-11.1	-7.2	-4.8	-8.1	-5.5	-3.5	-5.3	-3.2	-1.6	10.2	9.7	10.7	12.9	10.9	11.7
91750	-21.5	-13.2	-10.9	-15.3	-9.7	-7.9	-10.9	-7.1	-5	10.9	9.2	10.2	13.9	10.4	11.2
92350	-11.8	-8.2	-7.3	-8.2	-5.7	-4.2	-5.4	-3.4	-1.8	10.8	9.7	10.5	13.2	10.8	11.4
93140	-21.2	-14.3	-12.2	-15.4	-9.5	-7.7	-10.3	-7	-5	10.9	9.8	10.8	14.26	11.3	11.9
93300	-34.59	-23.4	-19.7	-24.4	-16.2	-14	-17.4	-12.6	-10.2	8.4	7.1	8	11.54	8.5	9.2
98550	-10.4	-8.6	-6.67	-7.4	-6.4	-3	-4.7	-4.1	-0.8	9.3	10.8	12.4	11.4	12.3	13.7

Table 26: 0.1, 1, 5, 95 and 99 percentiles for temperature in Autumn (obs=observations, ctr=control period, sc=scenario)

station	0.1			1			5			95			99		
	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
12680	-30.0	-20.2	-19.1	-22.6	-15.6	-14.3	-16.8	-12.3	-10.8	2.3	-1.3	-0.9	5.1	-0.3	0.4
13670	-29.7	-21.4	-22.0	-20.6	-16.6	-15.8	-16.3	-12.6	-11.3	0.5	-3.3	0.1	3.2	-2.5	1.1
16610	-29.71	-18.86	-20.95	-20.7	-16.2	-16.5	-16.6	-13.11	-12.4	0.2	-3.1	-2.8	2.9	-2.4	-2.2
17150	-23.96	-20.97	-18.38	-18.3	-13.6	-12.9	-13.8	-9.51	-7.11	5.4	4.9	6.01	6.99	6.6	7.4
18700	-21.73	-17.94	-15.19	-17.09	-12.3	-11	-12.7	-9	-7	4.2	2.5	3.3	5.9	4.1	4.7
24880	-33.66	-30.66	-27.92	-26	-20.5	-19.3	-21.38	-15.3	-14.01	3.38	-1.99	-1.4	6.8	-0.1	0.4
25590	-33.42	-23.55	-23.78	-22.39	-17.5	-16.1	-17.48	-13.6	-11.4	2.2	-1.5	-0.9	4.2	-0.5	0
28800	-31.03	-24.79	-21.26	-22.3	-19.4	-16.4	-18.27	-15.2	-11.3	3.9	-2.89	1.7	6.4	-1.6	4.1
31620	-29.16	-21.17	-20.82	-20.61	-15.7	-15.2	-16.58	-12.9	-11	1	-1.8	-1.3	3.01	-0.8	-0.1
37230	-24.78	-20.67	-18.41	-19.1	-13.8	-12.4	-13.28	-10.3	-8	5.78	3.7	4.7	8.2	5.6	6.5
39040	-23.72	-11.87	-12.25	-16	-9	-8	-10.28	-5.6	-3.5	6.9	7.6	8.9	8.4	9.1	9.9
42920	-29.35	-17.79	-18.75	-22.1	-14.4	-13.6	-16.4	-11.5	-9.81	3	2.5	3.5	4.3	4.5	5.2
44560	-15.86	-9.26	-8.85	-10	-6	-5.4	-6.2	-3.61	-2.6	7.3	7.1	7.9	8.5	8.2	8.9
46610	-21.77	-12.17	-12.5	-14.49	-9.2	-8.5	-10.1	-7.11	-5.71	5.6	3.71	4.6	7.5	5	6
50300	-20.74	-13.35	-13.62	-14.3	-10.5	-9.4	-10.8	-7.91	-6.8	4.2	2.4	3.8	5.5	4.4	5.1
50540	-14.22	-8.71	-7.99	-7.5	-5.51	-4.8	-4.3	-3.1	-1.91	7.5	6.3	7.5	8.8	7.7	8.5
51590	-29.59	-17.18	-19.08	-21.1	-13.9	-13.1	-15.95	-10.91	-9.5	4.8	1.9	3.2	6.3	3.8	4.6
60500	-13.86	-8.46	-7.05	-9.6	-5.4	-5.1	-6.5	-3.21	-2.3	8.1	4.3	4.8	10.5	5.1	5.8
60990	-9.16	-4.19	-4.15	-4.3	-2	-1.8	-2.5	-0.3	0.6	7.4	6.7	7.4	8.9	7.7	8.2
65110	-19.17	-11.49	-11.37	-14.13	-7.8	-8	-9.7	-5.5	-4.61	5.07	3	3.6	6.83	4	4.7
66730	-23.8	-18.05	-18.52	-19.04	-12.6	-11.8	-13.8	-9.4	-8.9	2.8	-0.69	-0.3	4.87	0.3	0.6
68340	-23.49	-15.89	-17.89	-19.18	-11.2	-10.9	-13.5	-8.4	-7.51	4.5	1.2	1.6	6	2.3	2.8
69100	-22.85	-13.55	-15.02	-18	-10.6	-9.9	-12.79	-7.5	-6.5	5.7	2.5	3.3	7.5	3.9	4.5
77750	-32.94	-25.43	-25	-27.6	-21	-18.4	-23.23	-16.41	-15.11	3.5	-1.6	-1.2	5.68	-1	0
80700	-13.73	-8.8	-7.96	-11.1	-7	-5.8	-8.5	-5.21	-4.3	6.17	3.1	3.6	7.5	3.9	5.2
82290	-15.47	-11.13	-9.45	-12.3	-8.6	-6.8	-9.47	-6.61	-4.9	5.4	4	4.7	6.7	5	6.2
86500	-15.71	-9.85	-7.88	-11.95	-8.4	-6.7	-8.3	-6.61	-5.2	5	3.01	3.6	6.3	4	5
89350	-33.64	-22.28	-22.6	-27.8	-19.8	-17.1	-23.7	-16.91	-14.4	3.2	-2.1	-1.7	5.2	-1.5	0.3
90450	-16.01	-11.66	-9.15	-13.5	-9.6	-7.8	-10.9	-8.1	-6.4	3.4	0.91	1.6	4.8	1.8	3.3
91750	-29.21	-19.06	-17.67	-22.89	-15.4	-13.6	-18.7	-13.5	-11.5	3.3	-1.3	-0.9	5.8	-0.4	0.7
92350	-18.71	-13.72	-12.39	-14.2	-11.1	-9.4	-11.6	-9	-7.5	3.9	0.6	1	5.6	1.3	2
93140	-30.4	-19.99	-18.71	-23	-16.8	-15	-19.3	-14.4	-12.7	2.9	-1.59	-1.1	5.2	-0.8	0.4
93300	-40.54	-27.08	-25.39	-33.7	-24.9	-22.4	-28.43	-21.11	-19.1	-1.1	-6.4	-5.8	1.68	-5.3	-4.6
98550	-19.07	-15.1	-11.58	-14.67	-11.5	-9.4	-10.67	-9.4	-7	1.7	-1	1	3.3	0	2.3

Table 27: 0.1, 1, 5, 95 and 99 percentiles for temperature in Winter (obs=observations, ctr=control period, sc=scenario)

station	5 years			10 years			25 years			50 years		
	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc	obs	ctr	s
12680	42.52	35.47	35.06	48.5	40.37	40.31	55.19	46.39	46.88	59.61	50.74	51
13670	36.65	34.43	32.28	41.26	39.57	37.01	47.17	45.69	42.6	51.6	49.98	46
16610	29.09	24.17	22.8	33.22	28.09	26.47	38.92	33.24	31.83	43.52	37.2	36
17150	38.96	65.57	64.3	41.21	81.03	76.37	43.39	104.17	92.49	44.63	124.38	10
18700	44.19	52.08	53.64	49.11	67.02	59.65	55.37	94.58	65.82	60.04	123.93	69
24880	35.62	33.55	34.21	43.23	38.41	37.92	55.3	44.45	41.65	66.44	48.85	43
25590	37.2	29.69	32.1	42.09	34.11	34.74	48.1	40.45	37.17	52.43	45.78	38
28800	48.76	31.49	47.96	55.26	36.47	51.72	63.14	42.89	55.03	68.78	47.73	56
31620	33.48	33.27	39.76	36.94	35.34	44.52	40.59	37.23	50.07	42.87	38.24	53
37230	59.42	49.51	64.03	65.19	56.63	73.46	71.31	67.02	85.06	75.14	75.87	93
39040	71.33	66.19	94.78	80.75	71.14	118.07	92.88	75.92	154.68	102.03	78.65	188
42920	68.95	65.46	77.68	76.18	76.57	89.73	84.81	96.1	110.83	90.87	116	132
44560	55.32	50.89	64.04	64.78	56.55	71.99	78.24	63.24	81.17	89.44	67.9	87
46610	92.53	65.19	92.22	95.76	71.39	102.43	97.81	80.25	115.09	98.57	87.66	124
50300	110.28	104.54	124.25	125.92	112.6	140.2	148.45	120.69	162.31	167.44	125.46	180
50540	83.97	86.78	95.19	91.41	98.22	107.47	98.86	112.33	123.45	103.26	122.54	135
51590	53.95	36.08	42.23	57.65	40.74	49.25	61.24	47.98	62.6	63.3	54.54	77
60500	52.29	27.5	29.28	59.29	30.7	32.34	68.3	34.92	36.27	75.11	38.19	39
60990	54.56	57.25	62.06	58.41	67.62	74.39	61.98	82.2	94.82	63.94	94.19	114
65110	69.77	58.53	59.77	79.61	68.96	68.04	90.52	83.54	79.21	97.65	95.44	88
66730	44.53	30.49	33.17	53.51	33.56	37.49	66.64	36.69	43.18	77.85	38.56	47
68340	40.15	31.9	36.9	44.96	36.4	44.57	50.64	42.1	57.36	54.6	46.34	69
69100	36.82	26.98	33.05	41.8	28.98	38.97	48.97	30.93	48.24	55.03	32.04	56
77750	74.56	28.97	31.87	85.55	32.88	35.34	98.02	37.79	39.37	106.37	41.4	42
80700	111.08	85.12	89.64	125.78	98.65	106.06	143.49	114.78	128.07	156.02	126.08	145
82290	51.85	36.06	40.91	58.99	41.59	48.83	67.4	48.3	60.19	73.22	53.07	69
86500	53.13	64.95	63.06	57.88	79.42	72.72	63.45	99.99	84.36	67.3	117.1	92
89350	34.34	21.65	23.38	39.36	24.72	26.43	45.69	28.7	30.26	50.37	31.72	33
90450	33.75	31.71	39.21	36.64	36.88	47.41	39.88	44.54	60.36	42.02	51.17	72
91750	33.44	22.9	24.91	36.06	25.74	28.35	38.56	29.15	33.53	39.97	31.55	38
92350	28.41	19.62	19.62	34.48	23.07	23.28	43.79	27.81	28.66	52.13	31.62	33
93140	23.96	17.39	19.63	26.65	20.77	23.74	29.81	25.58	29.63	31.99	29.57	34
93300	31.68	28.72	31.04	35.69	33.95	40.68	40.39	40.7	57.73	43.62	45.8	75
98550	34.61	27.81	35.54	39.43	37.6	47.55	45.38	57.8	70.06	49.71	81.68	94

Table 28: Precipitations: return values using GEV distribution with annual maximum precipitations (in mm)

station	5 years			10 years			25 years			50 years		
	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
12680	16.03	13.6	13.57	19.48	17.32	16.66	26.38	22.28	22.83	33.28	26.01	27.46
13670	13.69	10.47	9.92	16.72	13.38	12.47	22.79	19.21	17.57	27.34	23.58	22.67
16610	9.98	8.35	8.96	13.57	11.09	11.09	18.36	14.74	15.35	24.34	19.31	19.61
18700	16.8	15.75	16.27	22.02	21.89	22.4	28.97	28.04	30.57	35.93	37.25	36.69
24880	11.11	11.04	10.31	14.84	13.8	13.91	22.32	19.34	18.72	26.06	24.87	24.73
25590	14.38	12.69	13.71	17.72	15.18	16.96	22.74	20.16	21.3	27.75	23.9	25.64
28800	17.71	12.59	16.49	21.55	15.58	21.53	31.15	21.57	28.25	36.92	27.56	34.96
31620	16.44	15.38	15.01	18.88	18.59	18.97	24.98	23.93	25.58	28.64	27.14	30.86
37230	22.01	21.91	21.41	28.06	27.05	26.42	38.15	33.91	36.43	46.23	40.77	46.44
39040	25.97	29.26	29.92	34.58	36.23	35.07	46.06	45.53	50.49	54.66	52.49	60.77
42920	34.86	34.01	34.3	42.12	40.91	41.75	51.81	50.11	56.65	59.08	57.02	67.83
44560	18.79	21.59	23.41	24.18	27.4	30.31	32.26	35.15	39.51	40.35	40.96	46.42
46610	39.2	35.71	36.76	47.4	40.87	47.24	61.07	51.2	61.2	72	58.95	71.67
50300	51.86	49.89	54.89	60.64	56.55	64.89	78.21	69.88	84.88	91.39	83.2	94.87
50540	37.22	36.16	37.98	45.88	42.84	45.56	57.42	56.19	60.74	68.96	66.21	72.12
51590	23.78	19.57	22.32	29.12	23.46	25.4	38.02	28.64	31.55	45.14	32.52	36.16
60500	21.53	15.38	15.68	28.33	18.78	19.27	35.13	22.19	22.86	41.93	24.46	25.25
60990	21.31	19.41	19.71	28.6	25.11	25.72	37.71	33.67	34.73	45	42.23	43.74
65110	24.27	24.92	25.52	30.05	31.41	32.38	41.62	40.06	41.52	53.18	48.72	50.66
66730	16.16	12.8	12.67	22.28	15.98	15.78	28.4	20.23	20.44	34.53	25.53	25.11
68340	16.65	14.41	13.51	19.69	17.08	16.45	27.27	22.42	22.33	31.82	26.42	28.21
69100	16.69	13.22	12.37	20.84	15.88	14.85	26.36	19.43	19.79	31.89	22.98	24.74
77750	19.67	14.25	14.4	28.93	16.59	18	41.29	20.1	22.8	53.64	23.61	26.4
80700	37.12	32.87	31.73	46.07	39.71	40.08	63.99	53.38	52.61	81.9	67.05	65.13
82290	17.97	16.03	15.79	24.36	19.32	18.96	32.88	25.9	25.3	39.27	30.84	31.64
86500	23.58	23.14	22.23	30.9	29.16	30.35	38.22	41.21	41.19	45.55	53.26	52.03
89350	12.79	10.19	11.29	15.74	12.08	13.24	20.15	15.87	17.16	26.05	17.77	21.07
90450	18.19	17.17	16.38	21.78	20.85	22.46	25.37	25.77	28.54	28.97	29.45	36.64
91750	14.63	10.79	11.85	18.12	12.51	13.99	22.76	16.8	18.28	27.41	20.24	22.56
92350	9.12	7.14	7.77	11.87	8.89	9.41	17.36	12.41	12.68	21.48	15.92	15.95
93140	8.46	6.12	7.03	11.16	7.55	8.05	15.65	11.14	12.13	20.14	14	15.19
93300	10.04	8.55	8.39	12.65	11.04	10.18	17.89	16.04	15.56	23.13	21.03	22.74
98550	11.55	9.98	9.98	15.56	9.98	12.25	20.91	14.53	16.81	26.26	19.08	21.36

