University of Zurich[UZH]

Master's Thesis

Institute of Mathematics

A spate of statistical tests to climate data validation

Author:
Carina Schneider
Matriculation number: 10-737-575
carina.schneider@uzh.ch

Supervisors:
Prof. Dr. Reinhard Furrer
Professor at the Institute of Mathematics
University of Zurich

Dr. David Masson
Postdoctorial Researcher

Zurich,
5th March, 2016

Contents

Abstract

In this master's thesis, an R framework for the analysis of the CMIP5-ng climate data has been developed and documented. The CMIP5-ng is a data portal which provides terra bytes of climate data. It has been set up by the Institute for Atmospheric and Climate Science (IAC) at the ETH Zurich in collaboration with the applied statistics group at the University of Zurich. The CMIP5-ng has evolved from the recently released and well-known CMIP Phase 5 (CMIP5) climate simulation data by the provision of the CMIP5 data using a coherent $2.5° \times 2.5°$ spatial resolution, which introduces new possibilities of analyzing the CMIP5 climate simulation runs. In this thesis, R code has been developed that can identify CMIP5-ng model projections with an unreasonable amount of missing values or suspiciously high or low data values. Furthermore, inhomogeneity detection methods have been implemented on the basis of SNHT, a spatio-temporal model based on GMRF and a spatial model called "Lattice Krig". These methods can be used to analyze and test parts of single CMIP5-ng model projections or whole classes of climate model projections for different sorts of inhomogeneities. The R application of the framework developed has given an indication that certain model projections of the CMIP5-ng diverge in their values at polar regions. Furthermore, it is suggested that variable projections of the Surface Upwelling Longwave Radiation (rlus) are more heterogeneous than the, probably more thoroughly analyzed, Near Surface Temperature (tas). Moreover, it has been discovered that the model projections for equal climate variables, scenarios and resolution also differ in the number of time units that have been modeled and vary regarding the number of missing values, which especially makes comparison among projections harder.

Acknowledgments

I would like to express my gratitude to Prof. Reinhard Furrer for the invaluable discussions and his competent supervision throughout my master's thesis. He has always been committed to having discussions despite his demanding schedule, which cannot be taken for granted. I would also like to thank Dr. David Masson for organizing a meeting with MeteoSwiss.
Furthermore, I would like to thank Josh Browning (PhD student at the Colorado School of Mines, USA) for the fruitful collaboration on the `snht` R package [Browning and Schneider, 2016] and giving me the opportunity to become a co-author of his package. Special thanks also go to Dr. Douglas Nychka (Senior Scientist at the National Center for Atmospheric Research, Boulder, USA) for his valuable suggestions on the usage of his Lattice Krig model as well as providing me with papers and further information.
My appreciation also goes to Carsten Rose (IT Systems Administrator of our Mathematics Faculty) for his immediate support on any IT based issues even outside office hours.
Finally, I wish to pay recognition to my family and friends for the warm-hearted support throughout my thesis.

Chapter 1

Introduction

Climate change is having an indisputable effect on the climate of the Earth and its biodiversity. The precise effect of such a change is still not well understood. Climate modeling projects such as the CMIP, however, have provided large pools of simulated climate data that are used to better understand the consequences of climate change. Nonetheless, these data pools are often not homogenized and raw for practical use. The goal of this master's thesis is to develop a framework of statistical tests that can detect corrupt or erroneous runs in the latest phase of the CMIP, the CMIP5 data. Nowadays, climate models are becoming more complex as the physical description of climate model processes enhances. Simultaneously,"[...] every bit of added complexity [...] also introduces new sources of possible error (e.g., via uncertain parameters) and new interactions between model components that may, if only temporarily, degrade a models simulation of other aspects of the climate system" [Stocker et al., 2013]. It is, therefore, important that newly developed models are investigated properly and this thesis aims to provide tools to do so.

1.1 Research questions

Throughout this thesis, the following research questions are answered with respect to the CMIP5-ng data and models:

1. How can one detect climate model projections with unusually many missing values or values that are not within a reasonable range?
2. How can the SNHT be applied to find anomalies in the climate data?
3. How can one detect drifts, large scale and small scale anomalies in the climate data using GMRF?
4. How can a newly developed spatial model called "Lattice Krig" be used to detect anomalies?

1.2 Structure

This thesis introduces three mathematical concepts namely SNHT, GMRF and Lattice Krig. Each concept is briefly presented in a theoretical manner but, for the most part, emphasis is put on practical applications of the inhomogeneity tests that were developed on the basis of SNHT, GMRF and Lattice Krig. R examples based on simulated and the CMIP5-ng data projections illustrate the usage of pre-built or newly developed R functions and aim to show other scientists and engineers ways and tools to approach anomaly or inhomogeneity detection in the CMIP5-ng model projections. Each concept

(SNHT, GMRF, Lattice Krig etc.) and its R implementations are discussed with respect to their performance in detecting different types of inhomogeneities (e.g., drifts, local shifts, global shifts) as well as runtime and stability. It must be pointed out that this thesis is conceptualized for the CMIP5-ng data and its analysis. However, some concepts may also be easily transferred to other climate data.

1.3 Inhomogeneity

As mentioned before, it is essential that new model projections are checked for erroneous runs. In order to detect these anomalies in the CMIP5-ng simulation runs, various, so-called, "homogeneity tests" have been developed and presented in this thesis. Before introducing the reader to these tests, it may be sensible to define what is referred to as a "homogeneous" data set in this thesis. This section serves to clarify the most important homogeneity and inhomogeneity related terms and briefly explains the causes and types of inhomogeneities in order to outline what anomalies ought to be detected.

1.3.1 Definition of inhomogeneity and homogeneity

In climate data sets, an inhomogeneity is a change point in the data caused by non-climatic factors [Toreti et al., 2011]. A climate data set is, therefore, called homogeneous if it is free from inhomogeneities, i.e., the only variation in the data ought to be due to real climate variability. This is equivalent to having no shifts in the mean level of the investigated time series of a climate variable after removing seasons and other climatic occurrences.

1.3.2 Causes of inhomogeneous data

Alexandersson and Moberg [1997] as well as Menne and Williams Jr [2009] see possible causes of inhomogeneities in the relocation of measuring stations, changes in the instrumentation or its exposure as well as observation practices or schedules.
Since the CMIP5 data does not consist of real observations but rather of computer based simulations, different causes of inhomogeneities have had to be considered. Shifts in the mean level originate from errors in the developed model simulation software. Mainly, the complex climate models which exist today are more vulnerable to these sorts of errors.

1.3.3 Types of inhomogeneities

Erroneous climate simulation runs may result in different types of inhomogeneities. Four possible cases of inhomogeneities are displayed in Figure 1.1, for each of which five time series have been generated. The graph above the plot of the series represents a possible spatial arrangement of the five series.

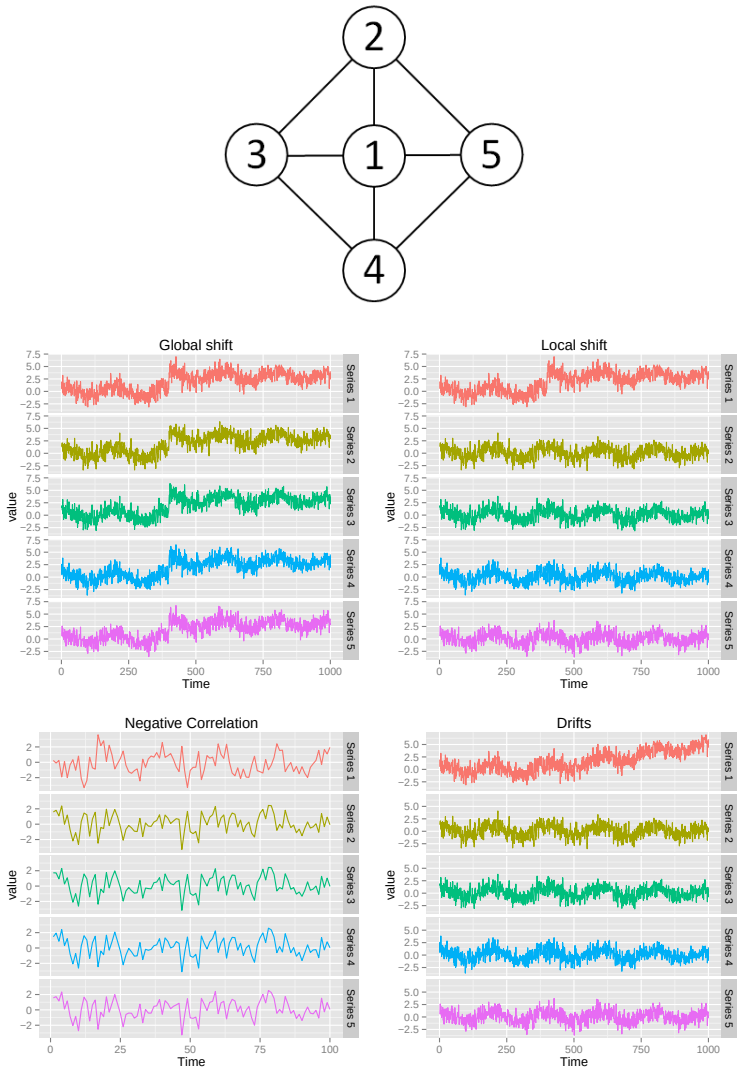

Figure 1.1: Different types of inhomogeneities illustrated for 5 locations (1st time series is inhomogeneous) for which the stations may be arranged as in the graph above the depicted series.

In the subsequent chapters, all of the homogeneity tests developed have been analyzed with respect to the performance of detecting these four types of inhomogeneities.

1.4 Methodology and contributions

Until now, a single method that can detect large and small scale inhomogeneities as well as inhomogeneities in space and time simultaneously and with the same accuracy has not been developed. Therefore, a framework of methods is preferable over single ones. Such a framework has been developed in the course of this thesis. It essentially consists of preanalysis tools that can be used to detect an unreasonable amount of missing values or suspicious data values, the SNHT, the applications of a GMRF spatio-temporal model and inhomogeneity indicator based on the Lattice Krig spatial model. The following section briefly elucidates the contributions of this thesis to existing methods and R code related

to the SNHT, GMRF and Lattice Krig and provides insight into the reasons for their selection for this thesis.
One part of the framework is the SNHT, which is a well established method in climate institutes such as MeteoSwiss. It is advantageous for finding inhomogeneities such as local shifts and drifts but is not suitable for the detection of global shifts or finding negatively correlated time series. Josh Browning has developed the `snht` R package [Browning and Schneider, 2015] that provides the `pairwiseSNHT()` function, which uses the pairwise difference series to detect inhomogeneities in space and time via the SNHT statistics. Part of this thesis has been the improvement of this package through the contribution of a vignettes description and code debugging.
Chapter 5 introduces the GMRF and a spatio-temporal model that is based on Rue and Held [2005]. The GMRF methods compensate for the lack of success in global shift detection of the SNHT but still cannot detect a range of negatively correlated time series. The usage of GMRF is feasible in the context of climate data since it models the conditional dependence structure of space and time in only one multivariate normal distribution that includes the spatio-temporal structure of climate data in its sparse precision matrix. Inhomogeneity testing can then be done on the basis of the likelihood ratio statistics, which is known to have a χ^2_1 distribution. The Maximum Likelihood Estimate (MLE) under the null and alternative hypotheses involves the calculation of the determinant of the sparse Symmetric and Positive Definite (SPD) precision matrix. This can be efficiently calculated by the Cholesky factorization. Applications of this GMRF spatio-temporal model were presented by Schibli [2011] in her master's thesis. Schibli [2011], however, has only developed R code for global shift detection via the GMRF spatial model. As part of this thesis, an additional local shift detection tool with the GMRF model has been implemented and runtime has been improved via the usage of newly developed R functions such as `precmat.GMRFreglat()` of the `spam` R package [Furrer, 2015]. Furthermore, the R `gmrfHomogeneityTestComp()` function is provided, which allows a more user friendly application of inhomogeneity detection with the spatio-temporal GMRF model.
Moreover, use has been made of a spatial model called "Lattice Krig". The Lattice Krig spatial model as well as the corresponding `LatticeKrig` R package [Nychka et al., 2015] were developed by Dr. Nychka and his working group. The Lattice Krig spatial model is a Kriging method that uses radial basis functions on different levels of spatial resolutions. The stochastic coefficients of these basis functions follow the principles of the GMRF. This construction ensures better runtime than many other spatial models [Nychka et al., 2013]. In this thesis, the Lattice Krig model estimation, which is provided by the `LatticeKrig` R package [Nychka et al., 2015], has been used as a basis for the development of the `refLatTest()` and `sigmaLatTest()` functions. These R functions use reference spatial fields, smoothing and noise parameters to find inhomogeneities in space, by which the time component is accordingly fixed. The applications of SNHT and GMRF analysis of the spatial structure of the whole Earth at a specific time can be done. The SNHT and GMRF methods are still only computationally feasible over relatively small spatial regions.
Overall, the methods included in this master's thesis have been selected based on their computational cost and methods have been sought that complement each other in finding different types of inhomogeneities. Recent development and extensions of the `spam` [Furrer, 2015] and `LatticeKrig` [**?**] R packages have provided efficient base-code that have been used to write state-of-the-art R scripts.