Table 29: Precipitations: return values using GPD distribution with annual maximum precipitation in mm (the threshold is the 90th percentile)

station	5 years			10 years			25 years			50 years		
	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
12680	-22.33	-17.57	-15.08	-26.09	-18.89	-16.39	-32.31	-20.27	-17.97	-38.30	-21.12	-19.10
13670	-21.99	-18.31	-17.56	-25.01	-19.42	-19.51	-30.23	-20.68	-21.96	-35.47	-21.51	23.75
16610	-22.78	-17.36	-18.08	-24.87	-18.16	-19.45	-27.69	-18.89	-20.99	-29.91	-19.28	-22.00
17150	-19.36	-15.67	-11.59	-21.39	-17.84	-13.70	-23.66	-20.43	-16.58	-25.13	-22.23	-18.90
18700	-17.54	-14.11	-11.65	-19.45	-15.54	-12.97	-21.68	-17.05	-14.42	-23.20	-17.99	-15.37
24880	-27.57	-23.19	-20.71	-29.78	-25.68	-23.23	-32.26	-28.92	-26.57	-33.90	-31.39	-29.16
25590	-24.15	-19.17	-17.98	-27.08	-20.57	-20.38	-31.49	-22.16	-23.71	-35.38	-23.22	-26.41
28800	-24.23	-20.97	-17.14	-26.79	-22.68	-18.71	-30.11	-24.95	-20.51	-32.64	-26.71	21.72
31620	-22.23	-17.76	-16.04	-25.04	-19.05	-17.64	-29.31	-20.41	-19.74	-33.09	-21.26	-21.36
37230	-19.92	-15.23	-12.18	-22.69	-17.32	-13.68	-25.83	-19.99	-15.60	-27.93	-21.99	-17.03
39040	-16.33	-9.68	-7.22	-19.13	-10.70	-8.67	-22.40	-11.62	-10.44	-24.65	-12.12	-11.71
42920	-23.53	-16.18	-13.79	-25.62	-16.95	-15.13	-27.74	-17.55	-16.70	-29.01	-17.83	17.79
44560	-11.21	-6.65	-5.55	-13.06	-7.77	-6.67	-15.10	-9.05	-7.96	-16.42	-9.92	-8.84
46610	-15.69	-10.53	-9.09	-17.95	-11.23	-10.13	-20.65	-11.85	-11.34	-22.54	-12.18	-12.16
50300	-16.00	-11.74	-9.91	-17.61	-12.50	-11.10	-19.45	-13.15	-12.60	-20.67	-13.48	-13.73
50540	-8.84	-6.89	-4.63	-10.48	-7.71	-5.88	-12.47	-8.44	-7.54	-13.89	-8.82	-8.85
51590	-23.01	-15.27	-14.04	-25.43	-16.16	-15.66	-28.10	-16.97	-17.76	-29.83	-17.40	-19.37
60500	-10.34	-6.93	-4.98	-11.46	-7.59	-6.21	-12.81	-8.18	-7.99	-13.75	-8.49	-9.48
60990	-5.48	-3.13	-1.54	-6.67	-3.61	-2.91	-8.28	-4.04	-6.45	-9.56	-4.25	-11.58
65110	-14.89	-9.10	-7.53	-16.92	-10.09	-9.32	-19.65	-11.13	-12.32	-21.8	-11.78	-15.24
66730	-20.60	-13.83	-12.99	-21.92	-15.71	-14.82	-23.12	-18.46	-17.47	-23.75	-20.82	-19.71
68340	-20.60	-13.20	-12.26	-21.91	-14.43	-14.33	-23.06	-15.87	-17.23	-23.66	-16.86	19.61
69100	-19.47	-12.24	-10.55	-20.91	-12.92	-12.17	-22.27	-13.49	-14.33	-23.03	-13.77	-16.04
77750	-29.76	-23.44	-20.12	-31.04	-24.33	-21.61	-32.19	-25.09	-23.35	-32.8	-25.48	-24.53
80700	-12.03	-8.18	-6.99	-12.91	-8.48	-7.43	-13.78	-8.71	-7.79	-14.29	-8.80	-7.97
82290	-12.66	-10.08	-7.69	-13.97	-10.57	-8.32	-15.76	-10.95	-8.93	-17.19	-11.12	- 9.29
86500	-12.50	-8.89	-7.16	-13.94	-9.32	-7.49	-15.58	-9.69	-7.76	-16.67	-9.88	-7.89
89350	-29.45	-21.67	-18.05	-30.96	-22.00	-19.18	-32.61	-22.19	-20.61	-33.67	-22.24	-21.69
90450	-14.68	-10.41	-8.56	-15.31	-10.91	-8.83	-15.80	-11.40	-9.04	-16.03	-11.68	9.15
91750	-26.13	-17.05	-14.99	-27.39	-17.92	-15.93	- 28.5	-18.75	-16.98	-29.08	-19.21	-17.67
92350	-15.95	-12.46	-10.26	-17.58	-13.02	-11.02	-19.75	-13.49	-11.92	-21.42	-13.72	-12.54
93140	-24.13	-19.00	-15.95	-26.05	-19.5	-17.36	-28.75	-19.85	-19.39	-30.98	-19.99	21.1
93300	-36.51	-26.46	-23.90	-38.08	-26.79	-24.56	-39.60	-26.98	-25.14	-40.48	-27.04	-25.45
98550	-15.98	-12.67	-10.38	-17.45	-13.53	1-0.92	-19.13	-14.37	1-1.41	2-0.27	-14.86	-11.66

Table 30:. Temperatures: return values using GEV distribution with annual minimum daily mean temperatures (in degrees Celsius)

station	5 years			10 years			25 years			50 years		
	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
12680	22.86	16.96	16.82	23.16	17.13	17.56	23.31	17.25	18.83	23.36	17.29	20.08
13670	19.3	13.38	16.48	19.99	13.84	16.97	20.6	14.36	17.57	20.91	14.71	18.01
16610	17.91	12.35	12.45	18.39	12.96	13.05	18.78	13.72	13.82	18.97	14.26	14.4
17150	22.72	17.63	17.5	23.35	18.05	18.08	23.99	18.5	18.93	24.36	18.79	19.63
18700	23.74	18.01	17.87	24.47	18.34	18.44	25.18	18.68	19.32	25.6	18.89	20.1
24880	23.25	17.25	17.35	23.96	17.59	18.08	24.6	17.93	19.26	24.93	18.13	20.36
25590	18.85	13.26	13.35	19.62	13.81	13.99	20.35	14.46	14.93	20.77	14.93	15.73
28800	21.78	13.24	16.21	22.21	13.63	16.83	22.51	14.06	17.72	22.64	14.35	18.48
31620	19.02	12.54	12.58	19.85	13.07	13.12	20.45	13.73	13.92	20.7	14.21	14.6
37230	22.12	16.97	16.89	22.84	17.66	17.52	23.53	18.68	18.57	23.93	19.55	19.59
39040	22.49	16.43	16.62	23.12	17.09	17.1	23.66	18.1	17.71	23.94	19.02	18.17
42920	19.51	14.25	14.16	19.97	14.9	14.77	20.39	15.76	15.66	20.61	16.41	16.41
44560	22.63	15.9	16.59	23.61	16.8	17.56	24.85	18.64	19.3	25.75	20.82	21.12
46610	22.49	16.91	16.96	23.19	17.47	17.6	23.9	18.13	18.51	24.32	18.6	19.28
50300	20.33	14.86	14.77	21.03	15.49	15.61	21.74	16.21	16.95	22.15	16.7	18.2
50540	22.88	18.72	18.78	23.41	19.29	19.89	23.89	19.86	21.72	24.14	20.19	23.46
51590	22.24	16.59	16.36	22.77	17.1	17.45	23.21	17.66	19.84	23.42	18.01	22.82
60500	22.48	15.73	16.08	23.76	16.05	16.44	25.41	16.39	16.82	26.65	16.61	17.07
60990	20.67	15	15.61	21.52	15.33	15.67	22.39	15.7	15.69	22.92	15.95	15.7
65110	22.8	15.9	15.9	23.7	16.4	16.3	24.2	16.9	16.8	24.5	17.2	17.2
66730	19.75	14.27	14.1	20.37	14.81	14.51	20.9	15.43	15	21.17	15.85	15.33
68340	22.05	16.47	16.09	22.93	17.12	16.6	23.91	17.87	17.25	24.56	18.38	17.73
69100	23.47	17.64	17.3	24.78	18.25	17.77	26.57	18.96	18.34	27.99	19.43	18.75
77750	21.14	15.99	15.43	22.23	16.56	15.81	23.92	17.22	16.24	25.45	17.66	16.53
80700	21.98	16.02	15.72	22.77	16.59	16.19	23.58	17.21	16.76	24.08	17.62	17.18
82290	21.68	16.2	16.32	22.55	16.71	16.68	23.44	17.26	17.09	23.98	17.61	17.36
86500	21.89	15.79	16	22.33	16.42	16.63	22.62	17.1	17.39	22.73	17.53	17.93
89350	22.54	16.87	16.84	23.1	17.37	17.63	23.56	17.86	18.71	23.79	18.14	19.57
90450	21.21	16.55	16.58	21.69	17.23	17.23	22.04	17.93	17.94	22.19	18.35	18.41
91750	22.81	16.53	16.61	23.39	17.23	17.28	23.88	18.06	18.1	24.11	18.65	18.7
92350	20.64	15.22	15.19	20.95	15.93	15.79	21.16	16.93	16.61	21.24	17.74	17.28
93140	23.48	16.58	16.62	24.3	17.58	17.24	25.08	19.25	18.05	25.51	20.88	18.68
93300	21.12	14.37	14.55	21.85	15.09	15.57	22.55	16.08	17.17	22.96	16.89	18.65
98550	18.13	15.09	16.13	19.25	15.44	16.57	20.51	15.7	17.02	21.35	15.82	17.3

Table 31: Temperatures: return values using GEV distribution with annual maximum daily mean temperatures(in degrees Celsius)

station	5 years			10 years			25 years			50 years		
	obs	**ctr**	sc	**obs**	**ctr**	sc	**obs**	**ctr**	sc	**obs**	**ctr**	sc
12680	18.44	15.67	15.35	20.09	16.05	15.77	21.73	16.81	16.61	22.83	17.57	17.03
13670	15.2	11.93	13.96	16.76	12.62	15.32	18.33	13.31	17.12	19.37	13.65	18.03
16610	14.2	10.43	10.46	15.61	11.46	11.16	17.48	12.15	11.86	18.42	12.85	12.57
17150	19.21	16.46	16.08	20.36	17.28	16.93	22.08	18.11	17.36	23.22	18.53	17.78
18700	20.42	17.1	16.7	21.64	17.51	17.13	23.47	18.33	17.99	24.69	18.75	18.43
24880	20.83	16.11	15.8	22.59	16.51	16.66	24.35	17.31	17.09	26.11	17.72	17.96
25590	15.09	11.77	11.87	16.1	12.46	12.24	18.13	13.15	12.97	19.14	13.83	13.71
28800	18.1	11.75	14.07	19.19	12.42	15.39	20.82	13.1	16.72	21.9	13.77	17.6
31620	14.62	11.32	11.09	15.67	11.96	11.78	17.24	12.61	12.48	18.29	13.26	12.83
37230	18.89	15.91	15.44	20	16.33	16.27	21.67	17.18	16.68	22.23	17.61	17.51
39040	18.88	15.59	15.22	20.02	15.98	15.61	21.72	16.77	16.41	22.85	17.16	16.81
42920	16.36	12.72	12.77	17.37	13.43	13.14	18.87	14.14	13.87	20.38	14.86	14.24
44560	17.98	14.99	14.6	19.79	15.37	15.4	21	16.13	15.8	22.21	16.51	16.6
46610	18.78	15.32	15.35	19.89	16.13	15.77	21.55	16.94	16.61	22.66	17.35	17.03
50300	16.58	13.12	12.76	17.64	13.5	13.53	19.24	14.64	13.92	20.3	15.02	14.69
50540	18.64	16.43	16.07	19.78	17.35	17.04	21.47	18.27	18.02	22.6	19.19	18.51
51590	18.14	14.72	14.72	19.81	15.53	15.12	21.48	16.34	15.94	22.59	16.75	16.34
60500	17.94	14.76	14.78	19.18	15.13	15.15	20.41	15.5	15.53	21.65	15.87	15.9
60990	16.36	13.95	14.27	17.48	14.65	14.61	18.6	15	15.29	19.73	15.35	15.63
65110	17.55	14.37	13.98	18.75	14.76	14.37	20.55	15.54	15.15	21.75	16.33	15.54
66730	16.01	12.33	12.21	17.57	13.09	12.91	19.14	13.85	13.61	20.18	14.62	13.96
68340	18.27	14.32	14.2	19.45	15.18	15	21.22	16.04	15.8	22.39	16.9	16.2
69100	18.74	15.81	15.27	20.04	16.72	16.11	21.98	17.62	16.53	23.28	18.07	16.96
77750	17.02	14.28	13.74	18.69	15.11	14.13	20.35	15.95	14.9	21.45	16.36	15.67
80700	17.08	14.12	14.09	18.9	14.92	14.49	20.71	15.73	15.29	21.92	16.54	15.68
82290	17.34	14.52	14.5	18.54	14.93	15.3	20.34	15.74	16.1	21.54	16.55	16.5
86500	15.89	13.82	13.86	17.56	14.62	14.7	19.23	15.43	15.53	20.34	16.24	15.95
89350	17.23	14.77	14.55	19.02	15.6	15.48	20.81	16.43	16.4	22	17.26	17.33
90450	16.25	14.15	14.18	17.37	15.01	15.05	19.59	15.87	16.36	20.71	16.73	17.24
91750	16.88	14.42	14.54	18.69	14.85	14.99	20.5	16.12	16.34	21.71	16.54	16.78
2350	15.78	13.44	13.49	16.81	14.22	14.29	18.37	14.99	15.08	19.92	15.37	15.88
93140	17.54	14.56	15	18.85	15.44	15.88	21.47	16.32	16.76	22.78	16.76	17.2
93300	15.97	12.57	12.58	17.13	13.32	13.01	19.44	14.06	13.87	20.59	14.44	14.73
98550	12.15	13.33	14.31	13.23	14.43	15.12	14.84	15.16	16.32	16.46	15.89	17.13

Table 32: Temperatures: return values using GPD distribution with annual maximum daily mean temperatures in degrees Celsius (the threshold is the 90th percentile)

station	5 years			10 years			25 years			50 years		
	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc	obs	ctr	sc
12680	-15.44	-11.35	-10.58	-17.97	-12.99	-12.14	-21.34	-14.64	-13.7	-23.02	-16.29	-15.26
13670	-14.96	-11.7	-12.98	-16.58	-13.43	-14.75	-19.01	-15.73	-17.11	-20.63	-17.46	-18.3
16610	-15.3	-12.13	-12.07	-16.93	-13.58	-13.7	-19.38	-15.52	-15.89	-21.83	-16.98	-17.52
17150	-12.08	-8.08	-7.18	-14.76	-9.91	-9.3	-17.44	-12.35	-11.42	-19.46	-14.18	-13.54
18700	-10.89	-7.52	-6.92	-13.39	-9.56	-8.22	-15.89	-11.6	-10.38	-17.76	-13.12	-12.12
24880	-19.35	-13.98	-13.62	-22.04	-16.52	-15.19	-25.62	-19.06	-18.33	-27.41	-21.6	-20.69
25590	-16.16	-12	-11.35	-18.93	-13.96	-13.3	-21.7	-16.56	-15.91	-23.55	-17.86	-17.86
28800	-16.44	-13.96	-13.77	-19.05	-15.94	-15.43	-21.66	-18.57	-18.19	-24.27	-20.55	20.4
31620	-15.09	-11.82	-11.14	-16.68	-13.56	-12.84	-19.87	-15.30	-14.54	-21.47	-16.46	-16.23
37230	-11.60	-9.07	-7.99	-13.73	-10.84	-9.58	-17.29	-13.2	-11.7	-19.42	-14.98	- 13.3
39040	-8.01	-4.30	-3.35	-10.86	-5.75	-5.24	-14.42	-7.93	-7.13	-16.56	-9.02	-8.64
42920	-13.95	-10.46	-9.38	-17.27	- 11.91	-10.96	-20.58	-13.84	-13.07	-23.07	-15.29	-14.66
44560	-4.55	-2.51	-2.05	-6.47	-4.00	-3.46	-8.87	5-.19	-5.15	-10.79	-6.08	-6.28
46610	-8.41	-6.12	-5.56	-11.04	-7.54	-6.67	-13.01	-8.96	-8.52	-14.97	-9.67	-9.63
50300	-9.42	-6.98	-6.37	-11.21	-8.11	-7.93	-13.6	-9.61	-9.1	-14.8	-10.74	-10.28
50540	-3.11	-2.14	-1.78	-4.90	-3.30	-3.08	-6.68	-5.05	-4.64	- 8.01	- 5.92	-5.69
51590	-13.52	-9.19	-8.92	-16.02	-11.08	-10.57	-20.18	-12.98	-12.77	-21.85	-14.40	-14.42
60500	-5.20	-2.33	-2.04	-6.86	-3.37	-3.12	-8.52	-4.68	-4.42	-9.77	-5.73	-5.29
60990	-1.94	0.59	0.47	-2.77	-0.36	-0.47	-4.16	-1.47	-1.71	-4.99	-2.42	-2.64
65110	-8.55	-4.77	-4.11	-10.78	-5.77	-5.44	-13.02	-7.43	-7.1	-14.7	-8.1	-8.43
66730	-12.97	-8.41	-8.51	-14.94	-9.89	-10.04	-17.57	-11.87	-12.08	-19.54	-13.35	-13.6
68340	-11.66	-7.25	-6.89	-14.26	-8.58	-8.45	-17.50	-10.81	-10.53	-20.10	-12.14	-12.09
69100	-10.45	-6.20	-5.99	-13.06	-7.77	-7.33	-16.34	-9.71	-9.57	-18.30	1-0.87	-10.92
77750	-20.66	-15.41	-14.74	-24.20	-17.43	-16.76	-27.75	-20.13	-18.78	-30.41	-21.48	-20.80
80700	-7.27	-4.59	-4.21	-8.84	-5.59	-5.12	-10.42	-6.59	-6.04	-11.59	-7.34	-6.49
82290	-8.58	-5.94	-4.86	-9.90	-6.89	-5.67	-11.66	-8.47	-6.75	-12.98	-9.11	-7.56
86500	-7.27	-6.15	- 5.21	-9.02	-7.23	-6.05	-10.78	-8.31	-6.89	-12.10	-9.13	-7.52
89350	-21.57	-15.84	-14.23	-24.24	-17.54	-15.98	-27.81	-19.80	-17.72	-30.49	-20.93	18.88
90450	-9.88	-7.59	-6.51	-11.58	-8.50	-7.21	-12.85	-9.72	-8.15	-14.13	-10.63	-8.85
91750	-17.14	-12.24	-11.55	-19.44	-13.71	-12.46	-22.52	-15.65	-13.83	-24.05	-16.63	15.2
92350	-10.41	-8.42	-7.36	-11.93	-9.55	-8.34	-13.96	-10.67	-9.33	-14.97	-11.80	-9.99
93140	-17.39	-13.46	-12.22	-19.87	-15.00	-13.67	-22.34	-17.04	-15.11	-24.81	-18.07	16.56
93300	-26.40	-19.82	-18.83	-29.60	-21.83	-20.70	-33.87	-24.51	-23.20	-36.00	-25.85	-24.45
98550	-9.95	-8.81	-8.21	-11.51	-9.60	-9.08	-13.60	-11.18	-10.52	-14.64	-11.97	-11.39