Chapter 2

Origin and structure of the data

The CMIP5-ng data and its quality which is investigated in this thesis is of extreme importance to the climate science community. This chapter gives insight into the origin of the CMIP5-ng and why it is such an invaluable project, but it also introduces the overall characteristics (variables, scenarios, resolution etc.) and the attributes of the NetCDF (*.nc) CMIP5-ng data files. This chapter and, especially Section 2.2.6, contains information which might be useful for the interpretation of R examples in the subsequent chapters.

2.1 CMIP and IPCC

The data that is investigated in this thesis originates from the Coupled Model Intercomparison Project (CMIP). The CMIP was initiated under the World Climate Research Programme in 1995 [Taylor, 2009]. Since then its working groups have provided globally coupled, ocean-atmosphere, general circulation models under certain boundary conditions "[...] such as the solar 'constant' and atmospheric concentrations of radiatively active gases and aerosols" [Covey et al., 2003]. Originally, the CMIP modeled "control runs" in which radiative forcing[1] was held constant. Later on, scenarios such as a constant 1% increase of CO_2 per year were modeled [Taylor, 2009]. Nowadays, the CMIP provides data under several newly developed scenarios as well as control runs to assess model performance. Furthermore, the CMIP has acquired more research groups from all over the world, which are involved in the climate modeling process. How much an increase of the number of models really enhances the overall quality of climate projections is, however, still controversial. Masson and Knutti [2011], for instance, criticize the assumption of models being independent and, therefore, contributing additional information as not always true since "[...] successful concepts in models are often copied or inherited". Nevertheless, nowadays, the CMIP models and their output data sets are an essential source for the Intergovernmental Panel on Climate Change (IPCC) in their Assessment Reports (AR). In the AR, the IPCC assesses climate change in a scientific manner. Furthermore, the release of these AR has probably contributed to IPCC's position as the world's leading provider of climate change information [IPCC, 2016].
In its latest assessment report the IPCC reported that climate models in the fifth phase of CMIP (CMIP5) have improved due to better physical description of climate processes, new model components and better resolution [Stocker et al., 2013]. For instance, several

[1] Radiative forcing or climate forcing is defined to be the difference of the solar irradiance which is absorbed by the Earth on a long term and the energy which is radiated back into space. A positive radiative forcing therefore means that there is more incoming energy than outgoing which implies that the system warms up whereas a negative radiative forcing is interpreted as a cooling effect [Stocker et al., 2013].

models of the CMIP5 (latest phase of CMIP) simulated complex climate phenomena as the Monsoon and the El Niño-Southern Oscillation (ENSO) better than previous phases of the CMIP [Stocker et al., 2013]. This is also clearly evident in Figure 2.1.

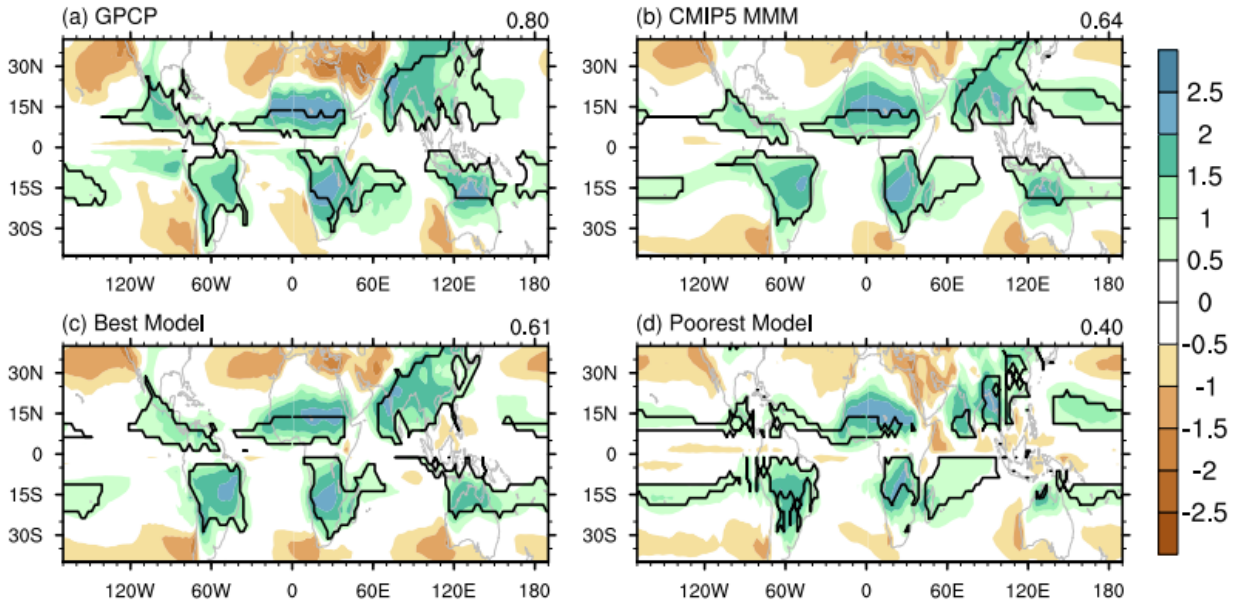

Figure 2.1: Monsoon precipitation intensity (shading) and domain (lines) for "(a) observation-based estimates from Global Precipitation Climatology Project (GPCP), (b) the CMIP5 multi-model mean, (c) the best model and (d) the worst model in terms of the threat score" [Stocker et al., 2013].

Overall, the CMIP is probably the most important and influential climate modeling project in the world, which has expanded tremendously over the last decade and improved its model performance. The quality of its climate simulations is of extreme importance to IPCC, the climate science community and the world.

2.2 CMIP5-ng

As mentioned before, CMIP5 provides a pool of simulated data from various research groups. Not all of the research groups have provided their simulations using the same spatial resolution. The Institute for Atmospheric and Climate Science (IAC) at the ETH Zurich has, therefore, edited the original CMIP5 data and created the data pool CMIP5-ng, which provides all the CMIP5 data on a coherent $2.5° \times 2.5°$ spatial grid. This is especially useful for comparison of data models and ensembles.
In order to get an understanding of these climate models and ensembles as well as the climate variables, scenarios and resolution of the CMIP5-ng, details are provided in the next few sections. At this point, it should be mentioned that the CMIP5 as well as the CMIP5-ng are only now being evaluated, i.e., there are only a few references available [Stocker et al., 2013]. Most of the information and plots provided in this Section 2.2 have been obtained and produced directly from the CMIP5-ng data files, which have been provided on an ETH server over the course of this thesis.

2.2.1 Climate variables

The CMIP5-ng provides simulated data for 21 different ocean, land and atmospheric climate variables. An overview can be derived from Table 2.1.

Variable	Longname	Unit	Comment
tas	Near-Surface Air Temperature	K	reported at 2 m height
tasmax	Daily maximum Near-surface Air Temperature	K	reported at 2 m height
tasmin	Daily minimum Near-surface Air Temperature	K	reported at 2 m height
clt	Total Cloud Fraction	%	for the whole atmospheric column as seen from the surface or the top of the atmosphere. Include both largescale and convective clouds
evspsbl	Evaporation	kg m^{-2} s^{-1}	at surface; flux of water into the atmosphere due to conversion of both liquid and solid phases to vapor
rsds	Surface Downwelling Shortwave Radiation	Wm^{-2}	
rsus	Surface Upwelling Shortwave Radiation	Wm^{-2}	
rsut	Top of Atmosphere Outgoing Shortwave Radiation	Wm^{-2}	
rtmt	Net Downward Flux at Top of Model	Wm^{-2}	reported only if it differs from the net downward radiative flux at the top of the atmosphere.
rlds	Surface Downwelling Longwave Radiation	Wm^{-2}	
rlut	Top of Atmosphere Outgoing Longwave Radiation	Wm^{-2}	
rlus	Surface Upwelling Longwave Radiation	Wm^{-2}	
pr	Total Precipitation	kg m^{-2} s^{-1}	at surface; includes both liquid and solid phases from all types of clouds (both large-scale and convective)
psl	Sea Level Pressure	Pa	
sos	Sea Surface Salinity (Practical Salinity Units)	psu	
tos	Sea Surface Temperature	K	may differ from "surface temperature" in sea ice regions
sic	Sea-ice Concentration	%	
mrro	Total Runoff	kg m^{-2} s^{-1}	part of precipitation which does not evaporate or transpire and flows back to water bodies
mrso	Total Soil Moisture Content	kg m^{-2}	
mrros	Surface Runoff	kg m^{-2} s^{-1}	part of total runoff which flows over the surface
mrsos	Moisture in the Upper Part of the Soil Column	kg m^{-2}	mass of water in all phases in uppermost 10 cm of soil

Table 2.1: Next generation CMIP5 variables [Taylor, 2013b].

2.2.2 Climate scenarios

The CMIP5 working groups simulated the above variables under certain climate scenarios. Scenario simulations in climate research are a way to explore the consequences of realistic or unrealistic changes in the climate of Earth. A scenario is not a forecast but rather a description of how climate may develop based on a set of coherent assumptions [Stocker et al., 2013]. These assumptions may be described through the amount of atmospheric greenhouse gases or aerosols at certain times (e.g., in "abrupt4xCO2", see Table 2.2) in the past or future but they can also be summarized through global characteristic numbers such as natural and anthropogenic radiative forcing. Representative Concentration Pathways (RCP) are scenarios assuming different levels of radiative forcing. As illustrated in Table 2.2, the CMIP5 provides data under four different RCP scenarios (RCP26,RCP45, RCP60 and RCP85), all of which have a different target radiative forcing level in the year 2100. In order to get a rough understanding of radiative forcing, Figure 2.2 illustrates what its natural or anthropogenic components and their quantities were on Earth between 1750 and 2011.

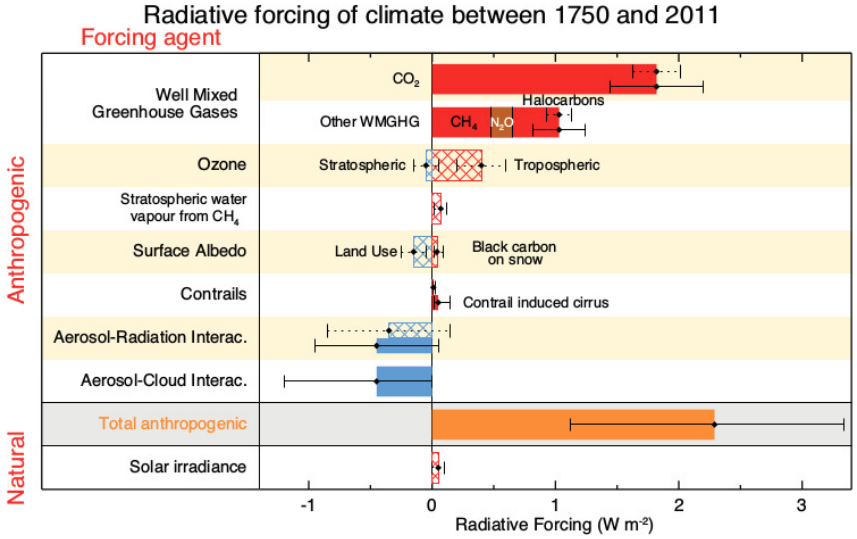

Figure 2.2: Radiative forcing factors from 1750 until 2011 as presented in the IPCC AR5 [Stocker et al., 2013].

One can clearly see that greenhouse gases were a driving force of radiative forcing. A certain combination of these factors above could then potentially lead to RCP45, RCP60 etc. scenarios in the future.
Apart from the future-related RCP scenarios, preindustrial runs such as "piControl" (preindustrial control run, from $\approx$ 1750-1850) and historical runs (1850-2005) such as "historicalGHG" or "historicalNat" are also simulated in the CMIP5. These past-oriented scenarios are used to control the performance of climate models. They, for instance, build a basis for the analysis of global surface temperature variability [Stocker et al., 2013]. Therefore, past-related scenarios become tools to model future-related scenarios such as the RCP-scenarios.
The so-called "abrupt4xCO2" scenario was designed to derive equilibrium climate sensi-

tivities[2] [Stocker et al., 2013].
More characteristics of all the scenarios briefly mentioned above can be extracted from the Table 2.2.

Scenario	Explanation
rcp26	Radiative forcing peak at $\approx 3\,\mathrm{Wm}^{-2}$ (equal to 421 ppm CO_2) before the year 2100 then decline to $2.6\,\mathrm{Wm}^{-2}$ until 2100
rcp45	Radiative forcing is stabilized at $\approx 4.5\,\mathrm{Wm}^{-2}$ (equal to 538 ppm CO_2) after 2100
rcp60	Radiative forcing is stabilized at $\approx 6\,\mathrm{Wm}^{-2}$ (equal to 670 ppm CO_2) after 2100
rcp85	High pathway; radiative forcing reaches $> 8.5\,\mathrm{Wm}^{-2}$ (equal to 936 ppm CO_2) by 2100 and continues to rise until 2250
piControl	Coupled atmosphere/ocean pre-industrial control run
abrupt4xCO2	Instantaneous quadrupling of CO_2, then stabilized
historicalNat	Historical simulation with natural forcing only
historicalGHG	Historical simulation with greenhouse gas forcing only

Table 2.2: Next generation CMIP5 scenarios Taylor [2014].

The RCP scenarios make up a large part of the CMIP5 scenarios and test examples in this thesis are based on them. Therefore, one may be interested in what time series look like under different RCP scenarios. As an illustration, the Chinese BCC-CSM1-1 model projections of the annual Near Surface Temperature in Switzerland under all RCP scenarios are displayed in Figure 2.3. Until 2005, data consists of historical simulations that agree under all scenarios since radiative forcing is set to historical levels. From 2005 onwards, the radiative forcing increases according to Table 2.2 above, resulting in different responses of the temperature and other climate variables that are not depicted here. Among other things, the plot shows the ubiquitous notion of increasing temperature until the year 2100 under all RCP scenarios, in which higher radiative forcing leads to higher temperatures.

[2] "Equilibrium climate sensitivity" is defined as the equilibrium change in the annual global mean surface temperature following a doubling of the atmospheric equivalent carbon dioxide concentration [Stocker et al., 2013].

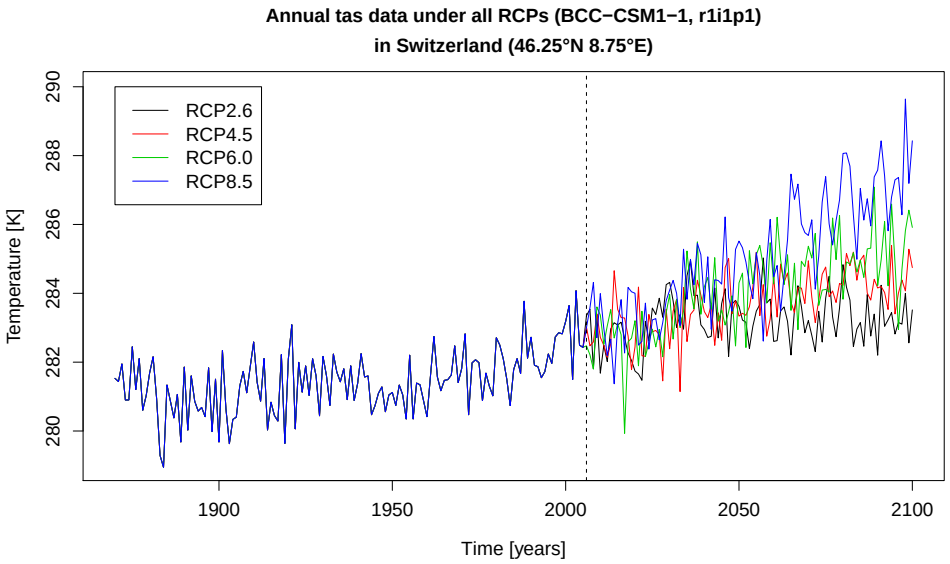

Figure 2.3: RCP projections with the Chinese BCC-CSM1-1 climate model of the annual Near Surface Temperature from 1850 until 2100.

2.2.3 Climate models

As mentioned in Section 2.1, research institutes from all around the world are involved in the CMIP. Each institute provides at least one climate model. A few examples of models and the institutions responsible can be found in Table 2.3.

Modeling Center	Model	Institution
BCC	BCC-CSM1.1 BCC-CSM1.1(m)	Beijing Climate Center China Meteorological Administration
CCCma	CanAM4 CanCM4 CanESM2	Canadian Centre for Climate Modelling and Analysis
CMCC	CMCC-CESM CMCC-CM CMCC-CMS	Centro Euro-Mediterraneo per I Cambiamenti Climatici
CNRM-CERFACS	CNRM-CM5 CNRM-CM5-2	Centre National de Recherches Meteorologiques Centre Europeen de Recherche et Formation Avancees en Calcul Scientifique
CSIRO-BOM	ACCESS1.0 ACCESS1.3	CSIRO (Commonwealth Scientific and Industrial Research Organisation, Australia), BOM (Bureau of Meteorology, Australia)
NASA GISS	GISS-E2-H GISS-E2-H-CC GISS-E2-R GISS-E2-R-CC	NASA Goddard Institute for Space Studies

Table 2.3: Next generation CMIP5 models, for more information about the working groups see Taylor [2013a]

Over all, the CMIP5 working groups have developed approximately 40 different climate models. One might be interested in an example of different model projections of the same

climate variable, scenario and resolution. Figure 2.4 visualizes such model projections of the Near Surface Temperature data under the "abrupt4xCO2" (see Table 2.2) scenario. The models do not agree with respect to their starting values but all show similar trends as a response to a quadrupling of the amount of CO_2 in the atmosphere.

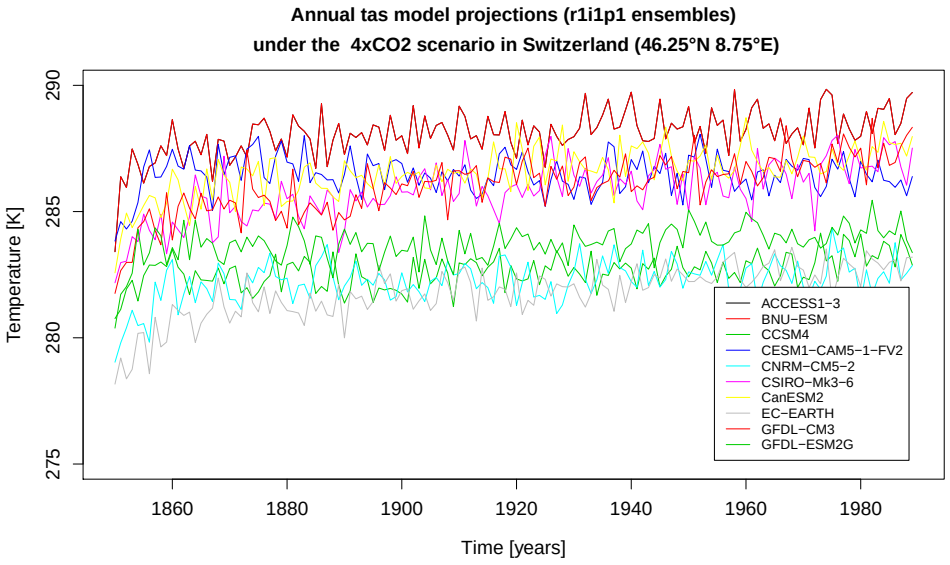

Figure 2.4: Different ensemble projections with the CNRM-CM5-2 model of the monthly Near Surface Temperature under an "piControl" scenario.

At that point, it might be interesting to point out that such comparisons of model projections are a common practice to evaluate model performance in order to estimate the model uncertainty through the amount of spread among the models [Stocker et al., 2013].

2.2.4 Climate ensembles

Climate models, as introduced above, do not just produce one single projection of a scenario for a certain variable and resolution but rather a range of projections are produced that are assumed to be equally likely. These projections using a single model are called "ensembles". Different projections, for instance, arise from different initial values in simulations. The CMIP pursues the idea of ensemble modeling since that offers another tool to quantify the uncertainty of simulations by a measure of spread across the ensembles [Masson and Knutti, 2011]. Different ensembles, that originate from the same model are illustrated in the Figure 2.5. One can clearly recognize dependencies.

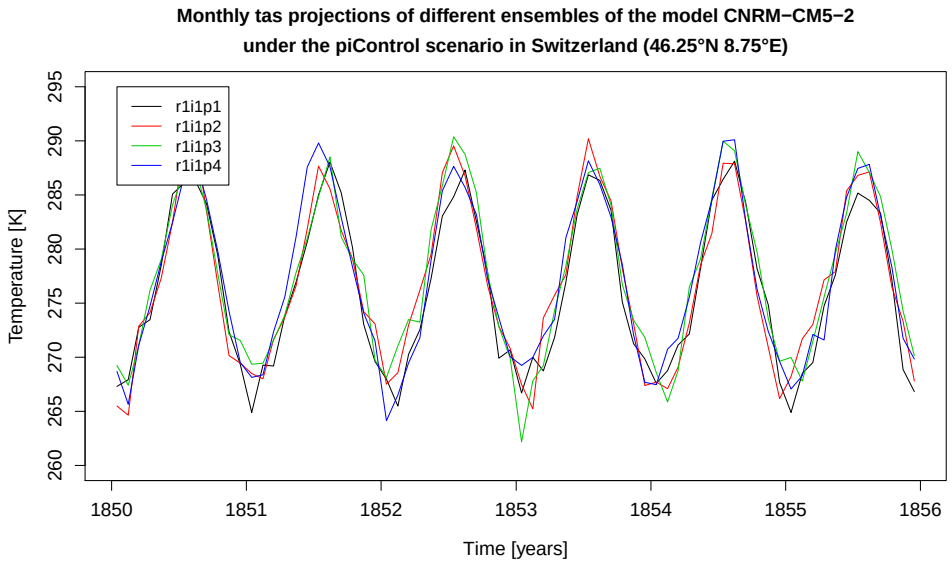

Figure 2.5: Different ensembles of annual Near Surface Temperature under the "abrupt4xCO2" scenario.

When it comes to finding an average representation of a specific scenario, variable and resolution, often throughout this thesis the arithmetic mean over specific models are used. Each chosen model is then represented as the average over all ensembles. In Section 5.5.1 tools are provided to prevent the usage of biased representations.

2.2.5 Spatial and temporal resolution

As mentioned above, CMIP5-ng model projections are available on a coherent $2.5° \times 2.5°$ longitude-latitude grid which corresponds to spatial squares of approximately 278 km $\times$ 278 km at the equator. The original CMIP5 resolution is also available, however, the grid size may differ based on which institute has produced the data.
Apart from spatial resolutions, there are three types of temporal resolutions provided in the CMIP5-ng data. These are,

- month (`mon`)
- season (`sea`) (i.e., 3 month periods in all consecutive possibilities: January/February/March, February/March/April etc.)
- year (`ann`)

The R code and applications that have been implemented in the course of this thesis usually are illustrated on the basis of data on a $2.5° \times 2.5°$ spatial grid choosing months as the temporal resolution to guarantee coherence and find inhomogeneities with respect to the highest possible temporal resolution.