Table 33: Temperature return values using GPD distribution with annual minimum daily mean temperatures in degrees Celsius (the threshold is the 90th percentile when considering the absloute values)

```
# Code to put all the files into the same format
# source("fix.files.R")
# example of call: dumb_fix.files()

# This routine gets the name of files stored on the disc in directory
# (direc), and returns the names in a vector of characters.

avail.files <- function(direc="",pattern.1="",pattern.2="") {
  dir.0 <- getwd()
  if (direc!="") setwd(direc)
  avail.files <- list.files(pattern=pattern.1)
  avail.files <- avail.files[grep(pattern.2,avail.files)]
  remove <- grep("~",avail.files)    # get rid of files with suffixe '.txt~'
  avail.files <- setdiff(avail.files,avail.files[remove])
  setwd(dir.0)
  avail.files          # returns all the files containing the datasets.
}

# This routine permits to define all the files with the same format.
# All the columns are separated by one white space and missing values are
# refered as NA.
# The datasets written in the new format are written in a new file (suffixe='.fix')

fix.files <- function(files="search",direc="obs80-99-txt") {
  print(getwd())
  dir.0 <- getwd()
  setwd(direc)
  print(getwd())
  if (files=="search") {
    files<-avail.files(pattern.1="obs2.txt")
  }
 # read all the files of the directory
   for (file in files) {
    con <- file(file, "r", blocking = FALSE)
    in.text <- readLines(con)
    close(con)
    out.text <- in.text

 # loop to read all the lines of the file and correct the format if necessary
    for (i.line in 1:length(in.text)) {
      corrected <- in.text[i.line]
      print(corrected)

    # dots between days, months and years are replaced by a white space
    # so that each column are separated in the same way.
      substr(corrected,9,9)<-" "
      substr(corrected,12,12)<-" "

    # Missing values are replaced by NA
          if ((substr(corrected,18,18)==" ")|(substr(corrected,18,19)=="- "))
      substr(corrected,18,21)<-"NA  "

           if ((substr(corrected,24,24)==" ")|(substr(corrected,24,25)=="- ")){
             if (nchar(corrected)<=26) corrected <- paste(corrected,"  ")
     substr(corrected,24,27)<-"NA  "
    }
      # all the corrections are written in the output file
        out.text[i.line]<-corrected
      }  # end of loop on lines

   # Definition of the name of the new well-formatted file
```

```
      fixed.file<-paste(substr(file,1,nchar(file)-3),"fix",sep="") # replace '.txt' by '.fix'
      print(c(file,' fixed ---> ',fixed.file))
      con <- file(fixed.file, "w", blocking = FALSE)
      writeLines(out.text,con)
      close(con)
   } # end of loop on files
   setwd(dir.0)
}

# This routine permits to order the columns in the same way for all the files.
# nbr.station is the number of the station. The full name of the file does not
# need to be specified.
# example of call: check.columns("99910")

check.columns <- function(nbr.station){
  data <- paste(nbr.station,"-obs2.ok",sep="")
  print(data)
  st.obs <- read.fwf(paste("/home/alexi/alexi/scenario/obs80-99-txt/",data,sep=""),
widths=c(5,3,3,5,6,6),col.names=c("station","day","month","year","temp","rr")                    ,as.is=
  st.obs$rr <- as.real(st.obs$rr)
  st.obs$temp <- as.real(st.obs$temp)
  source("check.data.R")
  res <- check.data(st.obs$rr,st.obs$temp)
  st.obs$rr <- res$rr
  st.obs$temp <- res$temp
  st.obs
}

check.all.files <- function(){
setwd("obs80-99-txt")
names <- avail.files(pattern.1="obs2.txt")
setwd("..")
print(names)
nbr <- c()
i <- 1
  for (i.names in 1:length(names)){
      if (substr(names[i.names],5,5)=="-")
          nbr[i] <- substr(names[i.names],1,4)
      else{
          if (substr(names[i.names],4,4)=="-")
             nbr[i] <- substr(names[i.names],1,3)
          else nbr[i] <- substr(names[i.names],1,5)
           }
  i <- i+1
  }
print(nbr)
  for (i.nbr in 1:length(nbr)) check.columns(nbr[i.nbr])
}

# This routine permits to replace all '0' by '0.01' in rr column.
# It has no argument
change.0 <- function(){
source("avail.files.R")
files <- avail.files("obs80-99-txt",".fix","-obs2")
# read all the files of the directory
   for (file in files) {
    con <- file(file, "r", blocking = FALSE)
    in.text <- readLines(con)
    close(con)
    out.text <- in.text
    found.string <- F
```

```
    i.line <- 1
    while (i.line<length(in.text) & found.string==F) {
      correct <- in.text[i.line]
# 0.0 is used to know the index of the column containing precipitation.
# (null temperatures are noted 0, and 0.0 never appears in the temp column).
      res.find <- find.substr("0.0",correct)
        if (res.find$success==T){
           start <- res.find$position
           found.string <- T
        }
    i.line <- i.line+1
    }
   if (found.string==T){
     for (i.line in 1:length(in.text)){
         correct <- in.text[i.line]
        if (substr(correct,start,start+1)=="0 ")
      substr(correct,start,start+3)<-"0.01"
        out.text[i.line]<-correct
      }
   }
     correct.file<-paste(substr(file,1,nchar(file)-3),"ok",sep="")
     con <- file(correct.file, "w", blocking = FALSE)
     writeLines(out.text,con)
     close(con)
  }
  setwd(dir.0)
}
```

```
# This function checks which of the two vectors
# contains rainfall (daily values) and which
# holds temperatures. The decision is based on
# statistical properties and 5 tests

check.data <- function(x.1,x.2) {
  test.result <- rep(NA,5)
  min.1 <- min(x.1[!is.nan(x.1) & is.finite(x.1)],na.rm=T)
  max.1 <- max(x.1[!is.nan(x.1) & is.finite(x.1)],na.rm=T)
  mean.1 <- mean(x.1,na.rm=T)
  min.2 <- min(x.2[!is.nan(x.2) & is.finite(x.2)],na.rm=T)
  max.2 <- max(x.2[!is.nan(x.2) & is.finite(x.2)],na.rm=T)
  mean.2 <- mean(x.2,na.rm=T)

# x.2 contains rainfall: minimum value is 0.0
# Temperature may be lower than OdegC.
  if ((min.1 < 0) & (min.2>=0)) {
   temp <- x.1
   rr <- x.2
  }
  if ((min.2 < 0) & (min.1>=0)) {
   temp <- x.2
   rr <- x.1
  }
```

```
  else{
   if ((min.1 < 0) & (min.2>=0)) {
    temp <- x.1
    rr <- x.2
    test.result[1] <- 1
   }
   if ((min.2 < 0) & (min.1>=0)) {
    temp <- x.2
    rr <- x.1
    test.result[1] <- -1
   }
   if (max.1 < max.2) {
    temp <- x.1
    rr <- x.2
    test.result[2] <- 1
   }
   if (max.2 < max.1) {
    temp <- x.2
    rr <- x.1
    test.result[2] <- -1
   }
   if (mean.1 < mean.2) {
    temp <- x.1
    rr <- x.2
    test.result[3] <- 1
   }
   if (mean.2 < mean.1) {
    temp <- x.2
    rr <- x.1
    test.result[3] <- -1
   }
   if (sum(x.1==0,na.rm=T) < sum(x.2==0,na.rm=T)) {
    temp <- x.1
    rr <- x.2
    test.result[4] <- 1
   }
   if (sum(x.2==0,na.rm=T) < sum(x.1==0,na.rm=T)) {
    temp <- x.2
    rr <- x.1
    test.result[4] <- -1
   }
# Last test:
   if (abs(sum(test.result,na.rm=T)) < 4) {
     print(summary(x.1))
     print(summary(x.2))
     print(test.result)
     i.temp <- as.integer(readline("Which is temperature (1 or 2)?"))
     if (i.temp == 1) {
       temp <- x.1
       rr <- x.2
     } else {
      temp <- x.2
      rr <- x.1
     }
   }
  }
   check.data <- list(temp=temp,rr=rr)
   check.data
}
```

```
# This routine searches the string for a matching
# pattern that may be embedded somewhere in the string.
# source("find.substr.R")

find.substr <- function(pattern,in.str) {
  pos<-c()
  found <- F
  for (i in 1:nchar(in.str)) {
    sstrng <- substr(in.str,i,nchar(in.str))
    if (!is.na(pmatch(pattern,sstrng))) {
      pos<-c(pos,i)
      found <- T
    }
  }
 res <- list(success=found, position=pos)
 res
}
```

```
# This code realizes several statistical tests (means, variance, percentiles).
# source("rain_study.R")

# Mean study: determines mean values (annual, seasonal,monthly)
#  data = file for the dataset
#  rr = T if the study is on precipitations,
# rr = F if the study is on temperatures.

mean.study <- function(data, rr){
  if (rr){
data.choice <- data$rr
variable <- 'precipitations'
}
  else{
data.choice <- data$temp
variable <- 'temperatures'
}
# yearly mean values
mean.year <- mean(data.choice,na.rm=T)

# seasonal mean values
# Spring
mean.spring <- mean(data.choice[(data$month==3 | data$month==4 | data$month==5)],na.rm=T)
# Summer
mean.summer <- mean(data.choice[(data$month==6 | data$month==7 | data$month==8)],na.rm=T)
# Autumn
mean.autumn <- mean(data.choice[(data$month==9 | data$month==10 | data$month==11)],na.rm=T)
# Winter
mean.winter <- mean(data.choice[(data$month==1 | data$month==2 | data$month==12)],na.rm=T)

# monthly mean values
mean.month <- c()
    for (i in 1:12){
m.month <- mean(data.choice[(data$month==i)],na.rm=T)
mean.month <- cbind(mean.month,m.month)
}
```

```
mean.values <- list(variable=variable,annual=round(mean.year,2),spring=round(mean.spring,2),
summer=round(mean.summer,2), autumn=round(mean.autumn,2), winter=round(mean.winter,2),month=round(mean.
mean.values
}

# Variance study: determines mean values (annual, seasonal,monthly)
#  data = file for the dataset
#  rr = T if the study is on precipitations,
# rr = F if the study is on temperatures.

var.study <- function(data,rr){
  if (rr){
data.choice <- data$rr
variable <- 'precipitations'
}
  else{
data.choice <- data$temp
variable <- 'temperatures'
}
var.month <- c()

# yearly variances
var.year <- var(data.choice,na.rm=TRUE)

# seasonal variances
# Spring
var.spring <- var(data.choice[(data$month==3 | data$month==4 | data$month==5)],na.rm=TRUE)
# Summer
var.summer <- var(data.choice[(data$month==6 | data$month==7 | data$month==8)],na.rm=TRUE)
# Autumn
var.autumn <- var(data.choice[(data$month==9 | data$month==10 | data$month==11)],na.rm=TRUE)
# Winter
var.winter <- var(data.choice[(data$month==1 | data$month==2 | data$month==12)],na.rm=TRUE)
# monthly variances
   for (i in 1:12){
v.month <- var(data.choice[(data$month==i)],na.rm=TRUE)
var.month <- cbind(var.month,v.month)
   }
var.values <- list(variable=variable,annual=round(var.year,2),spring=round(var.spring,2),
summer=round(var.summer,2), autumn=round(var.autumn,2), winter=round(var.winter,2), month=round(var.mor
var.values
}

# monthly, seasonal and annual mean values
# determined for each year

detail.means <- function(data,rr){
if (rr){
variable <- 'precipitations'
obs.choice <- data$rr
}
    else{
variable <- 'temperatures'
obs.choice <- data$temp
}

# yearly mean values
meanplot.year <- c()
first.year <- min(data$year,na.rm=T)
last.year <- max(data$year,na.rm=T)
  for(y in first.year:last.year)
```

```
{
mplot.year <- mean(obs.choice[data$year==y],na.rm=T)
meanplot.year <- rbind(meanplot.year,mplot.year)
}

# for seasonal mean values
meanplot.spring <- c()
meanplot.summer <- c()
meanplot.autumn <- c()
meanplot.winter <- c()
meanplot.month <- c()

# Spring
   for(y in first.year:last.year){
       i.select <- ((data$month==3 | data$month==4 | data$month==5) & (data$year==y))
       if (sum(i.select)>0){
       mplot.spring <- mean(obs.choice[(data$month==3 | data$month==4 | data$month==5)
& (data$year==y)],na.rm=T)
meanplot.spring <- rbind(meanplot.spring,mplot.spring)
}
       else meanplot.spring <- rbind(meanplot.spring, NA)
     }
# Summer
   for(y in first.year:last.year){
        i.select <- ((data$month==6 | data$month==7 | data$month==8) & (data$year==y))
        if (sum(i.select)>0){
mplot.summer <- mean(obs.choice[(data$month==6 | data$month==7 | data$month==8)
& (data$year==y)],na.rm=T)
meanplot.summer <- rbind(meanplot.summer,mplot.summer)
}
        else meanplot.summer <- rbind(meanplot.summer, NA)
     }
# Autumn
   for(y in first.year:last.year){
        i.select <- ((data$month==9 | data$month==10 | data$month==11) & (data$year==y))
        if (sum(i.select)>0){
mplot.autumn <- mean(obs.choice[(data$month==9 | data$month==10 | data$month==11)
& (data$year==y)],na.rm=T)
meanplot.autumn <- rbind(meanplot.autumn,mplot.autumn)
}
        else meanplot.autumn <- rbind(meanplot.autumn, NA)
     }
# Winter
   for(y in first.year:last.year){
        i.select <- ((data$month==1 | data$month==2 | data$month==12) & (data$year==y))
        if (sum(i.select)>0){
mplot.winter <- mean(obs.choice[(data$month==1 | data$month==2 | data$month==12)
& (data$year==y)],na.rm=T)
meanplot.winter <- rbind(meanplot.winter,mplot.winter)
}
        else meanplot.winter <- rbind(meanplot.winter, NA)
      }
# for monthly mean values
   for(y in first.year:last.year){
for (i in 1:12){
   mplot.month <- mean(obs.choice[(data$month==i) & (data$year==y)],na.rm=T)
            meanplot.month <- rbind(meanplot.month,mplot.month)
          }
   }
dim(meanplot.month) <- c(12,last.year-first.year+1)
res <- list(annual=meanplot.year,spring=meanplot.spring,summer=meanplot.summer,
```

```
            autumn=meanplot.autumn, winter=meanplot.winter, month=meanplot.month)
res
}

# seasonal and annual variances
# determined for each year

detail.var <- function(data,rr){
if (rr){
variable <- 'precipitations'
data.choice <- data$rr
}
    else{
variable <- 'temperatures'
data.choice <- data$temp
}

# yearly mean values
variance.year <- c()
first.year <- min(data$year,na.rm=T)
last.year <- max(data$year,na.rm=T)
  for(y in first.year:last.year)
{
var.year <- var(data.choice[data$year==y],na.rm=T)
variance.year <- rbind(variance.year,var.year)
}

# plots for seasonal mean values
variance.spring <- c()
variance.summer <- c()
variance.autumn <- c()
variance.winter <- c()
variance.month <- c()

# Spring
   for(y in first.year:last.year){
      i.select <- ((data$month==3 | data$month==4 | data$month==5) & (data$year==y))
      if (sum(i.select)>0){
          aux <- data.choice[(data$month==3 | data$month==4 | data$month==5) & (data$year==y)]
          var.spring <- var(aux,na.rm=T)
  variance.spring <- rbind(variance.spring,var.spring)
}
        else variance.spring <- rbind(variance.spring, NA)
     }
# Summer
   for(y in first.year:last.year){
      i.select <- ((data$month==6 | data$month==7 | data$month==8) & (data$year==y))
      if (sum(i.select)>0){
      var.summer <- var(data.choice[(data$month==6 | data$month==7 | data$month==8)
& (data$year==y)],na.rm=T)
      variance.summer <- rbind(variance.summer,var.summer)
}
        else variance.summer <- rbind(variance.summer, NA)
     }
# Autumn
   for(y in first.year:last.year){
i.select <- ((data$month==9 | data$month==10 | data$month==11) & (data$year==y))
        if (sum(i.select)>0){
var.autumn <- var(data.choice[(data$month==9 | data$month==10 | data$month==11)
& (data$year==y)],na.rm=T)
variance.autumn <- rbind(variance.autumn,var.autumn)
```

```
}
        else variance.autumn <- rbind(variance.autumn, NA)
    }
# Winter
   for(y in first.year:last.year){
        i.select <- ((data$month==1 | data$month==2 | data$month==12) & (data$year==y))