2.2.6 NetCDF files and their naming convention

Sections 2.2.1 to 2.2.4 above have illustrated the coarse dimensions and characteristics of the CMIP5-ng data. The CMIP5-ng data is provided as 74,595 NetCDF (*.nc) files.

These CMIP5-ng files play an important role and and are referred to as "NetCDF" files throughout this thesis. Using NetCDF files in R requires the `ncdf` R package [Pierce, 2015]. Each file contains an individual model projection of a specific variable, scenario, ensemble and resolution, as described above, as well as some meta data on the spatial grid, units of time etc.
The data files are named after their content, i.e., the climate variable, scenario, model, ensemble, spatial and temporal resolution. More precisely, the files are named according to the following pattern:

```
'variable name'_'temporal resolution'_'model'_
'scenario'_'ensemble'_'spatial resolution'.nc
e.g.,
tas_ann_CanESM2_rcp26_r1i1p1_g025.nc
```

variable name	`tas`, `tasmax`, `tasmin` etc. (see Section 2.2.1)
temporal resolution	`mon`, `ann`, `sea`
model	BCC-CSM1.1, CanAM3, CMCC-CESM etc.
scenario	`rcp26`, `rcp45`, `rcp60`, `rcp85` etc. (see Section 2.2.2)
ensemble	`r1i1p1`, `r2i1p1` etc.
spatial resolution	`g025` ($2.5° \times 2.5°$), `native` (original CMIP5 resolution)

Chapter 3

Preanalysis of the data

Before applying homogeneity tests to different models, scenarios, ensembles etc., one should roughly verify the reasonableness of the data. The two tests presented in this chapter are less time consuming than the homogeneity tests presented in the subsequent chapters. The output of the preanalysis R functions provides meta data on single or multiple NetCDF files. The tests are feasible to detect major mistakes in the data such as wrong signs, an unexpected amount of missing values, outliers etc. in an early phase of analysis. Tolerance intervals are used to indicate outliers with respect to the means, maxima and minima of the n CMIP5-ng data sets that have been investigated.
It is thereby assumed that the n arising means : $\bar{x}^{(1)}, \ldots, \bar{x}^{(n)}$ are samples from a normal distribution. The n minima and n maxima, on the other hand, are thought to be samples from the Gumbel extreme value distribution.
The tolerance interval bounds have been calculated by the function `normtol.int()` and `exttol.int()` from the `tolerance` R package [Young, 2015] and the parameters have been estimated by the Maximum Likelihood Estimation employing the Newton-Raphson algorithm.

3.1 Application and results of the preanalysis

The `standardTest()` and `multipleStanTest()` R functions, which have been developed in the course of this thesis, allow to perform such preanalysis tests as described above. This section illustrates the usage of these functions on the basis of the CMIP5-ng NetCDF files.

3.1.1 The R function `standardTest()`

`standardTest()` takes the path to a single NetCDF file as input and, basically returns the type of the climate variable (`sea`,`land` or `global`), some standard statistics, information on any missing values, the range of the data values and the format of the data. One needs to recall that the existence of missing values is generally not undesirable. If the climate variable type is `sea` or `land`, then by definition, there should be missing values in complementary regions of land or sea respectively. In such cases, the absolute and relative numbers of missing values and a comment (`missComment` $\in$ {`ok`,`suspicious`}) are returned by `standardTest()`. `ok` is returned if the ratio of missing values does not exceed a certain threshold. This threshold is set to 0.4 for sea-type-variables and 0.8 for land-type-variables, which may not in all cases be ideal and can be modified by the user depending on how restrictive one wants to be. In a continuous setting, land is known to cover approximately 29% of the Earth while the ocean covers about 71% but, since the CMIP5-ng has a spatial resolution of $2.5° \times 2.5°$, it would be too restrictive to set the

thresholds to 0.29 and 0.71.
At this point, one may be interested in what output the `standardTest()` method produces. Below, two model projections of the same variable, scenario and resolution are passed to `standardTest()` and their output is compared.

Example 3.1.1. *`standardTest()` applied to the Australian ACCESS1-0 model and the Norwegian NorESM1 model for the Sea Surface Temperature (tos) climate variable, under a ("piControl") preindustrial control run scenario gives the following output:*

```
#standardTest input: path to NetCDF file
#standardTest output:
#name: NetCDF file name
#varname:     climate variable (see section on climate variables)
#type:        'land', 'sea' or 'global'
#             (depending on where the variable can be measured)
#             E.g., tos, i.e., sea surface temperature can only be
#             measured at regions of sea.
#missing:     TRUE/FALSE, if TRUE -->there are missing values
#                     if FALSE--> no missing values
#numbOfNA:    number of missing values
#ratioNA:     ratio of missing values, i.e.,
#             ratioNA=(number of missing values)/(144*72*timeDim)
#             144*72*timeDim corresponds to the total number of values
#             that can be assigned for a 2.5x2.5 degree pixel.
#missComment: "ok"/"suspicious", depending on the ratioNA.
#             "suspicious" if it is higher than a set threshold
#             thresholds: 40% for sea type, 80% for land type variables
#sgn:         sign of the climate variable values
#totmax:      maximum of the climate variable values
#totmin:      minimum of the climate variable values
#average:     arithmetic mean of the climate variable values
#std:         standard deviation of the climate variable values
#timeDim:     number of time units (months, years, seasons) that are modeled
#range:       "ok"/"suspicious"
#             "suspicious": variable values are higher or lower then
#             predefined bounds

> out0 <- standardTest("/.../tos_mon_ACCESS1-0_piControl_r1i1p1_g025.nc")
> out0
                                        name varname type missing
1 tos_mon_ACCESS1-0_piControl_r1i1p1_g025.nc     tos  sea    TRUE

 numbOfNA   ratioNA missComment sgn   totmax   totmin average
 23376000 0.3757716          ok >=0 307.7083 271.2249 286.955

       std  timeDim range
  11.09679     6000    ok

> out1 <- standardTest("/.../tos_mon_NorESM1-M_piControl_r1i1p1_g025.nc")
> out1
                                        name varname type missing
1 tos_mon_NorESM1-M_piControl_r1i1p1_g025.nc     tos  sea    TRUE
```

```
numbOfNA   ratioNA missComment sgn  totmax   totmin  average
22845600 0.3665123          ok >=0 305.611 271.3318 286.5824

      std  timeDim range
11.11418      6012    ok
```

The ACCESS1-0 as well as NorESM1-M have produced non-suspicious projections as the ***range*** *and number of missing values (****missComment****) are declared as* ***ok****. One may recall that* ***missComment*** *and* ***range*** *are marked with* ***ok*** *for the tos sea-type variable, if the data values do not exceed predefined interval bounds for glstos and if there are not more than 40% missing values.*
Nevertheless, ***timeDim*** *differs among the two model projections even though both data sets are model projections of the same climate variable (tos) under the same scenario ("piControl") and resolution. Different numbers of time units make it impossible to do model comparisons as one does not know which values to compare. Due to this issue, Section 3.1.2 provides more details and an R function to find time coherent model projections.*
Apart from ***timeDim****, one might have recognized that the number of missing values also differs between the two model projections. This might be due to different original* ***native*** *spatial resolutions resulting in an unequal specification of boundary pixels between land and sea. The ACCESS1-0 model has produced more missing values relative to the NorESM1 model. At this point, one might be interested in which spatial regions the ACCESS1-0 and NorESM1-M projection differ regarding the missing values. In Figure 3.1, the pixels to which Sea Surface Temperature values have been assigned by the NorESM1-M model but missing values have been assigned by the ACCESS1-0 model are depicted.*

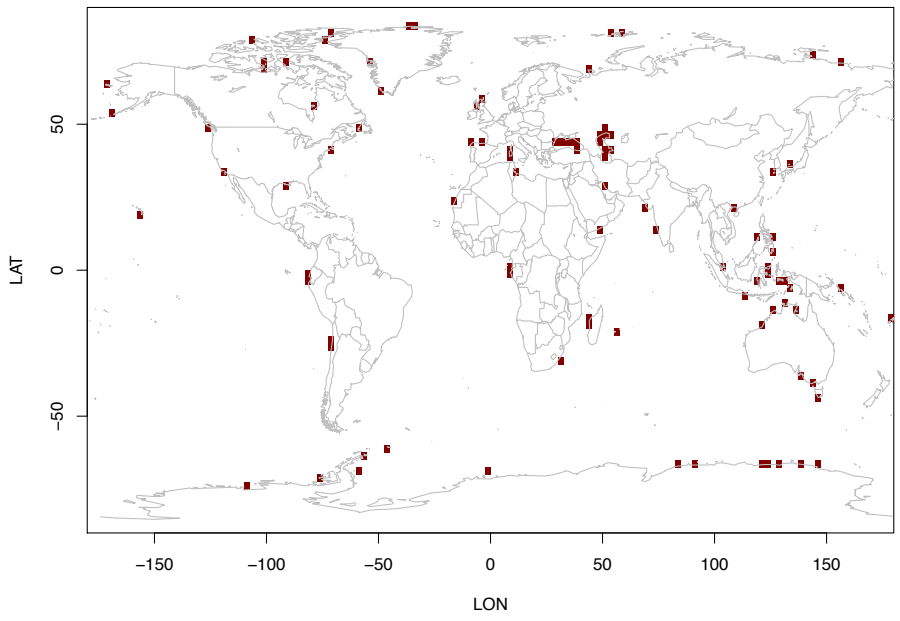

Figure 3.1: Red pixels are regions where ACCESS1-0 has missing values but simultaneously the NorESM1-M has Sea Surface Temperature assigned.

As presumed, the red pixels in Figure 3.1 are found in adjacent regions of land and sea. This fact may need to be kept in mind when comparisons among models and analysis is

conducted in boundary regions of land and sea.
It might be tedious to check the range and missing values separately for every single NetCDF file with the ***standardTest()*** *function. Hence, the* ***multipleStanTest()*** *function was implemented, which automates the* ***standardTest()*** *procedure and gives further indications of outliers in the data via tolerance intervals.*

3.1.2 The R function `multipleStanTest()`

`multipleStanTest()` not only performs `standardTest()` on a single NetCDF file but on a directory containing several NetCDF files. This directory should contain files with model projections from the same climate variable, resolution and scenario to allow reasonable comparison. The `multipleStanTest()` method returns an overview of simple statistics such as the mean, the minimum and maximum, the dimension of time (number of months, seasons, years) as well as a comment about the overall reasonableness (`ok` or `susp`) for each NetCDF file. Furthermore, a tolerance interval, which uses a normal distribution, is defined based on the means of the data values of each file and tolerance intervals for the minimum and maximum are being defined based on the Gumbel extreme value distribution. Stars ("*") indicate mean, maximum and minimum values of data sets that lie outside of these tolerance intervals.