        if (sum(i.select)>0){
var.winter <- var(data.choice[(data$month==1 | data$month==2 | data$month==12)
& (data$year==y)],na.rm=T)
variance.winter <- rbind(variance.winter,var.winter)
}
        else variance.winter <- rbind(variance.winter, NA)
    }
res <- list(annual=variance.year,spring=variance.spring,summer=variance.summer,
           autumn=variance.autumn, winter=variance.winter)
res
}

# Study of the significance of the difference in mean values  using student test
# (useful only for temperatures)
#  data1 = dataset for the first mean
# data2 = dataset for the second mean (H0: mean(data1)==mean(data2))
# conf is the confidence level of the interval.

student.comparison <- function(data1,data2,conf){
  res.1 <- detail.means(data1,rr=F)
  res.2 <- detail.means(data2,rr=F)

# yearly values
diff.year <- t.test(res.1$annual,res.2$annual,conf.level=conf,
alternative = "two.sided")$p.value

# seasonal values
# Spring
diff.spring <- t.test(res.1$spring,res.2$spring,conf.level=conf,
alternative = "two.sided")$p.value
# Summer
diff.summer <- t.test(res.1$summer,res.2$summer,conf.level=conf,
alternative = "two.sided")$p.value
# Winter
diff.winter <- t.test(res.1$winter,res.2$winter,conf.level=conf,
alternative = "two.sided")$p.value
# Spring
diff.autumn <- t.test(res.1$autumn,res.2$autumn,conf.level=conf,
alternative = "two.sided")$p.value

# significant.diff is True if the difference is statistically significant
diff <- cbind(diff.year,diff.summer,diff.autumn,
#diff.mean.winter,diff.mean.spring,diff.mean.month)
                diff.winter,diff.spring)

significant.diff <- diff < 1-conf
difference <- list(confidence.level=conf,annual=round(diff.year,4), summer=round(diff.summer,4),
autumn=round(diff.autumn,4),winter=round(diff.winter,4), spring=round(diff.spring,4),
               significant.diff=significant.diff)
difference
}

# Study of the significance of the difference in mean values using wilcoxon test
#          data1 = file for the observations
# data2 = file for the forecasts
```

```
#               rr = T if the precipitations are studied
#                      if the temperatures are studied
# conf is the confidence level of the interval.
#               var =  T if variance comparison is computed,
#                      F if mean comparison is computed

wilcoxon.comparison <- function(data1,data2,rr,conf,var){
if (var==F){
  res.1 <- detail.means(data1,rr)
  res.2 <- detail.means(data2,rr)
  }

 else{
  res.1 <- detail.var(data1,rr)
  res.2 <- detail.var(data2,rr)
  }

#yearly mean values
diff.year <- wilcox.test(res.1$annual,res.2$annual,conf.level=conf,
alternative = "two.sided",exact=FALSE)$p.value
# seasonal mean values
# Spring
diff.spring <- wilcox.test(res.1$spring,res.2$spring,conf.level=conf,
alternative = "two.sided",exact=FALSE)$p.value
# Summer
diff.summer <- wilcox.test(res.1$summer,res.2$summer,conf.level=conf,
alternative = "two.sided",exact=FALSE)$p.value
# Winter
diff.winter <- wilcox.test(res.1$winter,res.2$winter,conf.level=conf,
alternative = "two.sided",exact=FALSE)$p.value
# Spring
diff.autumn <- wilcox.test(res.1$autumn,res.2$autumn,conf.level=conf,
alternative = "two.sided",exact=FALSE)$p.value

# significant.diff is True if the difference is significant
diff <- cbind(diff.year,diff.summer,diff.autumn,
#diff.mean.winter,diff.mean.spring,diff.mean.month)
                diff.winter,diff.spring)
significant.diff <- diff < 1-conf
difference <- list(confidence.level=conf,annual=round(diff.year,4), summer=round(diff.summer,4),
autumn=round(diff.autumn,4),winter=round(diff.winter,4), spring=round(diff.spring,4),
                significant.diff=significant.diff)
difference
}

# Study of the significance of the difference in variances using fisher test
#  data1 = population (dataset) for the first variance
# data2 = population (dataset) for the second variance (H0: variance(data1)==variance(data2))
# conf is the confidence level of the interval.

fisher.comparison <- function(data1,data2,conf){
    res.1 <- detail.var(data1,rr=F)
    res.2 <- detail.var(data2,rr=F)
#yearly mean values
diff.year <- var.test(res.1$annual,res.2$annual,conf.level=conf,
alternative = "two.sided")$p.value
# seasonal mean values
# Spring
diff.spring <- var.test(res.1$spring,res.2$spring,conf.level=conf,
alternative = "two.sided")$p.value
# Summer
```

```
diff.summer <- var.test(res.1$summer,res.2$summer,conf.level=conf,
alternative = "two.sided")$p.value
# Winter
diff.winter <- var.test(res.1$winter,res.2$winter,conf.level=conf,
alternative = "two.sided")$p.value
# Spring
diff.autumn <- var.test(res.1$autumn,res.2$autumn,conf.level=conf,
alternative = "two.sided")$p.value
# significant.diff is True if the difference is significant
diff <- cbind(diff.year,diff.summer,diff.autumn,diff.winter,diff.spring)
significant.diff <- diff < 1-conf
difference <- list(confidence.level=conf,annual=round(diff.year,4), summer=round(diff.summer,4),
autumn=round(diff.autumn,4),winter=round(diff.winter,4), spring=round(diff.spring,4),
                significant.diff=significant.diff)
difference
}

# plot to compare observed annual mean values and modelled annual mean values
#  obs = file for the observations
# sim = file for the forecasts
#  rr = T if the study is on precipitations,
# rr = F if the study is on temperatures.

compare.plot <- function(obs,sim,rr){
meanplot.year <- c()
meanplot.pr.year <- c()
    if (rr){
variable <- 'precipitations'
obs.choice <- obs$rr
sim.choice <- sim$rr
}
    else{
variable <- 'temperatures'
obs.choice <- obs$temp
sim.choice <- sim$temp
}
# yearly mean values
first.year <- min(obs$year,na.rm=T)
last.year <- max(obs$year,na.rm=T)
  for(y in first.year:last.year)
{
mplot.year <- mean(obs.choice[obs$year==y],na.rm=T)
meanplot.year <- rbind(meanplot.year,mplot.year)
}
   for(y in first.year:last.year)
{
mplot.pr.year <- mean(sim.choice[sim$year==y],na.rm=T)
meanplot.pr.year <- rbind(meanplot.pr.year,mplot.pr.year)
}
y.comp <- cbind(meanplot.year,meanplot.pr.year)
y.comp <- t(y.comp)
leg <- as.character(c(first.year:last.year))

   for (l in 1:length(y.comp))
if (is.na(y.comp[l])) y.comp[l]<-0
   barplot(y.comp, beside = TRUE,col=c('red','lavender'),
   names.arg=leg,legend=c('observation','model'))
   title(main = paste("observed and modelled yearly mean values for",variable),
font.main = 4)
}
```

```
# plot to compare observed mean values and modelled mean values
# comparison based on seasons and months.
#  data = R-object for the dataset to study
#  rr = T if the study is on precipitations,
# rr = F if the study is on temperatures.

plot.means <- function(data,rr){
res <- detail.means(data,rr)
 if (rr) variable <- 'precipitations'
  else variable <- 'temperatures'

# Plot of seasonal mean values over the whole period
x11()
op <- par(mfrow = c(2, 2))
first.year <- min(data$year,na.rm=T)
last.year <- max(data$year,na.rm=T)
leg <- as.character(c(first.year:last.year))

W <- barplot(res$winter, beside = TRUE,col='red',names.arg=leg)
title(main = paste("winter mean values for",variable), font.main = 4)
Sp <- barplot(res$spring, beside = TRUE,col='red',names.arg=leg)
        title(main = paste("spring mean values for",variable), font.main = 4)
S <- barplot(res$summer, beside = TRUE,col='red',names.arg=leg)
title(main = paste("summer mean values for",variable), font.main = 4)
A <- barplot(res$autumn, beside = TRUE,col='red',names.arg=leg)
title(main = paste("autumn mean values for",variable), font.main = 4)

# Plot of yearly mean values over the whole period
x11()
op <- par(mfrow = c(1, 2))
  for (l in 1:length(res$annual))
if (is.na(res$annual[l])) res$annual[l] <- 0
   barplot(res$annual,names.arg=leg,col='lavender',beside = TRUE)
   title(main = paste("yearly mean values for",variable), font.main = 4)

# Global overview
legend3 <- c('winter','spring','summer','autumn')
m <- cbind(res$winter,res$spring,res$summer,res$autumn)

global <- barplot(m, beside = TRUE,col='red',names.arg=legend3)
title(main = paste("seasonal mean values",variable), font.main = 4)
}

# Plotting frequency distribution
plot.freq <- function(data,rr){
if (rr){
variable <- 'precipitations'
data.choice <- data$rr
}
    else{
variable <- 'temperatures'
data.choice <- data$temp
}
# Over all the period
  if (rr){
x11()
vect.cpt <- c()
cpt0 <- sum(data==0,na.rm=TRUE)
cpt1 <- sum(data.choice<=1,na.rm=TRUE)
cpt2 <- sum(data.choice<=2,na.rm=TRUE)
cpt3 <- sum(data.choice<=3,na.rm=TRUE)
```

```
cpt4 <- sum(data.choice<=4,na.rm=TRUE)
cpt5 <- sum(data.choice<=5,na.rm=TRUE)
cpt6 <- sum(data.choice<=6,na.rm=TRUE)
cpt8 <- sum(data.choice<=8,na.rm=TRUE)
cpt10 <- sum(data.choice<=10,na.rm=TRUE)
cpt15 <- sum(data.choice<=15,na.rm=TRUE)
cpt20 <- sum(data.choice<=20,na.rm=TRUE)
cpt30 <- sum(data.choice<=30,na.rm=TRUE)
cpt50 <- sum(data.choice<=50,na.rm=TRUE)
cpt100 <- sum(data.choice<=100,na.rm=TRUE)
cpt <- length(data.choice)

vect.cpt <- cbind(cpt0,cpt1,cpt2,cpt3,cpt4,cpt5,cpt6,cpt8,cpt10,
cpt15,cpt20,cpt30,cpt50,cpt100,cpt)
percentile <- vect.cpt*100/cpt
lgd <- c('0','1','2','3','4','5','6','8','10','15','20','30','50','100','<=max')
rr.cum <- barplot(percentile,names.arg=lgd,xlab=c('precipitations (mm)'),
ylab=c('percentage'))
segments(rr.cum,percentile,rr.cum, percentile + 2*sqrt(1000*percentile/100), lwd = 1.5)
mtext(side = 3, line = -2,at= rr.cum,text = formatC(round(percentile,1)),
col = "black",font=1)
title(main = "Cumulative amount of percipitations over all the period for precipitations" ,font.main
}
##################
# Cumulative distribution: Annual analysis
# example: for year 1990
library(stepfun)
x11()
#y <- 1990
#yea <- data.choice[data$year==y]
#y.cum <- ecdf(yea)
#plot(y.cum, main = paste("Cumulative distribution for year",y))
y.cum <- ecdf(data.choice)
print(ecdf(data.choice)$y)
plot(y.cum, main = "Cumulative distribution")

###############
# Cumulative distribution: Seasonal analysis
x11()
op <- par(mfrow = c(2, 2))

# for season = winter
wint <- data.choice[(data$month==1 | data$month==2| data$month==12)]
w.cum <- ecdf(wint)
plot(w.cum, main = 'Winter')
abline(v=knots(w.cum),lty=2,col='gray70')

# for season = spring
spr <- data.choice[(data$month==3 | data$month==4 | data$month==5)]
sp.cum <- ecdf(spr)
plot(sp.cum, main = 'Spring')
abline(v=knots(sp.cum),lty=2,col='gray70')

# for season = summer
summ <- data.choice[(data$month==6 | data$month==7| data$month==8)]
s.cum <- ecdf(summ)
plot(s.cum, main = 'Summer')
abline(v=knots(s.cum),lty=2,col='gray70')

# for season = autumn
aut <- data.choice[(data$month==9 | data$month==10| data$month==11)]
```

```
a.cum <- ecdf(aut)
plot(a.cum, main = 'Autumn')
abline(v=knots(a.cum),lty=2,col='gray70')
}

# Routine to determine a specified percentile
#       data: dataset
#       percent: percentile to determine
#       rr: True if precipitation percentile
#           False if temperature percentile
# Example of call: percentile(st.obs,0.0001,F)

percentile <- function(data,percent,rr){
if (rr){
variable <- 'precipitations'
yea <- data$rr
wint <- data$rr[(data$month==1 | data$month==2| data$month==12)]
spr <- data$rr[(data$month==3 | data$month==4 | data$month==5)]
summ <- data$rr[(data$month==6 | data$month==7| data$month==8)]
aut <- data$rr[(data$month==9 | data$month==10| data$month==11)]
}
else{
  variable <- 'temperatures'
yea <- data$temp
wint <- data$temp[(data$month==1 | data$month==2| data$month==12)]
spr <- data$temp[(data$month==3 | data$month==4 | data$month==5)]
summ <- data$temp[(data$month==6 | data$month==7| data$month==8)]
aut <- data$temp[(data$month==9 | data$month==10| data$month==11)]
}
abs1 <- quantile(yea[!is.na(yea)],percent)
abs2 <- quantile(wint[!is.na(wint)],percent)
abs3 <- quantile(spr[!is.na(spr)],percent)
abs4 <- quantile(summ[!is.na(summ)],percent)
abs5 <- quantile(aut[!is.na(aut)],percent)

percent <- list(annual=round(abs1,2),spring=round(abs3,2),summer=round(abs4,2),autumn=round(abs5,2),wir
percent
}

# This routine plots on the same graph the cumulative distribution function of
# the observed, modelled and scenario periods.
# Both dry and wet days are considered.

plot.percentile <- function(data1,data2,data3){
annual1 <- hist(data1$rr,breaks=100)
annual2 <- hist(data2$rr,breaks=100)
annual3 <- hist(data3$rr,breaks=100)
e.d.f1 <- cumsum(annual1$density)/sum(annual1$density)
e.d.f2 <- cumsum(annual2$density)/sum(annual2$density)
e.d.f3 <- cumsum(annual3$density)/sum(annual3$density)

postscript("cumulative.ps")
plot(c(0,max(data1$rr,data2$rr,data3$rr,na.rm=T)),c(min(e.d.f1),max(e.d.f1))
     ,type="n",main=paste("Cumulative Distribution of P (station=", data1$station[1],")", sep=""),
      xlab="P(mm)",ylab="Percentiles")
lines(annual1$mids,e.d.f1,col="green",lty=1,lwd=1)
lines(annual2$mids,e.d.f2,col="blue",lty=3,lwd=2)
lines(annual3$mids,e.d.f3,col="red",lty=2,lwd=2)
legend(max(annual1$mids,na.rm=T)-30,0.7,
         c("observations","control period","scenario period"),
         col=c("green","blue","red"),lwd=c(1,2,2),
```

```
        lty=c(1,3,2),bg="grey95")
grid()

# if 95, 99 or 99th percentiles need to be shown on the graph
#abs1 <- min(annual1$mids[e.d.f1 > 0.95])
#abs2 <- min(annual1$mids[e.d.f1 > 0.99])
#abs3 <- min(annual1$mids[e.d.f1 > 0.991])
#lines(c(0,abs1),rep(0.95,2),col="red",lty=2)
#lines(rep(abs1,2),c(0.95,0),col="red",lty=2)
#lines(c(0,abs2),rep(0.99,2),col="blue",lty=2)
#lines(rep(abs2,2),c(0.99,0),col="blue",lty=2)
#lines(c(0,abs3),rep(0.991,2),col="green",lty=2)
#lines(rep(abs3,2),c(0.991,0),col="green",lty=2)
dev.off()
}
```

```
# This code permits to use GEV Distribution to study extreme values.
# source("extreme_value.R")

library(evd)

# Selection of the annual maximum
#       data: R-object for the dataset
#       rr: True if the study is on precipitations,
#           False if on temperatures

select.max.year <- function(data,rr){

   if (rr) st <- data$rr
   else    st <- data$temp
vect.max_c()
first.year <- min(data$year,na.rm=T)
last.year <- max(data$year,na.rm=T)
  for (y in first.year:last.year){
       aux <- st[data$year==y]
       aux <- aux[!is.nan(aux) & is.finite(aux)]
      if (length(aux)>0) vect.max_rbind(vect.max, max(aux,na.rm=T))
  }
max <- list(annual.year=vect.max)
max
}

# Seasonal and monthly maxima
#       data: R-object for the dataset
#       rr: True if the study is on precipitation,
#           False if on temperature

month.season.max <- function(data,rr){
 if (rr){
  data.choice <- data$rr
  variable <- 'precipitations'
  }
 else{
  data.choice <- data$temp
  variable <- 'temperatures'
  }
vect.max.wint_c()
vect.max.spr_c()
vect.max.sum_c()
vect.max.aut_c()
first.year <- min(data$year)
last.year <- max(data$year)

# Selection of the maximum by year and season
  for (y in first.year:last.year){
     # Winter
      aux.w <- data.choice[data$year==y & (data$month==1 | data$month==2| data$month==12)]
      aux.w <- aux.w[!is.nan(aux.w) & is.finite(aux.w)]
      if (length(aux.w)>0) vect.max.wint_rbind(vect.max.wint,max(aux.w,na.rm=T))

      # Spring
       aux.sp <- data.choice[data$year==y & (data$month==3 | data$month==4 | data$month==5)]
       aux.sp <- aux.sp[!is.nan(aux.sp) & is.finite(aux.sp)]
       if (length(aux.sp)>0) vect.max.spr_rbind(vect.max.spr,max(aux.sp,na.rm=T))

      # Summer
       aux.sum <- data.choice[data$year==y & (data$month==6 | data$month==7| data$month==8)]
```

```
      aux.sum <- aux.sum[!is.nan(aux.sum) & is.finite(aux.sum)]
      if (length(aux.sum)>0) vect.max.sum_rbind(vect.max.sum,max(aux.sum,na.rm=T))

     # Autumn
      aux.aut <- data.choice[data$year==y & (data$month==9 | data$month==10| data$month==11)]
      aux.aut <- aux.aut[!is.nan(aux.aut) & is.finite(aux.aut)]
      if (length(aux.aut)>0) vect.max.aut_rbind(vect.max.aut,max(aux.aut,na.rm=T))
  }
season.max <- rbind(vect.max.wint,vect.max.spr,vect.max.sum,vect.max.aut)

# Selection of the maximum by year and month
   # Monthly maximum
      vect.max.jan_c()
      vect.max.apr_c()
      vect.max.jul_c()
      vect.max.oct_c()

    # January
  for (y in first.year:last.year)
    vect.max.jan_rbind(vect.max.jan,
   max(data.choice[data$year==y & (data$month==1)],na.rm=T))
    # April
  for (y in first.year:last.year)
    vect.max.apr_rbind(vect.max.apr,
   max(data.choice[data$year==y & (data$month==4)],na.rm=T))
    # July
  for (y in first.year:last.year)
    vect.max.jul_rbind(vect.max.jul,
   max(data.choice[data$year==y & (data$month==7)],na.rm=T))
    # October
  for (y in first.year:last.year)
    vect.max.oct_rbind(vect.max.oct, max(data.choice[data$year==y & (data$month==10)],na.rm=T))

maxima <- list(variable=variable,winter=vect.max.wint,spring=vect.max.spr,
summer=vect.max.sum,autumn=vect.max.aut,jan=vect.max.jan,
apr=vect.max.apr,jul=vect.max.jul,oct=vect.max.oct,season=season.max)
maxima
}

# Seasonal and monthly minima
#       data: R-object for the dataset

temperature.min <- function(data){
data.choice <- data$temp
vect.min.wint_c()
vect.min.spr_c()
vect.min.sum_c()
vect.min.aut_c()
first.year <- min(data$year)
last.year <- max(data$year)
vect.min_c()
first.year <- min(data$year,na.rm=T)
last.year <- max(data$year,na.rm=T)

  for (y in first.year:last.year){
       aux <- data.choice[data$year==y]
       aux <- aux[!is.nan(aux) & is.finite(aux)]
      if (length(aux)>0) vect.min_rbind(vect.min, min(aux,na.rm=T))
  }

# Selection of the minimum by year and season
```

```
  for (y in first.year:last.year){
     # Winter
      aux.w <- data.choice[data$year==y & (data$month==1 | data$month==2| data$month==12)]
      aux.w <- aux.w[!is.nan(aux.w) & is.finite(aux.w)]
      if (length(aux.w)>0) vect.min.wint_rbind(vect.min.wint,min(aux.w,na.rm=T))

      # Spring
       aux.sp <- data.choice[data$year==y & (data$month==3 | data$month==4 | data$month==5)]
       aux.sp <- aux.sp[!is.nan(aux.sp) & is.finite(aux.sp)]
       if (length(aux.sp)>0) vect.min.spr_rbind(vect.min.spr,min(aux.sp,na.rm=T))

      # Summer
       aux.sum <- data.choice[data$year==y & (data$month==6 | data$month==7| data$month==8)]
       aux.sum <- aux.sum[!is.nan(aux.sum) & is.finite(aux.sum)]
       if (length(aux.sum)>0) vect.min.sum_rbind(vect.min.sum,min(aux.sum,na.rm=T))

      # Autumn
       aux.aut <- data.choice[data$year==y & (data$month==9 | data$month==10| data$month==11)]
       aux.aut <- aux.aut[!is.nan(aux.aut) & is.finite(aux.aut)]
       if (length(aux.aut)>0) vect.min.aut_rbind(vect.min.aut,min(aux.aut,na.rm=T))
  }
 season.min <- rbind(vect.min.wint,vect.min.spr,vect.min.sum,vect.min.aut)

 # Selection of the minimum by year and month
   # Monthly maximum
      vect.min.jan_c()
      vect.min.apr_c()
      vect.min.jul_c()
      vect.min.oct_c()

    # January
  for (y in first.year:last.year)
    vect.min.jan_rbind(vect.min.jan,
   max(data.choice[data$year==y & (data$month==1)],na.rm=T))
    # April
  for (y in first.year:last.year)
    vect.min.apr_rbind(vect.min.apr,
   min(data.choice[data$year==y & (data$month==4)],na.rm=T))
    # July
  for (y in first.year:last.year)
    vect.min.jul_rbind(vect.min.jul,
   min(data.choice[data$year==y & (data$month==7)],na.rm=T))
    # October
  for (y in first.year:last.year)
    vect.min.oct_rbind(vect.min.oct, max(data.choice[data$year==y & (data$month==10)],na.rm=T))

minima <- list(annual=abs(vect.min), winter=abs(vect.min.wint),spring=abs(vect.min.spr),
summer=abs(vect.min.sum),autumn=abs(vect.min.aut),jan=abs(vect.min.jan),
apr=abs(vect.min.apr),jul=abs(vect.min.jul),oct=abs(vect.min.oct),season=abs(season.min))
minima
}

# Diagnostic plots (PP-plot,QQ-plot, return level plot)
# for a gev distribution on an annual, sesaonal and monthly basis.
#       data: R-object for the dataset
#       rr: True if the study is on precipitations,
#           False if on temperatures

plot.gev <- function(data,rr){
  if (rr){
     variable <- 'precipitations'
```

```
     res <- rr.max.month(data)
     yea <- select.max.year(data,rr=T)
         }
  else{
     variable <- 'temperatures'
     res <- temp.max.month(data)
     yea <- select.max.year(data,rr=F)
      }

# Annual analysis
   print("GEV: Annual analysis")
   yea <- select.max.year(data,rr)
   M1 <- fgev(yea$annual.year)
   plot(M1)

# Seasonal analysis
   print("GEV: winter")
   M2 <- fgev(res$winter)
   plot(M2)
   print("GEV: spring")
   M3 <- fgev(res$spring)
   plot(M3)
   print("GEV: summer")
   M4 <- fgev(res$summer)
   plot(M4)
   print("GEV: autumn")
   M5 <- fgev(res$autumn)
   plot(M5)

# Monthly analysis
   print("GEV: january")
   M6 <- fgev(res$jan)
   plot(M6)
   print("GEV: april")
   M7 <- fgev(res$apr)
   plot(M7)
   print("GEV: july")
   M8 <- fgev(res$jul)
   plot(M8)
   print("GEV: october")
   M9 <- fgev(res$oct)
   plot(M9)
}

# Return period values for a gev distribution
# vect is a vector containing maxima or minima

gev.return.period <- function(vect){
M <- fgev(vect, std.err=F)
period <- c(5,10,25,50)
return.value <- c()
i <- 1
    for (t in period){
      return.value <- rbind(return.value,
M$estimate[1]+M$estimate[2]*((-1/log(1-1/t))^M$estimate[3]-1)/M$estimate[3])
      #print(paste('Gev return period values for',t,'years',round(return.value[i],2)))
         i <- i+1
       }
ret.value <- list(r.val5=round(return.value[1],2),r.val10=round(return.value[2],2),
                  r.val25=round(return.value[3],2),r.val50=round(return.value[4],2))
ret.value
```

```
}

# Plot GEV Distributions and
# comparison with GEV Distribution using Coles estimators (for data=oxford)
# example of call:
# data(oxford)
# try(oxford)

try <- function(x){
x11()
op <- par(mfrow = c(2, 2))
period <- c(5,10,15,20,25,30,50,100,200)
a <- fgev(x)
rl(a)
z.gev <- c()
z.coles <- c()
  for (t in period){
      y <- -1/log(1-1/t)
      z.gev <- cbind(z.gev,a$estimate[1] + a$estimate[2]*(y^a$estimate[3]-1)/a$estimate[3])
      z.coles <- cbind(z.coles, 83.8385453+4.2600510*(y^(-0.2872659)-1)/(-0.2872659))
  }
plot(period,z.gev,type="l",
main="gev-oxford",
sub=paste("mu,sigma & xi=",
round(a$estimate[1],2),round(a$estimate[2],2),round(a$estimate[3],2)))
points(period,z.coles,col="blue",lty=3,pch=20)
legend(100,91,
         c("rl","Coles"),
        pch=c(26,20),col=c("black","blue"),lwd=c(3,1),
         lty=c(1,0),bg="grey95")
}
obs.ctr.sc.gev.rl <- function(st.obs,st.ctr,st.sc){
yea.obs <- select.max.year(st.obs,rr=T)
yea.ctr <- select.max.year(st.ctr,rr=T)
yea.sc <- select.max.year(st.sc,rr=T)
gev.obs <- fgev(yea.obs$annual.year)
gev.ctr <- fgev(yea.ctr$annual.year)
gev.sc <- fgev(yea.sc$annual.year)

x11()
rl.obs <- rl(gev.obs,sub=paste("Observations"),plot=F)
rl.ctr <- rl(gev.ctr,sub=paste("CTR"),plot=F)
rl.sc <- rl(gev.sc,sub=paste("SC"),plot=F)
x11()
plot(c(min(-1/log(ppoints(gev.obs$tdata))),max(-1/log(ppoints(gev.obs$tdata)))),
     c(0,max(c(gev.obs$tdata,gev.ctr$tdata,gev.sc$tdata))),
     log="x",type="n",main="Return Level Plot",xlab="Return period (year)",
     ylab="Return level")
points(-1/log(ppoints(gev.obs$tdata)),sort(gev.obs$tdata),col="black",pch=20,cex=0.8)
points(-1/log(ppoints(gev.ctr$tdata)),sort(gev.ctr$tdata),col="red",pch=20,cex=0.8)
points(-1/log(ppoints(gev.sc$tdata)),sort(gev.sc$tdata),col="blue",pch=20,cex=0.8)
lines(rl.obs$x,rl.obs$y,lwd=3,col="black",lty=2)
lines(rl.ctr$x,rl.ctr$y,lwd=3,col="red",lty=2)
lines(rl.sc$x,rl.sc$y,lwd=3,col="blue",lty=1)
grid()
legend(2,50,
         c("Observations","Control period","Scenario period"),
        pch=c(20,20,20),col=c("black","red","blue"),lwd=c(3,3,3),
        lty=c(2,2,1),bg="grey95")
}
```

```
# Code to estimate the return values and return intervals assuming a
# General Pareto Distribution (GPD).
# By Alex Imbert & Rasmus Benestad, Oslo July 2002.
# source("estimator.R")

library(evd)

# Mean excess function
# support routine for gpd.mean.excess
# u: threshold for the dataset
# x: dataset

mean.excess <- function(u,x) {
   n <- length(u)
   mean.excess<-rep(NA,n)
   for (i in 1:n) {
     x.1 <- x[x>u[i]]-u[i]
     mean.excess[i] <- sum(x.1,na.rm=T)/length(x.1)
   }
   mean.excess
}

# Estimate the GPD parameters by mean excess function.
# Ref. Coles p. 33 & Patrik & Guiaihi, pt. 5
# support routine for gpd.fit.iter
# u: threshold for the dataset
# x: dataset

gpd.mean.excess <- function(u,x,lplot=T) {
   x.100 <- sort(x)
   n.x <- length(x)
   if (n.x > 100) x.100<-x[(n.x-99):n.x] else x.100<-x
   u.rng <- seq(min(x.100[!is.nan(x.100) & is.finite(x.100)],na.rm=T),max(x.100[!is.nan(x.100) & is.fi
   E <- mean.excess(u.rng,x)
   fit <- lm(E ~ u.rng)
   a0 <- fit$coefficients[1]   # the intercept
   a1 <- fit$coefficients[2]   # the slope
   xi <- a1/(1+a1)
   sigma <-a0*(1-xi)
   if (lplot) {
     plot(u.rng,E,pch=19,main="GPD Mean Excess",col="grey",
          xlab="Treshold",ylab="Mean exceedance",
          sub=paste("intercept & slope =",round(a0,2),round(a1,2),
                    "xi & sigma=",round(xi,2),round(sigma,2)))
     abline(fit,col="red",lwd=2)
     grid()
   }
   x <- x[x>u]-u
   h<-hist(x,breaks=25,plot=F)
   k <- xi * -1
   f <- exp(1/k*log(1-k*h$mids/sigma))
   gpd <- list(pdf=f,hist=h$density,
               cdf=cumsum(f)/sum(f),edf=cumsum(h$density)/sum(h$density),
               xi=xi,sigma=sigma,u=u,x=h$mids+u,
               rmse=sum((h$density-f)^2)/length(f),nexc=length(x))
}

# Moment estimator of the GPD parameters:
# Patrik & Guiaihi, pt. 5
# support routine for gpd.fit.iter
# u: threshold for the dataset
```

```
# x: dataset

gpd.moment <- function(u,x) {
 x <- x[x>u]
 mu <- mean(x,na.rm=T)
 k <- 0.5*(mu^2/var(x,na.rm=T) - 1)
 sigma <- 0.5*mu*(mu^2/var(x,na.rm=T)+1)
 h<-hist(x,breaks=25,plot=F)
 f <- exp(1/k*log(1-k*h$mids/sigma))
 gpd <- list(pdf=f,hist=h$density,
             cdf=cumsum(f)/sum(f),edf=cumsum(h$density)/sum(h$density),
             xi=k*-1,sigma=sigma,u=u,x=h$mids+u,
             rmse=sum((h$density-f)^2)/length(f),nexc=length(x))
 print(c(sigma,k*-1))
 gpd
}

# Calculate the GPD pdf.
gpd <- function(xi,sigma,x) {
 a <- 1+xi*x/sigma
 f <- 1- exp(-1/xi*log(a))
 f
}

# Estimate the GPD parameters using the moment estimator
# as a first guess.
# u: threshold for the dataset
# x: dataset
# example of call: gpd.fit.iter(30,rain.data)

gpd.fit.iter <- function(u,x,sig.lim=25,xi.lim=3,lmoment=F,
                         accuracy=0.0001,N.steps=500,
                         f.guess=NULL) {
  if (is.null(f.guess)) {
    if (lmoment) f.guess <- gpd.moment(u,x) else
                 f.guess <- gpd.mean.excess(u,x,lplot=F)
    sig.est <- NA
    xi.est <- NA
  }
  sigma.rng<-seq(max(0,f.guess$sigma-sig.lim),
                 f.guess$sigma+sig.lim,length=N.steps)
  xi.rng<-seq(f.guess$xi-xi.lim,f.guess$xi+xi.lim,length=N.steps)
  sig.step <- (max(sigma.rng)-min(sigma.rng))/N.steps
  xi.step <- (max(xi.rng)-min(xi.rng))/N.steps
#  print(paste("N.steps=",N.steps," xi step=",
#              round(xi.step,6),"sig step=",round(sig.step,6)))
  rmse <- 100
  w<-seq(0.1,1,length=length(f.guess$x))^2
  w<-w/sum(w)
  gpd.try <- gpd(f.guess$xi,f.guess$sigma,f.guess$x-u)
  for (sig.try in sigma.rng) {
    for (xi.try in xi.rng) {
      a.test <- min(1 + xi.try*f.guess$x/sig.try)
      if (a.test > 0) {
        gpd.try <- gpd(xi.try,sig.try,f.guess$x-u)
        rmse.try <- sum(w*(gpd.try-f.guess$edf)^2,na.rm=T)/length(gpd.try)
#print(paste("rmse.try",rmse.try, "rmse", rmse))
        if (rmse.try < rmse) {
          sig.est<-sig.try
          xi.est<-xi.try
          rmse <- rmse.try
```

```
          edf.est <- gpd.try
#           print(c(sig.est,xi.est,rmse))
        }
      }
    }
  }
  pdf.est<-c(diff(edf.est),0)
  gpd.fit.iter <- list(pdf=pdf.est,hist=f.guess$hist,
                       cdf=edf.est,edf=cumsum(f.guess$hist)/sum(f.guess$hist),
                       data=x,n=length(x),sig.step=sig.step,xi.step=xi.step,
                       xi=xi.est,sigma=sig.est,u=u,x=f.guess$x,
                       xi.f.guess=f.guess$xi,sigma.f.guess=f.guess$sigma,
                       pdf.f.guess=f.guess$pdf,nexc=f.guess$nexc,
                       rmse=sum((f.guess$density-pdf.est)^2)/length(pdf.est))
  gpd.fit.iter

}

# Estimate the GPD parameters iteratively, using the moment estimator
# as a first guess, then the iterative estimator recursively.
# carries out the nested iteration (a kind of recurtion)
# u: threshold for the dataset
# x: dataset
# example of call: gpd.fit(30,rain.data)

gpd.fit  <- function(u,x,lmoment=F,accuracy=0.0001,max.steps=500) {
  gpd.est <- gpd.fit.iter(u,x,lmoment=lmoment,accuracy=accuracy,N.steps=50)
  i.step <- 50
  rec.lev <- 1
  while ( ((gpd.est$sig.step > accuracy) | (gpd.est$xi.step > accuracy)) &
          (i.step < max.steps) ) {
    gpd.est <- gpd.fit.iter(u=u,x=x,sig.lim=gpd.est$sig.step,
                            xi.lim=gpd.est$xi.step,
                            lmoment=lmoment,accuracy=accuracy,
                            N.steps=50,f.guess=gpd.est)
    i.step <- i.step + 50
    rec.lev <- rec.lev + 1
#    print(paste("Recursion: istep=",i.step," Level=",rec.lev,
#                "xi=",gpd.est$xi,"sigma=",gpd.est$sigma,
#                "rmse=",gpd.est$rmse))
  }
  gpd.est
}

# plot GPD Distribution using parameters from mean excess function.
# x is the evd object got from gpd.fit.iter

plot.gpd <- function(x,gev.obj=NULL,lpeg=F,v.name="x") {
  xlim <- range(x$x[!is.na(x$x)])
  if (!lpeg) x11() else
  jpeg(file="gpd_cdf.jpg",width=480,height=480,pointsize=12,quality=100) # different plot formats
   plot(xlim,
       c(0,max(c(x$cdf,x$edf,x$cdf.f.guess),na.rm=T)),type="n",