Example 3.1.2. *Again, one can analyze the "piControl" scenario for the Sea Surface Temperature (tos) on a monthly resolution.* ***multipleStanTest()*** *gives the following output:*

```
> out <- multipleStanTest("/.../tos",alpha=0.05,P=0.8)
There are 15  suspicious files in your directory.
```

```
> out
      name                                                timeDim max        min        mean       susp/ok
 [1,] "tos_mon_ACCESS1-0_piControl_r1i1p1_g025.nc"        "6000"  "307.71 *" "271.22 "  "286.95 "  "ok"
 [2,] "tos_mon_ACCESS1-3_piControl_r1i1p1_g025.nc"        "6000"  "305.86 "  "271.24 "  "286.99 "  "ok"
 [3,] "tos_mon_BNU-ESM_piControl_r1i1p1_g025.nc"          "6708"  "306.28 "  "271.36 "  "286.47 "  "ok"
 [4,] "tos_mon_CCSM4_piControl_r1i1p1_g025.nc"            "12612" "305.64 "  "271.12 "  "286.73 "  "ok"
 [5,] "tos_mon_CCSM4_piControl_r2i1p1_g025.nc"            "1872"  "305.28 "  "271.13 "  "286.71 "  "ok"
 [6,] "tos_mon_CCSM4_piControl_r4i1p1_g025.nc"            "600"   "305.23 "  "271.16 "  "286.74 "  "ok"
 [7,] "tos_mon_CESM1-BGC_piControl_r1i1p1_g025.nc"        "6000"  "305.29 "  "271.08 "  "286.76 "  "ok"
 [8,] "tos_mon_CESM1-CAM5-1-FV2_piControl_r1i1p1_g025.nc" "600"   "307.7 *"  "271.22 "  "287 "     "ok"
 [9,] "tos_mon_CESM1-CAM5_piControl_r1i1p1_g025.nc"       "3828"  "306.34 "  "271.24 "  "286.69 "  "ok"
[10,] "tos_mon_CESM1-FASTCHEM_piControl_r1i1p1_g025.nc"   "2664"  "305.51 "  "271.14 "  "286.74 "  "ok"
[11,] "tos_mon_CMCC-CESM_piControl_r1i1p1_g025.nc"        "3324"  "307.64 *" "271.05 *" "286.74 "  "susp"
[12,] "tos_mon_CMCC-CMS_piControl_r1i1p1_g025.nc"         "6000"  "306.93 "  "271.05 *" "286.85 "  "susp"
[13,] "tos_mon_CMCC-CM_piControl_r1i1p1_g025.nc"          "3960"  "305.85 "  "271.11 "  "286.6 "   "susp"
[14,] "tos_mon_CNRM-CM5-2_piControl_r1i1p1_g025.nc"       "4920"  "306.08 "  "270.1 *"  "286.49 "  "ok"
[15,] "tos_mon_CNRM-CM5-2_piControl_r1i1p2_g025.nc"       "1680"  "305.84 "  "270.23 *" "286.43 "  "ok"
[16,] "tos_mon_CNRM-CM5-2_piControl_r1i1p3_g025.nc"       "1680"  "305.9 "   "270.74 *" "286.24 "  "ok"
#... (see appendix, Section 8.2.1)
[38,] "tos_mon_IPSL-CM5A-LR_piControl_r1i1p1_g025.nc"     "12000" "305.36 "  "271.19 "  "285.64 *" "susp"
[39,] "tos_mon_IPSL-CM5A-MR_piControl_r1i1p1_g025.nc"     "3600"  "304.96 *" "271.15 "  "286.17 "  "susp"
[40,] "tos_mon_IPSL-CM5B-LR_piControl_r1i1p1_g025.nc"     "3600"  "306.4 "   "271.24 "  "287.16 "  "susp"
[41,] "tos_mon_MIROC-ESM_piControl_r1i1p1_g025.nc"        "8160"  "306.71 "  "271.26 "  "286.61 "  "ok"
[42,] "tos_mon_MIROC5_piControl_r1i1p1_g025.nc"           "8040"  "305.86 "  "271.24 "  "286.97 "  "ok"
[43,] "tos_mon_MPI-ESM-LR_piControl_r1i1p1_g025.nc"       "12000" "306.85 "  "271.25 "  "286.4 "   "ok"
[44,] "tos_mon_MPI-ESM-MR_piControl_r1i1p1_g025.nc"       "12000" "309.83 *" "271.25 "  "286.61 "  "ok"
[45,] "tos_mon_MPI-ESM-P_piControl_r1i1p1_g025.nc"        "13872" "307.29 *" "271.25 "  "286.49 "  "ok"
[46,] "tos_mon_MRI-CGCM3_piControl_r1i1p1_g025.nc"        "6000"  "304.2 *"  "271.29 "  "287.02 "  "ok"
[47,] "tos_mon_NorESM1-ME_piControl_r1i1p1_g025.nc"       "3024"  "305.67 "  "271.33 "  "286.3 "   "ok"
[48,] "tos_mon_NorESM1-M_piControl_r1i1p1_g025.nc"        "6012"  "305.61 "  "271.33 "  "286.58 "  "ok"
[49,] "tos_mon_bcc-csm1-1-m_piControl_r1i1p1_g025.nc"     "4800"  "305.18 "  "271.26 "  "286.4 "   "ok"
[50,] "tos_mon_bcc-csm1-1_piControl_r1i1p1_g025.nc"       "6000"  "305.12 "  "271.26 "  "286.31 "  "ok"
[51,] "tos_mon_inmcm4_piControl_r1i1p1_g025.nc"           "6000"  "305.33 "  "270.73 *" "287.33 "  "ok"
```

*A file is thereby declared as suspicious (**susp**) if data values are extremely high or low (i.e., the **range** as an output of **standardTest()** is marked as **suspicious**) or if the number of missing values is greater than a set threshold for sea- or land-type variables. All of the 15 suspicious files have a large amount of missing values which is apparent in the **standardTest()** summaries. The GISS-E2-R model projection, e.g., contains approximately 44% missing values, which is over the 40% threshold and, hence, has been classified as **suspicious**:*

```
                                          name varname type missing
1 tos_mon_GISS-E2-R_piControl_r1i1p141_g025.nc     tos  sea    TRUE

 numbOfNA  ratioNA missComment sgn   totmax   totmin average
 63150900 0.436439  suspicious >=0 305.0261 271.2438 287.272

       std timeDim range
  11.19767   13956    ok
```

The large amount of missing values in these files may likely be a result of the homogenization process (transformation from CMIP5 to CMIP5-ng) to guarantee a $2.5° \times 2.5°$ *spatial grid.*
One may also notice that the files that have a mean that lies outside of the tolerance interval (marked with a "" in the "mean" column above) are also declared as suspicious. These two tests (tolerance interval test and missing value/range test), thus, seem to be quite consistent.*
*Apart from different numbers of missing values, it has become apparent that model projections of the same variable, scenario and resolution also vary with respect to their time dimension (i.e., the number of months, years or seasons that are modeled). This is especially undesirable if model comparisons are conducted. If the challenge is to extract only time-coherent files from a certain directory, the function **extractTimeCoh()** (see appendix) might be helpful. **extractTimeCoh()** seeks the time dimension that is attained most frequently. It then returns a subset of the original set of files with this time dimension. In the above example, **timeDim** attains the value 6000 most frequently. Therefore, those files with time series of the length of 6000 months are returned by **extractTimeCoh()**. Suspicious files are not discarded and it is left to the user to decide which files to use.*

```
> extractTimeCoh(out)
The most frequent number of time steps is: 6000
      name                                                timeDim susp/ok
 [1,] "tos_mon_ACCESS1-0_piControl_r1i1p1_g025.nc"     "6000"  "ok"
 [2,] "tos_mon_ACCESS1-3_piControl_r1i1p1_g025.nc"     "6000"  "ok"
 [3,] "tos_mon_CESM1-BGC_piControl_r1i1p1_g025.nc"     "6000"  "ok"
 [4,] "tos_mon_CMCC-CMS_piControl_r1i1p1_g025.nc"      "6000"  "susp"
 [5,] "tos_mon_CSIRO-Mk3-6-0_piControl_r1i1p1_g025.nc" "6000"  "susp"
 [6,] "tos_mon_FGOALS-s2_piControl_r1i1p1_g025.nc"     "6000"  "ok"
 [7,] "tos_mon_GFDL-ESM2G_piControl_r1i1p1_g025.nc"    "6000"  "ok"
 [8,] "tos_mon_GFDL-ESM2M_piControl_r1i1p1_g025.nc"    "6000"  "ok"
 [9,] "tos_mon_MRI-CGCM3_piControl_r1i1p1_g025.nc"     "6000"  "ok"
[10,] "tos_mon_bcc-csm1-1_piControl_r1i1p1_g025.nc"    "6000"  "ok"
[11,] "tos_mon_inmcm4_piControl_r1i1p1_g025.nc"        "6000"  "ok"
```

Chapter 4

SNHT

The SNHT is an inhomogeneity detection test for climate data. It was constructed to find shifts in the mean level of time series at a specific point in time and space, but may be used for drift detection as well (see Section 4.3).
Preliminary, this chapter introduces the SNHT as an analysis tool for single time series via the `snht()` R function (`snht` R package [Browning and Schneider, 2016]) and for spatially arranged series via the `pairwiseSNHT()` R function (`snht` R package [Browning and Schneider, 2016]).
In either instance, a SNHT statistic must be calculated. A very well known SNHT statistic was introduced by Alexandersson and Moberg in 1997. Since then, modified versions have been developed. This chapter shortly discusses different SNHT approaches and then focuses on the `snht` R package version [Browning and Schneider, 2015] that was co-developed as part of the thesis and is put to practice on CMIP5-ng data sets at the end of the chapter.

4.1 Original SNHT

The following section is based on the theory of Alexandersson and Moberg [1997].
A standard normally distributed time series $Z = \{Z_t : t \in \{1, \dots, T\}\}$ with no shift or a single shift at time t_0 can be expressed through the following null and alternative hypotheses:

$$\begin{aligned} H_0 &: Z_t \sim \mathcal{N}(0,1), t \in \{1, \dots, T\} \\ H_1 &: Z_t \sim \mathcal{N}(\mu_1, 1), t \in \{1, \dots, t_0\} \\ &\quad Z_t \sim \mathcal{N}(\mu_2, 1), t \in \{t_0+1, \dots, T\}, \mu_1 \neq \mu_2, \text{ where } \mu_1 = 0 \text{ or } \mu_2 = 0. \end{aligned}$$

Definition 4.1.1. *The original SNHT statistic is defined as:*
$T^s_{max} = \max_{1 \leq t_0 \leq T-1}\{T^s_{t_0}\} = \max_{1 \leq t_0 \leq T-1}\{t_0 \cdot \bar{z_1}^2 + (T - t_0) \cdot \bar{z_2}^2\}$*, where*
$\bar{z}_1 := \frac{1}{t_0} \sum_{t=1}^{t_0} z_t$,
$\bar{z}_2 := \frac{1}{T-t_0} \sum_{t=t_0+1}^{T} z_t$
are the arithmetic means of the standard normally distributed time series realizations $\{z_t\}_{t \in \{1,\dots,T\}}$.

Remark 4.1.1. *The above statistics* T^s_{max} *does not have a closed form distribution.*

Thereby, the standard normally distributed realizations $\{z_t\}_{1 \leq t \leq T-1}$ are obtained from the data through:

$$Z_t = \frac{Q_t - \bar{Q}}{\sigma_Q}, \text{ where} \tag{4.1}$$

$$Q_t = \begin{cases} Y_t - \left(\frac{\sum_{j=1}^{k} \rho_j^2 (X_{jt} - \bar{X}_j + \bar{Y})}{\sum_{j=1}^{k} \rho_j^2} \right), & \text{for temperature data,} \\ \frac{Y_t \cdot \sum_{j=1}^{k} \rho_j^2}{\left(\sum_{j=1}^{k} \rho_j^2 X_{jt} \bar{Y} / \bar{X}_j \right)}, & \text{for precipitation data.} \end{cases}$$

$$\begin{aligned} \rho_j &: \text{correlation coefficient between the candidate and a reference station} \\ X_{j \in \{1,\ldots,k\},t} &: \text{time series of surrounding reference location } j \text{ evaluated at time } t \\ k &: \text{number of reference series} \\ Y_t &: \text{candidate time series evaluated at time } t \\ \bar{X}_j &: \text{mean over time of reference time series } X_j \\ \bar{Y} &: \text{mean over time of candidate time series} \end{aligned}$$

If Alexandersson's method is applied, then for each time series, i.e., for each spatial pixel, the neighbor series would need to be found, the Q-series and Z-series as well as the correlation coefficients to the neighbors would have to be calculated or estimated. Furthermore, a table should be produced via the likelihood ratio statistic that indicates the critical values of T^s_{max}. This is computationally expensive and tedious due to the in-existence of a closed form distribution for the statistic.