       main="General Pareto Distribution & best fit",
       sub="gpd.R",xlab=v.name,ylab="c.d.f./e.d.f.")
  lines(x$x,x$cdf,lwd=3)
  points(x$x,x$edf,pch=20,col="red",lwd=2)
  lines(x$x,x$cdf.f.guess,col="blue",lty=2)
  grid()

# Add the cdf from the GEV to the plot
```

```
if (!is.null(gev.obj)) {
  x.gev <- sort(gev.obj$data)
  points(x.gev,ppoints(x.gev),pch=15,col="darkred",cex=0.5)
  x.pts <- seq(xlim[1],xlim[2],length=100)
  if (gev.obj$model=="gev") loc <- gev.obj$param["loc"]
  if (gev.obj$model=="gev.quantile") loc <- attributes(gev.obj)$loc
  pdf.gev <- dgev(x.pts,loc=loc,scale=gev.obj$param["scale"],
                  shape=gev.obj$param["shape"])
  i.nas <- is.na(pdf.gev)
  pdf.gev[i.nas] <- 0
  cdf.gev <- cumsum(pdf.gev)/sum(pdf.gev)
  pdf.gev[i.nas] <- NA
  cdf.gev[i.nas] <- NA
  lines(x.pts,cdf.gev,lty=2,col="blue")
  legend(min(x$x,na.rm=T),max(c(x$cdf,x$edf,x$cdf.f.guess),na.rm=T),
        c("Empirical (GPD)","GPD fit","Empirical (GEV)","GEV fit"),
         pch=c(20,26,15,26),col=c("red","black","darkred","blue"),
         lwd=c(0,3,0,2),lty=c(0,1,0,2),bg="grey95",cex=0.8)
 } else {
 legend(min(x$x,na.rm=T),max(c(x$cdf,x$edf,x$cdf.f.guess),na.rm=T),
       c("c.d.f.","e.d.f."),
      pch=c(26,20),col=c("black","red"),lwd=c(3,1),
       lty=c(1,0),bg="grey95",cex=0.8)
}
 if (ljpeg) dev.off()

# Compute the return level:
 ret.ints <- c(1.001,10^(seq(0,3,len=300))[-1])
 r.freq <- 1/ret.ints
 ret.lev.emp <-  rep(NA,length=length(r.freq))
 rl.gpd <-  rl.gpd(x)

 for (i in 1:length(r.freq)) ret.lev.emp[i] <- min(x$x[x$edf > r.freq[i]])
 x.val <- ppoints(sort(x$x))
 f.gpd <- 1/rl.gpd$return.interval
 if (!lpeg) x11() else
 jpeg(file="gpd_return.jpg",width=480,height=480,pointsize=12,quality=100)
 plot(-1/log(1-f.gpd),rl.gpd$return.value,lwd=3,log="x",type="l",
      main=paste("General Pareto Distribution & Return Values for",v.name),
      sub="gpd.R",ylab="Return value",xlab="Return interval")
 points(-1/log(r.freq),ret.lev.emp,pch=20,col="red")
 if (!is.null(gev.obj)) {
   if (gev.obj$model=="gev") loc <- gev.obj$param["loc"]
   if (gev.obj$model=="gev.quantile") loc <- attributes(gev.obj)$loc
   rl.gev <- qgev(r.freq,loc=loc,scale=gev.obj$param["scale"],
                  shape=gev.obj$param["shape"],lower.tail=F)
   emp.freq <- ppoints(gev.obj$tdata)
   lines(-1/log(1-r.freq),rl.gev,lty=2,lwd=2,col="blue")
   points(-1/log(emp.freq),sort(gev.obj$tdata),
          pch=15,col="darkred",cex=0.5)
   grid()
   legend(min(-1/log(x.val),na.rm=T),max(rl.gpd$return.value,na.rm=T),
          c("Empirical (GPD)","GPD fit","Empirical (GEV)","GEV fit"),
          pch=c(20,26,15,26),col=c("red","black","darkred","blue"),
         lwd=c(0,3,0,2),lty=c(0,1,0,2),bg="grey95",cex=0.8)
  } else {
   legend(min(-1/log(r.freq),na.rm=T),max(rl.gpd$return.value,na.rm=T),
          c("Points","GPD"),
          pch=c(20,26),col=c("red","black"),lwd=c(0,3),
          lty=c(0,1),bg="grey95",cex=0.8)
  }
```

```
  if (ljpeg) dev.off()
}

# add the GPD Distribution using Coles GPD parameters (for dataset=rain.data).
add.coles<-function(x) {
  lines(x$x,gpd(0.184499,7.4402693,x$x),col="grey",lwd=2)
}

# Plot the probability plot for GPD Distribution
pp.gpd<- function(x) {
  y.1 <- cumsum(x$pdf)/sum(x$pdf)
  y.2 <- cumsum(x$hist)/sum(x$hist)
  plot(c(0,1),c(0,1),type="n",main="PP-plot for GPD",
       xlab="model",ylab="empirical")
  lines(c(0,1),c(0,1))
  points(y.1,y.2,pch=20)
}

# E.d.f and c.d.f for GPD Distribution
rl.gpd <- function(x) {
  ret.val <-seq(min(x$x),3*max(x$x),length=100)
  X<- ret.val-x$u
  p <- gpd(x$xi,x$sigma,X)
  ret.int <- 1/(1-p)
  ret.val99<-approx(p,ret.val,0.99)
  ret.int99<- min(ret.int[ret.val99> ret.val],na.rm=T)
  edf.gpd <- cumsum(x$hist)/sum(x$hist)
  cdf.gpd <- cumsum(x$pdf)/sum(x$pdf)
  lev99.edf<-x$x[edf.gpd >= 0.99]
  lev99.cdf<-x$x[cdf.gpd >= 0.99]
  rr.gpd <- list(return.value=ret.val,return.interval=ret.int,ln.p=log(abs(p)),
                 ret.val99=ret.val99,ret.int99=ret.int99,lev99.edf=lev99.edf,
                 lev99.cdf=lev99.cdf)
  rr.gpd
}

# determine the return level values for different return periods
# x is the evd object (from gpd.fit.iter)

extreme.quantiles <- function(x){
rl <- rl.gpd(x)
period <- c(5,10,25,50,100)
x.p <- c()
x.p.f.guess <- c()
x.p.f.coles <- c()

for (t in period){
  x.p <- cbind( x.p, min(rl$return.value[rl$return.interval > t],na.rm=T) )
  x.p.f.guess <- cbind(x.p.f.guess,
                      ((1/t)^(-1*x$xi.f.guess)-1/x$sigma.f.guess)*x$sigma.f.guess/x$xi.f.guess)
#  x.p.f.coles <- cbind(x.p.f.coles,
#                       ((1/t)^(-1*0.184499)-1/7.4402693)*7.4402693/0.184499)
# used to make the comparison with Coles estimators for the dataset=rain.data
}
print(paste("sigma=",x$sigma, "xi=",x$xi,"nbr exceedances",x$nexc))
#print(paste(" x.p=",round(x.p,2), " x.p.coles=",round(x.p.f.coles,2),
#            " x.p.f.guess=",round(x.p.f.guess,2)))
#print(paste(" x.p=",round(x.p,2),
#            " x.p.f.guess=",round(x.p.f.guess,2)))

r.values <- list(r.p5=round(x.p[1],2),r.p10=round(x.p[2],2),
```

```
                r.p25=round(x.p[3],2),r.p50=round(x.p[4],2),
                r.p100=round(x.p[5],2))
print(paste("r.values",r.values))
r.values
}

# GPD Parameters against threshold
plot.u <- function(data){
sigma.u <- c()
xi.u <- c()
x.sort <- sort(data)
plot(c(min(x.sort)+1,max(x.sort)),c(-5,30),type="n",main="$\xi$ parameter against threshold")
grid()
u.rng <- seq(min(x.sort)+1,0.99*max(x.sort),length=50)
   for(u in u.rng){
    sigma.u <- c(sigma.u,gpd.fit(u,data)$sigma)
    xi.u <- c(xi.u,gpd.fit(u,data)$xi)
    points(u,sigma.u[length(sigma.u)],pch=19)
    points(u,xi.u[length(xi.u)],pch=20)
  }
#op <- par(mfrow = c(2, 1))
lines(u.rng,sigma.u,type="l",col="blue")
points(u.rng,sigma.u,pch=19)
x11()
plot(c(min(x.sort)+1,max(x.sort)),c(-1,5),type="n",main="$\sigma$ parameter against threshold")
lines(u.rng,xi.u,type="l",col="red")
points(u.rng,xi.u,pch=20)
grid()
}

# Compare two different distributions (histograms)
# Example of call to compare GPD and GEV distributions.
# u <- quantile(st.obs$rr,0.99)  (definition of the threshold)
# y.exc <- st.obs$rr - u
# y.exc[y.exc<0] <- NA
# y.exc <- y.exc + u
# y.exc <- y.exc[!is.na(y.exc)]  (only the values above the threshols are kept)
# vect.max <- select.max.year(st.obs,rr=T)  (annual maximum)
# cmp.distr(y.exc,vect.max$annual.year)

cmp.distr <- function(x.1,x.2) {
  rng <- range(c(x.1,x.2))
  x.vec <- seq(rng[1],rng[2],length=25)
  h.1 <- hist(x.1,breaks=x.vec,plot=F)
  h.2 <- hist(x.2,breaks=x.vec,plot=F)
  plot(h.1$mids,h.1$density,lwd=3,type="l",col="blue")
  lines(h.1$mids,h.2$density,lwd=3,col="red")
  grid()
}
```

```
# This code permits to realize the complete statistical analysis
# and to write the results in a table.
# If the program is run for the first time, it's necessary to first write
# the files in the appropriate format with:
# 1) dumb <- fix.files()
# 2) dumb2 <- change.0()
# example of call: write.results()

# source("select_analysis.R")

source("fix.files.R")
source("check.data.R")
source("find.substr.R")
source("avail.files.R")
source("extreme_value.R")
source("rain_study.R")
source("estimator.R")

# obs: dataset for observations
# ctr: dataset for control period
# sc: dataset for the scenario.

# Realizes the complete analysis for one station.
# rr=T is the study is on precipitations, =F if it's on temperatures
# conf is the confidence level for wilcox.test
# i.data is the number of the station taking into account
# if mean.test=T, tests are made on means, variances and percentiles.
# if mean.test=F, tests are made on gev return periods.

analysis <- function(rr,conf,i.data,mean.test){
source("check.data.R")
# import into R oberved, controled periods and scenario for the same station.
read.st <- read.files(i.data)
st.obs <- read.st$obs
st.ctr <- read.st$ctr
st.sc <- read.st$sc

  if (mean.test==T){

# Mean values: month, season, annual for the three different files.
M1 <- mean.study(st.obs,rr)
M2 <- mean.study(st.ctr,rr)
M3 <- mean.study(st.sc,rr)

# Variances: month, season, annual for the three different files.
V1 <- var.study(st.obs,rr)
V2 <- var.study(st.ctr,rr)
V3 <- var.study(st.sc,rr)

# Comparison between mean values or variances
 if (rr==F){
    comp.mean <- student.comparison(st.obs,st.ctr,0.95)
    comp.model <- student.comparison(st.sc,st.ctr,0.95)
    comp.var <- fisher.comparison(st.obs,st.ctr,0.95)
    comp.var.mod <- fisher.comparison(st.sc,st.ctr,0.95)
  }
 else{
    comp.mean <- wilcoxon.comparison(st.obs,st.ctr,rr,0.95,var=F)
    comp.model <- wilcoxon.comparison(st.sc,st.ctr,rr,0.95,var=F)
 }
```

```
# Percentiles
obs0.1 <- percentile(st.obs,0.0001,rr)
obs1 <- percentile(st.obs,0.01,rr)
obs5 <- percentile(st.obs,0.05,rr)
obs95 <- percentile(st.obs,0.95,rr)
obs99 <- percentile(st.obs,0.99,rr)
obs991 <- percentile(st.obs,0.991,rr)
ctr0.1 <- percentile(st.ctr,0.0001,rr)
ctr1 <- percentile(st.ctr,0.01,rr)
ctr5 <- percentile(st.ctr,0.05,rr)
ctr95 <- percentile(st.ctr,0.95,rr)
ctr99 <- percentile(st.ctr,0.99,rr)
ctr991 <- percentile(st.ctr,0.991,rr)
sc0.1 <- percentile(st.sc,0.0001,rr)
sc1 <- percentile(st.sc,0.01,rr)
sc5 <- percentile(st.sc,0.05,rr)
sc95 <- percentile(st.sc,0.95,rr)
sc99 <- percentile(st.sc,0.99,rr)
sc991 <- percentile(st.sc,0.991,rr)

means.res <- list(
                m.obs.annual=M1$annual,m.obs.spring=M1$spring,m.obs.summer=M1$summer,
                m.obs.autumn=M1$autumn,m.obs.winter=M1$winter,
                m.ctr.annual=M2$annual,m.ctr.spring=M2$spring,m.ctr.summer=M2$summer,
                m.ctr.autumn=M2$autumn,m.ctr.winter=M2$winter,
                m.sc.annual=M3$annual,m.sc.spring=M3$spring,m.sc.summer=M3$summer,
                m.sc.autumn=M3$autumn,m.sc.winter=M3$winter,
                c.m.annual=comp.mean$annual,c.m.spring=comp.mean$spring,c.m.summer=comp.mean$summer,
                c.m.autumn=comp.mean$autumn,c.m.winter=comp.mean$winter,
                c.annual.sc=comp.model$annual,c.spring.sc=comp.model$spring,
                c.summer.sc=comp.model$summer,c.autumn.sc=comp.model$autumn,
                c.winter.sc=comp.model$winter,
#               c.v.annual=comp.var$annual,c.v.spring=comp.var$spring,
#               c.v.summer=comp.var$summer,c.v.autumn=comp.var$autumn,
#               c.v.winter=comp.var$winter,
#               c.v.annual.mod=comp.var.mod$annual,c.v.spring.mod=comp.var.mod$spring,
#               c.v.summer.mod=comp.var.mod$summer,c.v.autumn.mod=comp.var.mod$autumn,
#               c.v.winter.mod=comp.var.mod$winter,
                v.obs.annual=V1$annual,v.obs.spring=V1$spring,v.obs.summer=V1$summer,
                v.obs.autumn=V1$autumn,v.obs.winter=V1$winter,
                v.ctr.annual=V2$annual,v.ctr.spring=V2$spring,v.ctr.summer=V2$summer,
                v.ctr.autumn=V2$autumn,v.ctr.winter=V2$winter,
                v.sc.annual=V3$annual,v.sc.spring=V3$spring,v.sc.summer=V3$summer,
                v.sc.autumn=V3$autumn,v.sc.winter=V3$winter,
                obs0.1.ann=obs0.1$annual,obs0.1.spr=obs0.1$spring,obs0.1.summ=obs0.1$summer,
                obs0.1.aut=obs0.1$autumn,obs0.1.wint=obs0.1$winter,
                obs1.ann=obs1$annual,obs1.spr=obs1$spring,obs1.summ=obs1$summer,
                obs1.aut=obs1$autumn,obs1.wint=obs1$winter,
                obs5.ann=obs5$annual,obs5.spr=obs5$spring,obs5.summ=obs5$summer,
                obs5.aut=obs5$autumn,obs5.wint=obs5$winter,
                obs95.ann=obs95$annual,obs95.spr=obs95$spring,obs95.summ=obs95$summer,
                obs95.aut=obs95$autumn,obs95.wint=obs95$winter,
                obs99.ann=obs99$annual,obs99.spr=obs99$spring,obs99.summ=obs99$summer,
                obs99.aut=obs99$autumn,obs99.wint=obs99$winter,
                obs991.ann=obs991$annual,obs991.spr=obs991$spring,obs991.summ=obs991$summer,
                obs991.aut=obs991$autumn,obs991.wint=obs991$winter,
                ctr0.1.ann=ctr0.1$annual,ctr0.1.spr=ctr0.1$spring,ctr0.1.summ=ctr0.1$summer,
                ctr0.1.aut=ctr0.1$autumn,ctr0.1.wint=ctr0.1$winter,
                ctr1.ann=ctr1$annual,ctr1.spr=ctr1$spring,ctr1.summ=ctr1$summer,
                ctr1.aut=ctr1$autumn,ctr1.wint=ctr1$winter,
                ctr5.ann=ctr5$annual,ctr5.spr=ctr5$spring,ctr5.summ=ctr5$summer,
```

```
                ctr5.aut=ctr5$autumn,ctr5.wint=ctr5$winter,
                ctr95.ann=ctr95$annual,ctr95.spr=ctr95$spring,ctr95.summ=ctr95$summer,
                ctr95.aut=ctr95$autumn,ctr95.wint=ctr95$winter,
                ctr99.ann=ctr99$annual,ctr99.spr=ctr99$spring,ctr99.summ=ctr99$summer,
                ctr99.aut=ctr99$autumn,ctr99.wint=ctr99$winter,
                ctr991.ann=ctr991$annual,ctr991.spr=ctr991$spring,ctr991.summ=ctr991$summer,
                ctr991.aut=ctr991$autumn,ctr991.wint=ctr991$winter,
                sc0.1.ann=sc0.1$annual,sc0.1.spr=sc0.1$spring,sc0.1.summ=sc0.1$summer,
                sc0.1.aut=sc0.1$autumn,sc0.1.wint=sc0.1$winter,
                sc1.ann=sc1$annual,sc1.spr=sc1$spring,sc1.summ=sc1$summer,
                sc1.aut=sc1$autumn,sc1.wint=sc1$winter,
                sc5.ann=sc5$annual,sc5.spr=sc5$spring,sc5.summ=sc5$summer,
                sc5.aut=sc5$autumn,sc5.wint=sc5$winter,
                sc95.ann=sc95$annual,sc95.spr=sc95$spring,sc95.summ=sc95$summer,
                sc95.aut=sc95$autumn,sc95.wint=sc95$winter,
                sc99.ann=sc99$annual,sc99.spr=sc99$spring,sc99.summ=sc99$summer,
                sc99.aut=sc99$autumn,sc99.wint=sc99$winter,
                sc991.ann=sc991$annual,sc991.spr=sc991$spring,sc991.summ=sc991$summer,
                sc991.aut=sc991$autumn,sc991.wint=sc991$winter)
 return(means.res)
  }
  else{
         if (rr){
  # Return periods using GPD method.
         obs90 <- percentile(st.obs,0.90,rr=T)
         u <- obs90$annual
         gpd.obs <- gpd.fit(u,st.obs$rr)
         gpd.ctr <- gpd.fit(u,st.ctr$rr)
         gpd.sc  <- gpd.fit(u,st.sc$rr)
         qu.obs <- extreme.quantiles(gpd.obs)
         qu.ctr <- extreme.quantiles(gpd.ctr)
         qu.sc <- extreme.quantiles(gpd.sc)

 # Return periods using GEV method.
         yea.obs <- select.max.year(st.obs,rr=T)
         yea.ctr <- select.max.year(st.ctr,rr=T)
         yea.sc <- select.max.year(st.sc,rr=T)
         rp.obs <- gev.return.period(yea.obs$annual.year)
         rp.ctr <- gev.return.period(yea.ctr$annual.year)
         rp.sc <- gev.return.period(yea.sc$annual.year)

    return.periods <- list(
                     rl.gev5.obs=rp.obs$r.val5,rl.gev10.obs=rp.obs$r.val10,
     rl.gev25.obs=rp.obs$r.val25,rl.gev50.obs=rp.obs$r.val50,
                     rl.gev5.ctr=rp.ctr$r.val5,rl.gev10.ctr=rp.ctr$r.val10,
                     rl.gev25.ctr=rp.ctr$r.val25,rl.gev50.ctr=rp.ctr$r.val50,
                     rl.gev5.sc=rp.sc$r.val5,rl.gev10.sc=rp.sc$r.val10,
                     rl.gev25.sc=rp.sc$r.val25,rl.gev50.sc=rp.sc$r.val50,
                     rl.gpd5.obs=qu.obs$r.p5,rl.gpd10.obs=qu.obs$r.p10,
    rl.gpd25.obs=qu.obs$r.p25,rl.gpd50.obs=qu.obs$r.p50,
                     rl.gpd5.ctr=qu.ctr$r.p5,rl.gpd10.ctr=qu.ctr$r.p10,
                     rl.gpd25.ctr=qu.ctr$r.p25,rl.gpd50.ctr=qu.ctr$r.p50,
                     rl.gpd5.sc=qu.sc$r.p5,rl.gpd10.sc=qu.sc$r.p10,
                     rl.gpd25.sc=qu.sc$r.p25,rl.gpd50.sc=qu.sc$r.p50)
  return(return.periods)
  }
  else{

# Return periods using GEV method.
  # With miminum temperatures
   min.year.obs <- temperature.min(st.obs)
```

```
    min.year.ctr <- temperature.min(st.ctr)
    min.year.sc <- temperature.min(st.sc)
    rp.min.obs <- gev.return.period(min.year.obs$annual)
    rp.min.ctr <- gev.return.period(min.year.ctr$annual)
    rp.min.sc <- gev.return.period(min.year.sc$annual)

  # With maximum temperatures
    max.yea.obs <- select.max.year(st.obs,rr=F)
    max.yea.ctr <- select.max.year(st.ctr,rr=F)
    max.yea.sc <- select.max.year(st.sc,rr=F)
    rp.obs <- gev.return.period(max.yea.obs$annual.year)
    rp.ctr <- gev.return.period(max.yea.ctr$annual.year)
    rp.sc <- gev.return.period(max.yea.sc$annual.year)

# Return periods using GPD method.
    # With miminum temperatures
    #u.min.obs <- quantile(-1*st.obs$temp[!is.na(st.obs$temp)],0.9)
    #u.min.mod <- quantile(-1*st.ctr$temp[!is.na(st.ctr$temp)],0.9)
    #u.min.sc <- quantile(-1*st.sc$temp[!is.na(st.sc$temp)],0.9)
    #gpd.obs <- gpd.fit(u.min.obs,-1*st.obs$temp)
    #gpd.ctr <- gpd.fit(u.min.mod,-1*st.ctr$temp)
    #gpd.sc  <- gpd.fit(u.min.mod,-1*st.sc$temp)

    # With maximum temperatures
    obs90 <- percentile(st.obs,0.90,rr=F)
    mod90 <- percentile(st.ctr,0.90,rr=F)
    sc90 <- percentile(st.sc,0.90,rr=F)
    u.obs <- obs90$annual
    u.mod <- mod90$annual
    gpd.obs <- gpd.fit(u.obs,st.obs$temp)
    gpd.ctr <- gpd.fit(u.mod,st.ctr$temp)
    gpd.sc  <- gpd.fit(u.mod,st.sc$temp)
    qu.obs <- extreme.quantiles(gpd.obs)
    qu.ctr <- extreme.quantiles(gpd.ctr)
    qu.sc <- extreme.quantiles(gpd.sc)

    return.periods <- list(
                      rl.gev5.obs=rp.obs$r.val5,rl.gev10.obs=rp.obs$r.val10,
      rl.gev25.obs=rp.obs$r.val25,rl.gev50.obs=rp.obs$r.val50,
                      rl.gev5.ctr=rp.ctr$r.val5,rl.gev10.ctr=rp.ctr$r.val10,
                      rl.gev25.ctr=rp.ctr$r.val25,rl.gev50.ctr=rp.ctr$r.val50,
                      rl.gev5.sc=rp.sc$r.val5,rl.gev10.sc=rp.sc$r.val10,
                      rl.gev25.sc=rp.sc$r.val25,rl.gev50.sc=rp.sc$r.val50,
                      rl.gev5.minobs=rp.min.obs$r.val5,rl.gev10.minobs=rp.min.obs$r.val10,
      rl.gev25.minobs=rp.min.obs$r.val25,rl.gev50.minobs=rp.min.obs$r.val50,
                      rl.gev5.minctr=rp.min.ctr$r.val5,rl.gev10.minctr=rp.min.ctr$r.val10,
                      rl.gev25.minctr=rp.min.ctr$r.val25,rl.gev50.minctr=rp.min.ctr$r.val50,
                      rl.gev5.minsc=rp.min.sc$r.val5,rl.gev10.minsc=rp.min.sc$r.val10,
                      rl.gev25.minsc=rp.min.sc$r.val25,rl.gev50.minsc=rp.min.sc$r.val50,
                      rl.gpd5.obs=qu.obs$r.p5,rl.gpd10.obs=qu.obs$r.p10,
      rl.gpd25.obs=qu.obs$r.p25,rl.gpd50.obs=qu.obs$r.p50,
                      rl.gpd5.ctr=qu.ctr$r.p5,rl.gpd10.ctr=qu.ctr$r.p10,
                      rl.gpd25.ctr=qu.ctr$r.p25,rl.gpd50.ctr=qu.ctr$r.p50,
                      rl.gpd5.sc=qu.sc$r.p5,rl.gpd10.sc=qu.sc$r.p10,
                      rl.gpd25.sc=qu.sc$r.p25,rl.gpd50.sc=qu.sc$r.p50)

  return(return.periods)
        }
    }
}
```

```
# Function to write results in a data.frame
# obs: dataset for observations
# ctr: dataset for control period
# sc: dataset for the scenario.
# example of call: write.results()

write.results <- function(conf=0.95,rr=T,mean.test=T){
source("fix.files.R")
setwd("obs80-99-txt")
names.obs <- avail.files(pattern.1="obs2.ok")
setwd("..")
vect.files <- look.for.files()
   # loop to realize the statistical study on all the files
   for (i.names in vect.files){
      res <- analysis(rr,conf,i.names, mean.test)

          if(mean.test==T){
   annual.results <- data.frame(
           station=i.names,
           mean=res$m.obs.annual,variance=res$v.obs.annual,
           mean=res$m.ctr.annual,variance=res$v.ctr.annual,
           mean=res$m.sc.annual,variance=res$v.sc.annual,
           diff.mean=res$c.m.annual, diff.model=res$c.annual.sc)
          # ,diff.var=res$c.v.annual,diff.var.model=res$c.v.annual.mod)

   spring.results <- data.frame(
           station=i.names,
           mean=res$m.obs.spring,variance=res$v.obs.spring,
           mean=res$m.ctr.spring,variance=res$v.ctr.spring,
           mean=res$m.sc.spring,variance=res$v.sc.spring,
           diff.mean=res$c.m.spring, diff.model=res$c.spring.sc)
          #,diff.var=res$c.v.spring,diff.var.model=res$c.v.spring.mod)

   summer.results <- data.frame(
           station=i.names,
           mean=res$m.obs.summer,variance=res$v.obs.summer,
           mean=res$m.ctr.summer,variance=res$v.ctr.summer,
           mean=res$m.sc.summer,variance=res$v.sc.summer,
           diff.mean=res$c.m.summer, diff.model=res$c.summer.sc)
           #,diff.var=res$c.v.summer,diff.var.model=res$c.v.summer.mod)

   autumn.results <- data.frame(
           station=i.names,
           mean=res$m.obs.autumn,variance=res$v.obs.autumn,
           mean=res$m.ctr.autumn,variance=res$v.ctr.autumn,
           mean=res$m.sc.autumn,variance=res$v.sc.autumn,
           diff.mean=res$c.m.autumn, diff.model=res$c.autumn.sc)
           #,diff.var=res$c.v.autumn,diff.var.model=res$c.v.autumn.mod)

   winter.results <- data.frame(
           station=i.names,
           mean=res$m.obs.winter,variance=res$v.obs.winter,
           mean=res$m.ctr.winter,variance=res$v.ctr.winter,
           mean=res$m.sc.winter,variance=res$v.sc.winter,
           diff.mean=res$c.m.winter, diff.model=res$c.winter.sc)
           #,diff.var=res$c.v.winter,diff.var.model=res$c.v.winter.mod)

   percentiles.annual <- data.frame(
           station=i.names,
           p.01=res$obs0.1.ann,p.01=res$ctr0.1.ann,p.01=res$sc0.1.ann,
           p.1=res$obs1.ann,p.1=res$ctr1.ann,p.1=res$sc1.ann,
```

```
          p.5=res$obs5.ann,p.5=res$ctr5.ann,p.5=res$sc5.ann,
          p.95=res$obs95.ann,p.95=res$ctr95.ann,p.95=res$sc95.ann,
          p.99=res$obs99.ann,p.99=res$ctr99.ann,p.99=res$sc99.ann,
          p.991=res$obs991.ann,p.991=res$ctr991.ann,p.991=res$sc991.ann)

 percentiles.spring <- data.frame(
          station=i.names,
          p.01=res$obs0.1.spr,p.01=res$ctr0.1.spr,p.01=res$sc0.1.spr,
          p.1=res$obs1.spr,p.1=res$ctr1.spr,p.1=res$sc1.spr,
          p.5=res$obs5.spr,p.5=res$ctr5.spr,p.5=res$sc5.spr,
          p.95=res$obs95.spr,p.95=res$ctr95.spr,p.95=res$sc95.spr,
          p.99=res$obs99.spr,p.99=res$ctr99.spr,p.99=res$sc99.spr,
          p.991=res$obs991.spr,p.991=res$ctr991.spr,p.991=res$sc991.spr)

 percentiles.summer <- data.frame(
          station=i.names,
          p.01=res$obs0.1.summ,p.01=res$ctr0.1.summ,p.01=res$sc0.1.summ,
          p.1=res$obs1.summ,p.1=res$ctr1.summ,p.1=res$sc1.summ,
          p.5=res$obs5.summ,p.5=res$ctr5.summ,p.5=res$sc5.summ,
          p.95=res$obs95.summ,p.95=res$ctr95.summ,p.95=res$sc95.summ,
          p.99=res$obs99.summ,p.99=res$ctr99.summ,p.99=res$sc99.summ,
          p.991=res$obs991.summ,p.991=res$ctr991.summ,p.991=res$sc991.summ)

 percentiles.autumn <- data.frame(
          station=i.names,
          p.01=res$obs0.1.aut,p.01=res$ctr0.1.aut,p.01=res$sc0.1.aut,
          p.1=res$obs1.aut,p.1=res$ctr1.aut,p.1=res$sc1.aut,
          p.5=res$obs5.aut,p.5=res$ctr5.aut,p.5=res$sc5.aut,
          p.95=res$obs95.aut,p.95=res$ctr95.aut,p.95=res$sc95.aut,
          p.99=res$obs99.aut,p.99=res$ctr99.aut,p.99=res$sc99.aut,
          p.991=res$obs991.aut,p.991=res$ctr991.aut,p.991=res$sc991.aut)

 percentiles.winter <- data.frame(
          station=i.names,
          p.01=res$obs0.1.wint,p.01=res$ctr0.1.wint,p.01=res$sc0.1.wint,
          p.1=res$obs1.wint,p.1=res$ctr1.wint,p.1=res$sc1.wint,
          p.5=res$obs5.wint,p.5=res$ctr5.wint,p.5=res$sc5.wint,
          p.95=res$obs95.wint,p.95=res$ctr95.wint,p.95=res$sc95.wint,
          p.99=res$obs99.wint,p.99=res$ctr99.wint,p.99=res$sc99.wint,
          p.991=res$obs991.wint,p.991=res$ctr991.wint,p.991=res$sc991.wint)
#write.table(annual.results,file='/home/alexi/scenario/two.sided.year2_temp.asc',append=TRUE,quote=F
#write.table(spring.results,file='/home/alexi/alexi/scenario/two.sided.spring2_temp.asc',append=TRUE
#write.table(summer.results,file='/home/alexi/alexi/scenario/two.sided.summer2_temp.asc',append=TRUE
#write.table(autumn.results,file='/home/alexi/alexi/scenario/two.sided.autumn2_temp.asc',append=TRUE
#write.table(winter.results,file='/home/alexi/alexi/scenario/two.sided.winter2_temp.asc',append=TRUE
write.table(percentiles.annual,file='/home/alexi/alexi/scenario/percent.annual_rr.asc',append=TRUE,q
#write.table(percentiles.spring,file='/home/alexi/alexi/scenario/percent.spring_temp.asc',append=TRU
#write.table(percentiles.summer,file='/home/alexi/alexi/scenario/percent.summer_temp.asc',append=TRU
#write.table(percentiles.autumn,file='/home/alexi/scenario/alexi/percent.autumn_temp.asc',append=TRU
#write.table(percentiles.winter,file='/home/alexi/alexi/scenario/percent.winter_temp.asc',append=TRU

#write.table(comparison.res,file='/home/alexi/alexi/scenario/sign_obs_ctr_varfinal_rr.asc',append=TR
#write.table(comparison.res,file='/home/alexi/alexi/scenario/sign_obs_sc_rr.asc',append=TRUE,quote=F
      }
      else{
        if(rr==F){
        extreme.min.gev.res <- data.frame(
                stat=i.names,
                R.PGEV.5=res$rl.gev5.minobs, R.PGEV.5=res$rl.gev5.minctr, R.PGEV.5=res$rl.gev5.mins
                R.PGEV.10=res$rl.gev10.minobs, R.PGEV.10=res$rl.gev10.minctr, R.PGEV.10=res$rl.gev1
                R.PGEV.25=res$rl.gev25.minobs, R.PGEV.25=res$rl.gev25.minctr, R.PGEV.25=res$rl.gev2
```

```
                R.PGEV.50=res$rl.gev50.minobs,R.PGEV.50=res$rl.gev50.minctr,R.PGEV.50=res$rl.gev50.mir
format(extreme.min.gev.res, justify="left")
write.table(extreme.min.gev.res,file='/home/alexi/alexi/scenario/gevmin_temp2.asc',append=TRUE,quote=F,
     }
       extreme.max.gev.res <- data.frame(
                stat=i.names,
                R.PGEV.5=res$rl.gev5.obs, R.PGEV.5=res$rl.gev5.ctr, R.PGEV.5=res$rl.gev5.sc,
                R.PGEV.10=res$rl.gev10.obs, R.PGEV.10=res$rl.gev10.ctr, R.PGEV.10=res$rl.gev10.sc,
                R.PGEV.25=res$rl.gev25.obs, R.PGEV.25=res$rl.gev25.ctr, R.PGEV.25=res$rl.gev25.sc,
                R.PGEV.50=res$rl.gev50.obs,R.PGEV.50=res$rl.gev50.ctr,R.PGEV.50=res$rl.gev50.sc)

       extreme.gpd.res <- data.frame(
                stat=i.names,
                 R.PGPD.5=res$rl.gpd5.obs,R.PGPD.5=res$rl.gpd5.ctr,R.PGPD.5=res$rl.gpd5.sc,
                 R.PGPD.10=res$rl.gpd10.obs,R.PGPD.10=res$rl.gpd10.ctr,R.PGPD.10=res$rl.gpd10.sc,
                 R.PGPD.25=res$rl.gpd25.obs,R.PGPD.25=res$rl.gpd25.ctr,R.PGPD.25=res$rl.gpd25.sc,
                 R.PGPD.50=res$rl.gpd50.obs,R.PGPD.50=res$rl.gpd50.ctr,R.PGPD.50=res$rl.gpd50.sc)

format(extreme.max.gev.res, justify="left")
write.table(extreme.max.gev.res,file='/home/alexi/alexi/scenario/gevmax_rr2.asc',append=TRUE,quote=F,se
format(extreme.gpd.res, justify="left")
write.table(extreme.gpd.res,file='/home/alexi/alexi/scenario/try.asc',append=TRUE,quote=F,sep=" & ",eol
write.table(extreme.gpd.res,file='/home/alexi/alexi/scenario/gpd2_rr_10_read.asc',append=TRUE,quote=F,s
}
   }
}

# This routine permits to determine wether or not observations, model
# and scenario can be found in the corresponding directories for the same station.
# This will permit to make comparisons between the three files.
look.for.files <- function(){
source("avail.files.R")
names.obs <- avail.files("obs80-99-txt",".ok","-obs2")
names.ctr <- avail.files("present-txt",".txt","-present")
names.sc <- avail.files("scenario-txt",".txt","-scenario")

     # selection of an observed dataset and lookink for the corresponding
     # controlled and scenario periods.
vect.found <- c()
   for (i.names in 1:length(names.obs)){
     found <- F
     k <- 1
     while (k <length(names.obs) & found==F){
      # selection of the number of the station
     if (substr(names.obs[i.names],5,5)=="-")
         n.obs <- substr(names.obs[i.names],1,4)
       else{
          if (substr(names.obs[i.names],4,4)=="-")
             n.obs <- substr(names.obs[i.names],1,3)

          else n.obs <- substr(names.obs[i.names],1,5)
           }
 i <- 1
 found.ctr <- F
 found.sc <- F
    # looking for the control period corresponding to the same station.
  while(found.ctr==F & i<length(names.ctr)){
      if (substr(names.ctr[i],5,5)=="-")
         n.ctr <- substr(names.ctr[i],1,4)
      else{
         if (substr(names.ctr[i],4,4)=="-")
```

```
            n.ctr <- substr(names.ctr[i],1,3)

         else n.ctr <- substr(names.ctr[i],1,5)
          }
      if (n.obs==n.ctr)  found.ctr <- T
      else i <- i+1
  }
   i <- 1
   # looking for the scenario period corresponding to the same station.
   while(found.sc==F & i<length(names.sc)){
       if (substr(names.sc[i],5,5)=="-")
          n.sc <- substr(names.sc[i],1,4)
       else{
          if (substr(names.sc[i],4,4)=="-")
             n.sc <- substr(names.sc[i],1,3)

          else n.sc <- substr(names.sc[i],1,5)
           }
        if (n.obs==n.sc) found.sc <- T
else i <- i+1
    }
  # if the three periods have been found for the same station,
  # the corresponding files are read.
  if ((found.ctr==T) & (found.sc==T)){
      found <- T
      vect.found <- cbind(vect.found,as.real( n.sc))
    }
k <- k+1
   }
  }
vect.found
}

# This routine imports into R the files corresponding to the observed,
# controlled and forecast periods for the same station.

read.files <- function(stat.nbr){
# for the OBSERVED PERIOD:
     st.obs <- check.columns(stat.nbr)
# for the CONTROL PERIOD:
     name.ctr <- paste(stat.nbr, "-present.txt",sep="")
     print(paste("station.ctr",stat.nbr ))
     st.ctr <- read.table(paste("/home/alexi/alexi/scenario/present-txt/",name.ctr,sep=""),
 header = TRUE, sep = "\t",col.names=c("year","month","day","rr","temp"))
# for the SCENARIO PERIOD:
     name.sc <- paste(stat.nbr, "-scenario.txt",sep="")
     print(paste("station.sc",stat.nbr))
     st.sc <- read.table(paste("/home/alexi/alexi/scenario/scenario-txt/",name.sc,sep=""),
header = TRUE, sep = "\t",col.names=c("year","month","day","rr","temp"))
stations <- list(obs=st.obs,ctr=st.ctr,sc=st.sc)
stations
}
```