4.2 Modified SNHT

Due to some of the disadvantages of the original Alexandersson method, the following section introduces an alternative SNHT approach which is first presented for single time series analysis and then expanded to the analysis of a spatial field of time series in Section 4.2.2.

4.2.1 SNHT on single time series

Apart from the disadvantages of the orginal SNHT mentioned above, the Alexandersson and Moberg version was also critized by Haimberger [2005] concerning the performance of detecting inhomogeneities. Haimberger [2005] came to the conclusion that the original SNHT tends to falsely detect inhomogeneities at the beginning and end of the time intervals, when either t_0 or $T - t_0$ is small and it poorly estimates the time of the change in case there are periodic signals in the data. Haimberger therefore suggested the following approach: For each observation, two means are computed, one for the N observations prior to the one at time t_0, $\bar{X}_{L,t_0}$, and one for the N observations following the one at time t_0, $\bar{X}_{R,t_0}$.
One may then consider the following test statistic for each $t_0 \in \{N+1, \ldots, T-N\}$ of a time series of length T:

$$T_{t_0} = \frac{N}{s^2_{t_0}} \left((\bar{X}_{L,t_0} - \bar{X}_{t_0})^2 + (\bar{X}_{R,t_0} - \bar{X}_{t_0})^2 \right), \tag{4.2}$$

$$\begin{aligned} \bar{X}_{t_0} &: \text{mean of } \bar{X}_{L,t_0} \text{ and } \bar{X}_{R,t_0}, \text{i.e., } \bar{X}_{t_0} = \frac{1}{2}(\bar{X}_{L,t_0} + \bar{X}_{R,t_0}) \\ s_{t_0} &: \text{estimated standard deviation over the } N \text{ observations} \\ &\quad \text{prior and } N \text{ observations following observation } t_0. \end{aligned}$$

Remark 4.2.1. *Haimberger [2005] himself defined the statistic slightly differently. He divided only by s_{t_0} instead of $s_{t_0}^2$. This is thought to be a mistake when again comparing the statistic to Alexandersson's version. The advantage of the formula (4.2) is the fact that the statistic follows an approximate χ_1^2 distribution under the null hypothesis, i.e., under the assumption that there is no change point in the time series.*

Hypothesis testing is then simpler: If the test statistic T exceeds some threshold at time $t_0{}^*$, it is probable that an inhomogeneity occurred at time $t_0{}^*$. This modified SNHT is implemented in R in the `snht` package [Browning and Schneider, 2015]. Its usage is illustrated below for different periods ($N = 20$ and $N = 150$) and normally distributed samples with two artificially generated shifts. Thereby, a Bonferroni adjusted significance level is chosen since multiple testing is conducted.

```
library(snht)
set.seed(123)
baseData <- rnorm(1000)
baseData[201:500] <- baseData[201:500]+.4
baseData[501:600] <- baseData[501:600]-.6
snhtStatistic <- snht(data=baseData,period=20)
plotSNHT(data=baseData,stat=snhtStatistic,
alpha=0.05/960)

snhtStatistic <- snht(data=baseData,period=150)
plotSNHT(data=baseData,stat=snhtStatistic,
alpha=0.05/700)
```

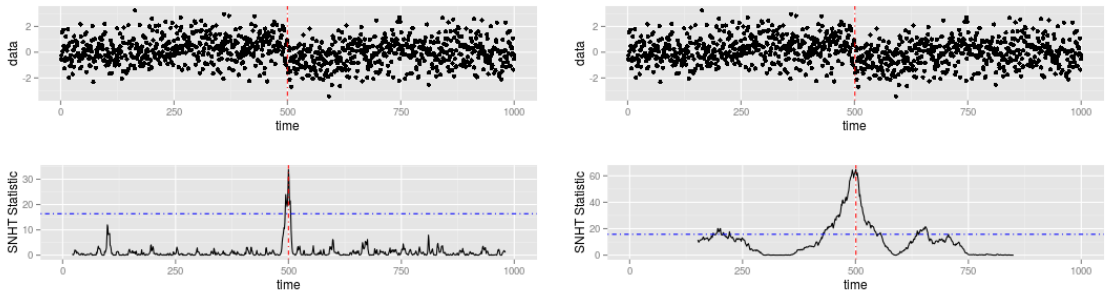

Figure 4.1: Illustration of the SNHT statistic for different period lengths N.

On the one hand, a large period gives better results in the above example as it detects both of the artificially generated shifts, whereas for $N = 20$ only the largest one was detected. On the other hand, large periods generate missing values of the SNHT statistic at the beginning and the end of the time interval. This is due to the definition of the modified SNHT statistic. Shifts that happen at a time $t_0 \leq N$ or $t_0 > (T - N)$ are therefore not detectable, which is especially a disadvantage if the period N is large.
Until now, the computational cost that has been gained from the modified definition of the SNHT compared to the original definition is that the mean does not have to be calculated over T values but only over $2N + 1 < T$ values for each time $t \in \{1, \ldots, T\}$ in a time series. Using `snht()` for every time series of a CMIP5-ng file is however still computationally expensive. Furthermore, no spatial structure is included up to this point. In order to reduce computational cost and to include spatial structure `pairwiseSNHT()` has been implemented as part of the `snht` R package [Browning and Schneider, 2015].

4.2.2 SNHT on pairwise difference series

If many time series from different locations need to be investigated, a relative homogeneity test is an option and has been implemented in the `pairwiseSNHT()` R function. Menne and Williams Jr [2009] suggested to use a relative homogeneity test which uses reference series to detect inhomogeneities at a certain location, an idea that was already part of Alexandersson's method. The `pairwiseSNHT()` R function puts this idea into practice by calculating the pairwise difference series of neighbor time series as described in Menne and Williams Jr [2009]. This has the advantage that overall periodic or linear trends are no longer dominant in the difference series which then can be investigated through the SNHT. It must be pointed out that the underlying assumption is that "similar variations in climate occur at nearby locations" [Menne and Williams Jr, 2009]. If this assumption is violated, the chances of committing a type I error may increase. In this regard, using the correlation coefficient as Alexandersson suggested might generate more accurate results for less correlated series, but at the same time it is computationally more expensive.
The `pairwiseSNHT()` investigates at most `k` difference series for each spatial location. These unique[1] difference series come about through subtracting the `k` closest neighbor series from the investigated candidate series. `pairwiseSNHT()` then counts the number of times the null hypothesis (i.e., having no shift in the difference series) has been rejected where a certain station is involved. The locations with the highest counts of rejections at a specific time are then assumed to be the locations where inhomogeneities occurred. More details on the `pairwiseSNHT()` function can be found in the vignettes of the `snht` R package [Browning and Schneider, 2015], which has been written as part of this thesis.

4.3 Inhomogeneity detection performance

The `pairwiseSNHT()`, as briefly introduced above, is suggested to be used as a CMIP5-ng inhomogeneity analysis tool. With this in view, this section serves to show what one can expect regarding the performance of detecting different types of inhomogeneities as presented in Section 1.3.3. The performance is illustrated for five simulated time series spatially arranged in the following manner:

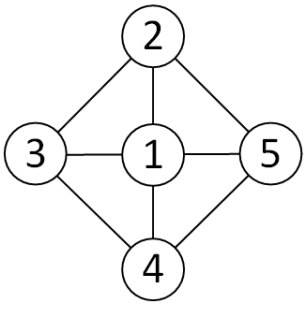

Figure 4.2: The arrangement of 5 stations for which time series were generated.

The stations $1, \ldots, 5$ are thought to be arranged with the following Euclidean distances

[1] The difference series are unique since for neighbor pairs i and j only the differences "series i-series j" or "series j-series i" are analyzed not both differences.

to each other:

$$D = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & \sqrt{2} & 2 & \sqrt{2} \\ 1 & \sqrt{2} & 0 & \sqrt{2} & 2 \\ 1 & 2 & \sqrt{2} & 0 & \sqrt{2} \\ 1 & \sqrt{2} & 2 & \sqrt{2} & 0 \end{pmatrix}.$$

D_{ij} represents the distance of station i to station j. In the following paragraphs the five series are generated with high correlation among each other and a periodic signal in order to simulate data with similar characteristics as the CMIP5-ng data.

4.3.1 Local shifts of the mean

At first, one can look at the reaction of `pairwiseSNHT()` to a local shift. For this purpose, Series 1 has been shifted by a magnitude of 0.5 at time $t_0 = 401$. In R, the data has been generated as follows:

```
set.seed(2)
Cor <- rbind(c(0.5,0.8,0.8,0.8,0.8),c(0,0.5,0.8,0.5,0.6),
c(0,0,0.5,0.8,0.5),c(0,0,0,0.5,0.6),c(0,0,0,0,0.5))
Cor <- t(Cor)+Cor
baseData <- rmvnorm(mean=rep(0,5),sig=Cor,n=1000)+cos(1:1000*2*pi/200)
baseData[401:1000,1] <- baseData[401:1000,1]+0.5
```

Figure 4.3 depicts the data graphically. The shift can hardly be visually detected.

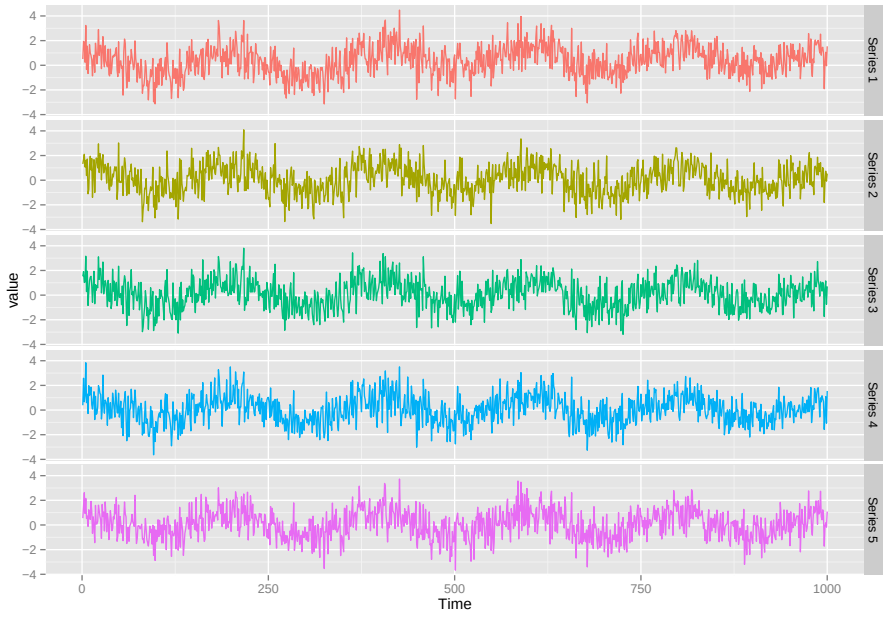

Figure 4.3: Five simulated time series where only Series 1 is shifted by 0.5 at time $t_0 = 401$.

In order to apply `pairwiseSNHT()` the following code can be run:

```
install.packages("/.../snht_1.0.4.tar.gz",type="src",repo=NULL)
```

```
library(snht)
colnames(baseData) <- "1":"5"
baseData <- data.frame(time = 1:1000, baseData)
baseData <- melt(baseData, id.vars = "time", variable.name = "location",
                 value.name = "data")
baseData$location <- gsub("X","",baseData$location)

out1 <- pairwiseSNHT(baseData, dist, k=3, period=200,
crit=qchisq(1-0.05/600,df=1), returnStat=T)
pairs <- colnames(out1)
out2 <- pairwiseSNHT(baseData, dist, k=3, period=200,
crit=qchisq(1-0.05/600,df=1), returnStat=F)
out2$breaks

# > out2$breaks
# time location     shift
#  402        1 0.5857773
```

The location, time and shift are all estimated well. Figure 4.4 visualizes the SNHT statistic for each difference pair, whereby the local shift in Series 1 is apparent.

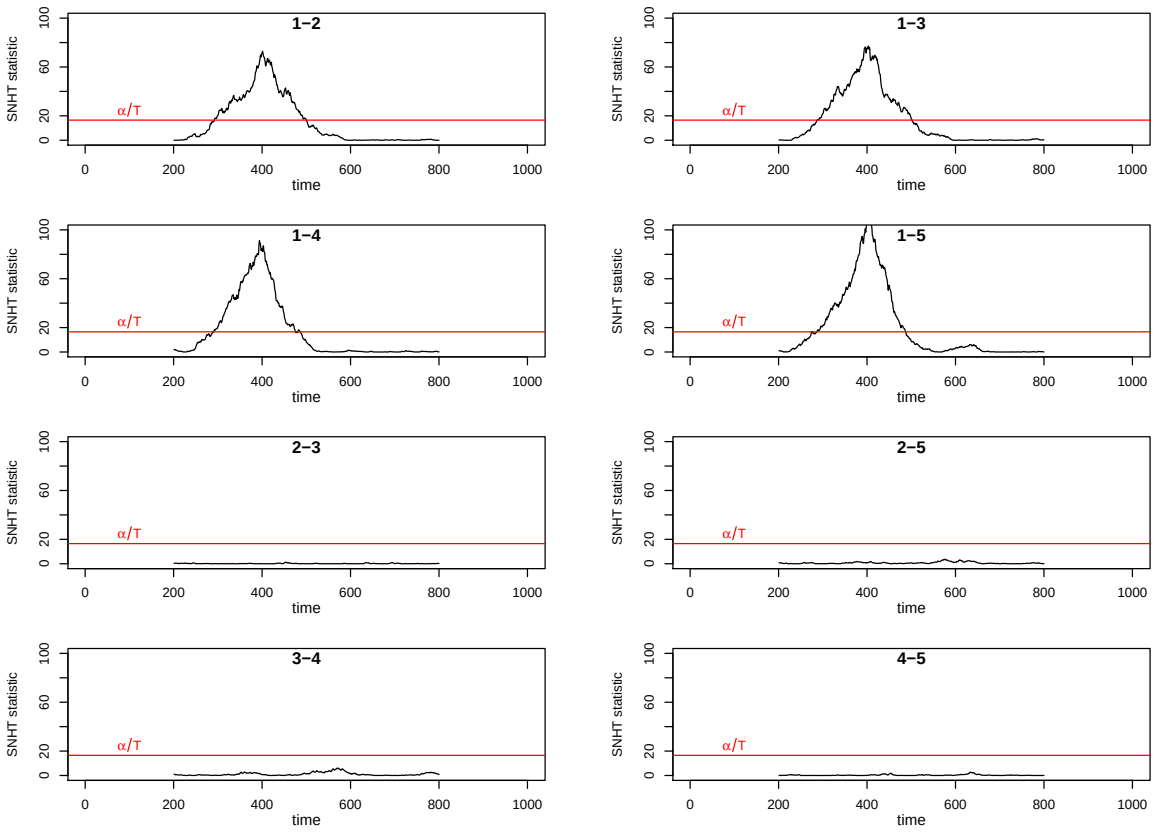

Figure 4.4: SNHT statistic for difference series of neighbor locations with high correlation among each other. Thereby, e.g., "1-2" refers to the difference series of time series 1 and time series 2.

Until now, `pairwiseSNHT()` seems to perform well at local shift detection. However, it might be obvious, but still important, to point out that the performance deteriorates with

lower correlation among the series. Simultaneously, the probability of committing a type I error can increase. The following example shows that the SNHT statistic is not as high as before with no correlation among the time series.

```
set.seed(2)
Cor <- diag(5)
baseData <- rmvnorm(mean=rep(0,5),sig=Cor,n=1000)+cos(1:1000*2*pi/200)
baseData[401:1000,1] <- baseData[401:1000,1]+0.5
#... (same as before)
out2 <- pairwiseSNHT(baseData, dist, k=3, period=200,
crit=qchisq(1-0.05/600,df=1), returnStat=F)
out2$breaks
# > out2$breaks
#  time location     shift
#   402        1 0.7974403
```

The magnitude of the shift is not as accurately estimated as before and the SNHT statistic is lower at time $t_0 = 402$ as apparent in Figure 4.5.

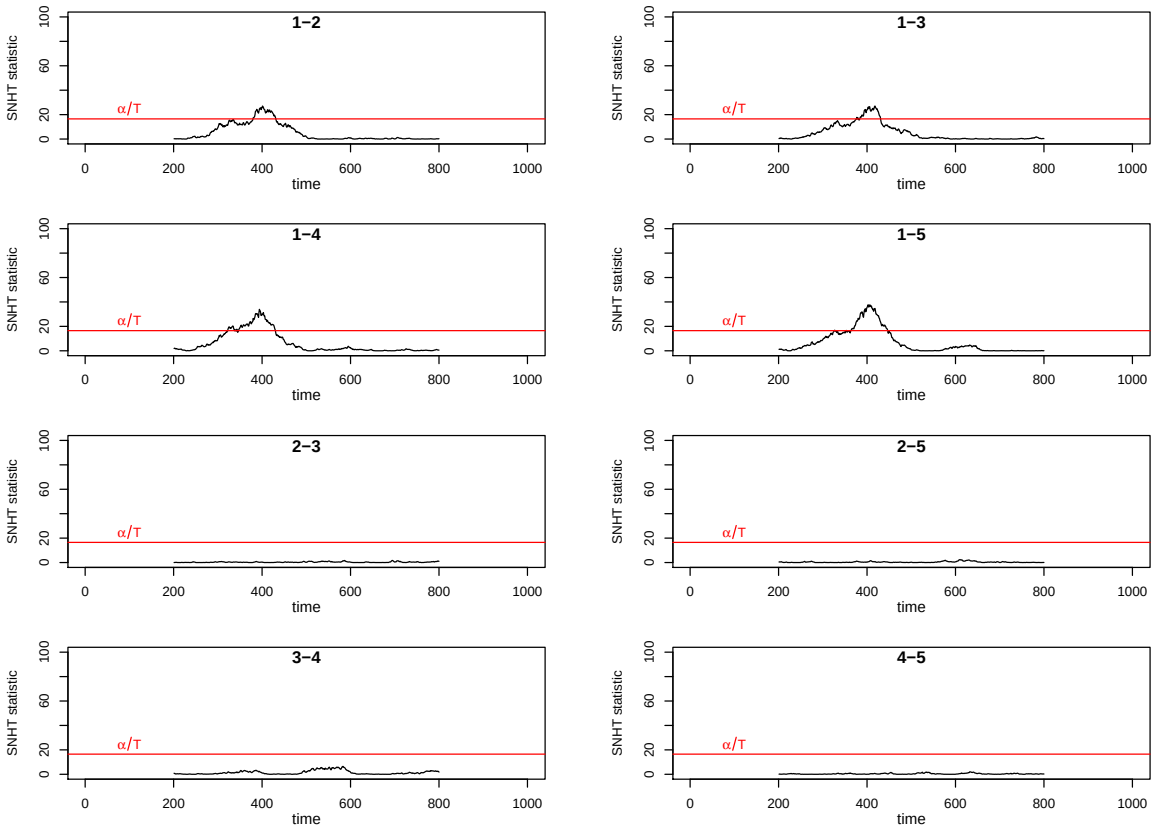

Figure 4.5: SNHT statistic for difference series of neighbor locations with no correlation among each other. Thereby, e.g., "1-2" refers to the difference series of time series 1 and time series 2.

Over all, the performance for cases of low correlation is still acceptable but not as good as for correlated data.

4.3.2 Local drifts

The SNHT and especially `pairwiseSNHT()` are not constructed to detect drifts in one or a few time series, i.e., the maximal SNHT statistic is not a good mean to detect the time or magnitude of the drift. `pairwiseSNHT()` can, however, detect the location where such a drift occurred under the assumption that the neighbor series are all homogeneous. The following example shows an extract of the code and the output of `pairwiseSNHT()` on a data set with a drift introduced in the first out of the five time series.

Figure 4.6: Five simulated time series with an introduced drift in series 2 at time 401.

```
Cor <- rbind(c(0.5,0.8,0.8,0.8,0.8),c(0,0.5,0.8,0.5,0.6),
c(0,0,0.5,0.8,0.5),c(0,0,0,0.5,0.6),c(0,0,0,0,0.5))
Cor <- t(Cor)+Cor
baseData <- rmvnorm(mean=rep(0,5),sig=Cor,n=1000)+cos(1:1000*2*pi/200)
baseData[401:1000,1] <- baseData[401:1000,1]+1/120*(401:1000)-10/3
#... (same as above)
out2 <- pairwiseSNHT(baseData, dist, k=3, period=200,
crit=qchisq(1-0.05/600,df=1), returnStat=F)
> out2$breaks
  time location     shift
1  744        1 1.7188140
2  534        1 1.5715238
3  333        1 0.3578464
```

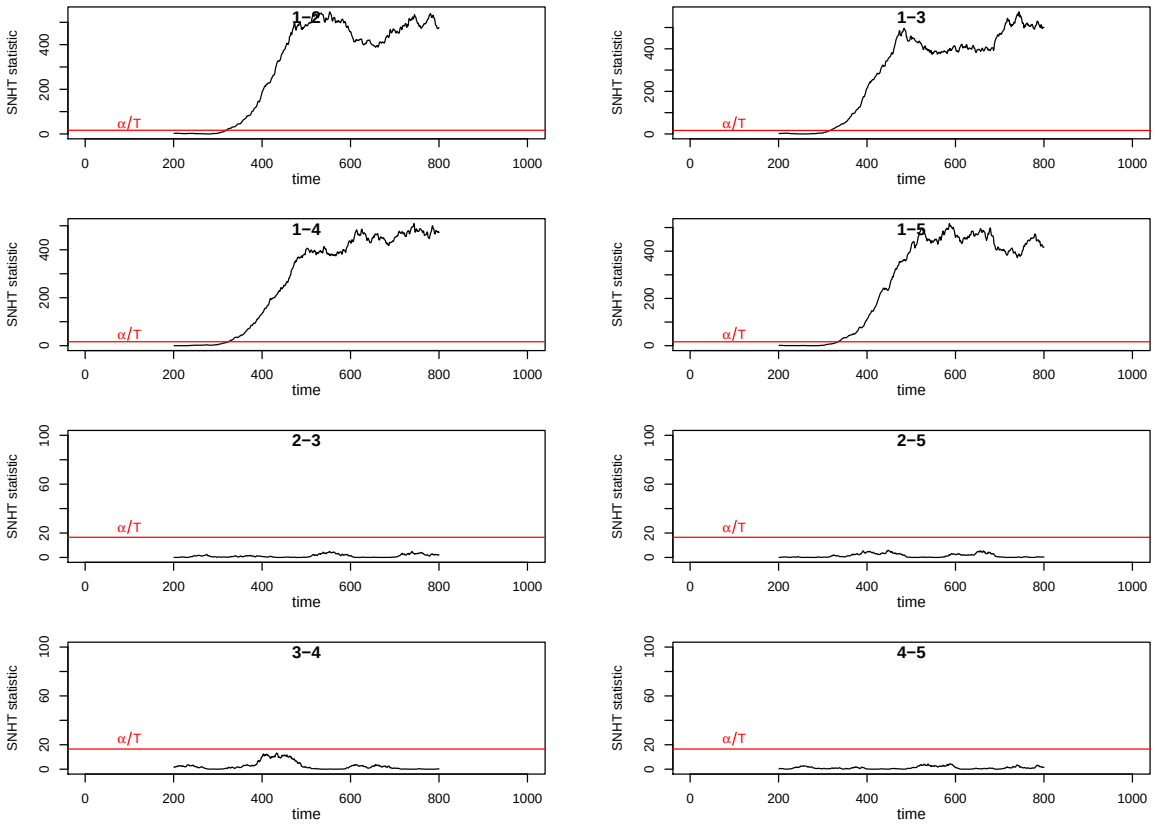

Figure 4.7: SNHT statistic for difference series of neighbor locations where the values of series 1 drift apart. Thereby, e.g., "1-2" refers to the difference series of time series 1 and time series 2.

It can be observed that the SNHT statistic increases to highly significant regions in cases of this slight drift of slope 1/120 in one series. The shape of the plot above might also give a user a visual indication if the exceedance of the SNHT threshold is more likely due to either a local drift or a shift in the mean level.

4.3.3 Global shifts and negatively correlated neighbor series

By definition, global shifts are not detected by `pairwiseSNHT()`. They could be detected by applying the `snht()` function to every time series individually but that is computationally expensive. In Section 8.3.3, better algorithms are provided for global shift detection. As long as there is no reasonably large shift in the mean level, `pairwiseSNHT()` does not detect negatively correlated series either. This is due to the definition of the SNHT statistic.

4.3.4 Summary of inhomogeneity detection performance

The Sections 4.3.1 – 4.3.3 have provided an insight into the performance of the `pairwiseSNHT()` function. This performance has been summarized in Table 4.1.

Inhomogeneity type	Performance of SNHT/ `pairwiseSNHT()`
local shifts in mean of single time series	++
local drifts	+ (good at finding the location, bad at detecting the time)
global shifts in mean	−−
negatively correlated neighbor series	−−

Table 4.1: Summary of the inhomogeneity detection performance of `pairwiseSNHT()`. ++: Very good performance, +: reasonable performance, −: poor performance, −−: extremely poor performance.

4.4 Empirical runtime estimation of `pairwiseSNHT()`

The focus of this chapter is on the `pairwiseSNHT()` R function as an inhomogeneity detection method. Before applying `pairwiseSNHT()` to CMIP5-ng data, a runtime analysis might be of interest to users. The runtime of `pairwiseSNHT()` depends on the various input parameters of `pairwiseSNHT()`. These input parameters are:

```
pairwiseSNHT(data,dist,k,period,crit,returnStat=T/F)

#k:          number of neighbor series that are used
#            as reference series for each candidate series
#period:     was denoted by N in the theory above,
#            i.e., it is the number of values that are
#            used to calculate the left and right mean
#crit:       the critical value of the statistics
#
#returnStat=TRUE/FALSE:
#if TRUE:    only the statistics is given back for each
#            difference pair
#if FALSE:   the location, time and shift as well as
#            the homogenized data is given back.
```

The runtime naturally depends on the data size but also on the input parameters `k`, `period` and `returnStat`.

`returnStat=TRUE`

If `returnStat=TRUE` then only the statistics are returned for each difference series. Runtime increases linearly for a growing number of time units as for each additional time unit the same statistics (including the calculation of $2N + 1$ means and the standard deviation over $2N$ values) have to be calculated for each difference series. For an increasing number of spatial pixels, the runtime increases approximately linearly as well. It is not an exact linear relation since it depends on how many pairwise differences of time series are generated with an additional pixel, which varies depending on the position of the pixel. Figure 4.8 shows what one may expect as runtime for a CMIP5-ng file of dimension $144 \times 72 \times 2772$, $k = 3$ and `period=200`.

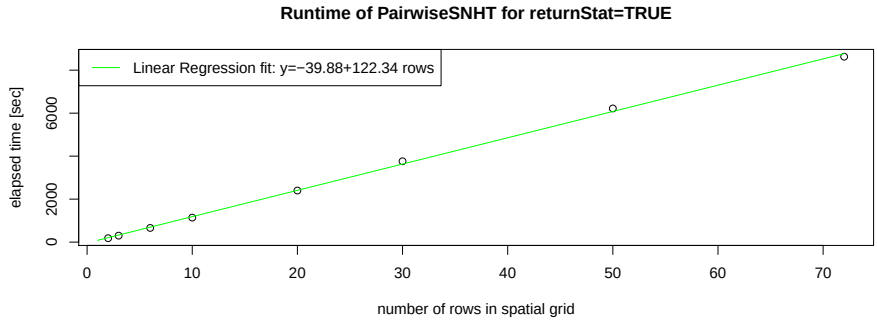

Figure 4.8: Runtime of `pairwiseSNHT()` for different numbers of spatial rows in a CMIP5-ng data set. The elapsed CPU time (in seconds) describes how long it took to execute `pairwiseSNHT()` for a CMIP5-ng file with a varying number of rows, 72 spatial columns and 2772 time units per time series.

As runtime has only been tested for 8 spatial regions, Figure 4.8 should be understood as an indicator for the runtime of `pairwiseSNHT()` for different sizes of data and not be confused with an analytically determined runtime calculation.
The input parameter `k` is thought to have an approximate linear effect on runtime. Runtime as a function of `k`, however, also depends on the shape of the spatial domain since a vertex or edge location has less closest neighbors than a pixel in the middle of the spatial field. An increasing value for `period` generally increases the runtime, but in a non-linear way. This is induced by the fact that the larger `period` is, the more values need to be considered when calculating the left and right mean. At the same time, only $T - 2N$ (with N representing the `period`) statistics need to be calculated which simultaneously reduces the cost. Figure 4.9 shows the non-linear effect of `period` on the runtime of `snht()` for a vector of random numbers of length 1000. The effect of a varying period on `pairwiseSNHT()` is then a multiple of the CPU time, shown in Figure 4.9, since it calls `snht()` multiple times (on each difference series), nonetheless, the relation between period and the CPU time stays less-than-linear.

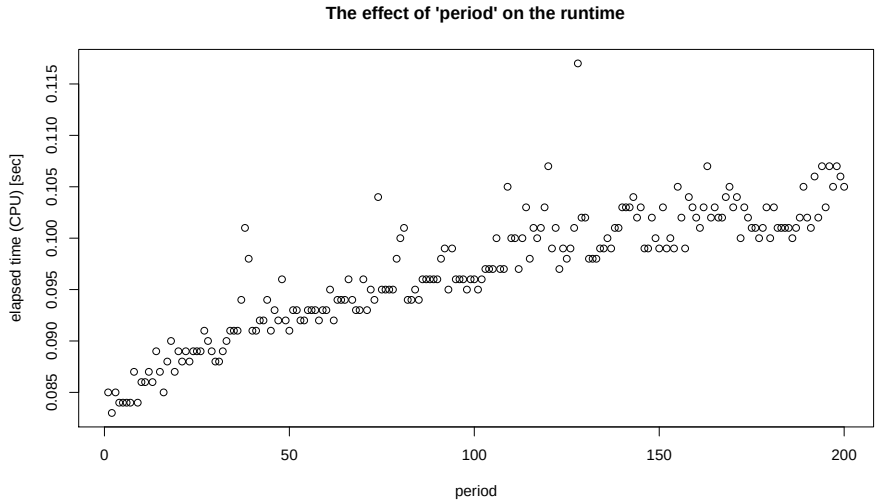

Figure 4.9: Runtime of `snht()` depending on `period` (denoted by N in equation (4.2)).

`returnStat=FALSE`

Setting `returnStat=FALSE` adds a lot of additional computational cost to the testing process since it implies that for each difference series the statistics not only need to be calculated, but it also needs to be checked at which time the statistic exceeds the threshold. Furthermore, it is deduced which spatial pixel is most probably responsible for the inhomogeneity. Applying `pairwiseSNHT()` with `returnStat=FALSE` on spatial fields of size 10×6 and 2772 time units as presented in Section 4.5 is reasonably fast and takes only about one minute of system CPU time. Applying it on a whole $144 \times 72 \times 2772$ CMIP5-ng file with `k=3,period=200`, however, takes about 26 hours, which is still rather long.

Space vs. time

Apart from the absolute runtime, a user may be interested in the question of "space vs. time" illustrated in Figure 4.10, i.e., one asks if it computationally faster to pass data on a large spatial field with only a few time units or if it is faster to pass data on a narrow spatial field with many time units. An experiment, comparing the `pairwiseSNHT()` runtime of a $3 \times 3 \times 100$ data set with the one of a $10 \times 10 \times 9$ data set, has suggested the latter. The experiment has been conducted on ACCESS1-0 (r1i1p1) Near Surface Temperature data under the RCP45 scenario (for code and exact output: see appendix, Section 8.1). In this particular example, it has been discovered that it is approximately 5 times faster to pass a $3 \times 3 \times 100$ data set to `pairwiseSNHT()` compared to a data set of dimension $10 \times 10 \times 9$.

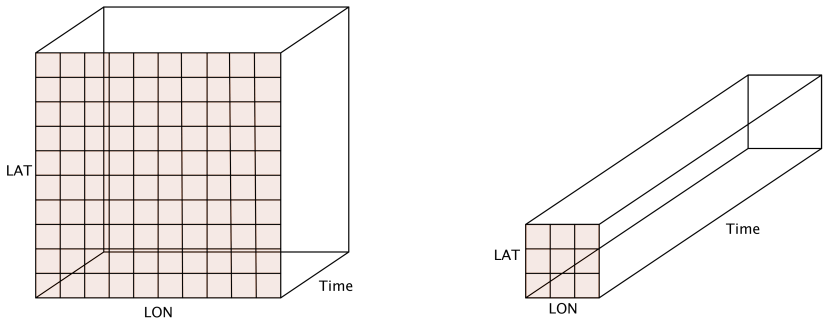

Figure 4.10: Left: Data block on a large spatial region with few time units. Right: Data block on a small spatial region with many time units.

4.5 SNHT methods on CMIP5-ng data

After giving a few ideas on the strengths and weaknesses of `pairwiseSNHT()`, users might be interested in finally seeing application results of the `pairwiseSNHT()` on CMIP5-ng data.

Monthly Near Surface Temperature over Europe (RCP45): ACCESS1-0 (r1i1p1)

As a first example, 60 locations over Europe have been chosen as displayed in Figure 4.11.

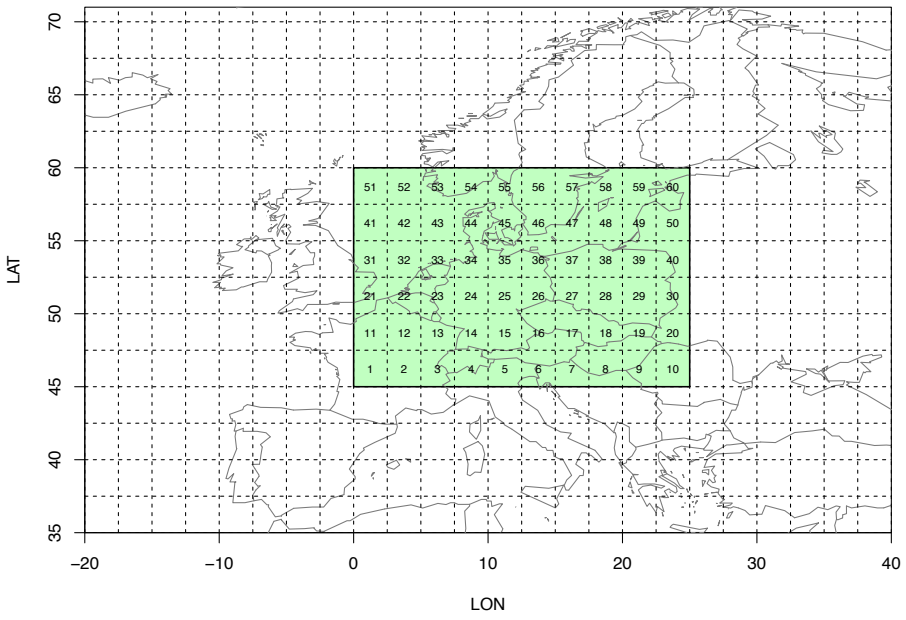

Figure 4.11: 60 2.5° × 2.5° pixels over Europe.

The monthly Near Surface Temperature projection of the ACCESS1-0 (r1i1p1) model is considered. The code below illustrates how a user can call the `pairwiseSNHT()` function and which preparation steps (such as the calculation of the distance matrix) are mandatory.

```
library(ncdf)
library(reshape2)
install.packages("/.../snht_1.0.4.tar.gz",type="src",repo=NULL)
library(snht)
source('/.../getCoord.R')
file <- '/.../tas_mon_ACCESS1-0_rcp45_r1i1p1_g025.nc'
nc <- open.ncdf(file)
data <- get.var.ncdf(nc)
close.ncdf(nc)

baseDataEurope <- data[c(1:10),c(55:60),]
coord <- getCoordinates(c(1:10),c(55:60))
#create coordinates
d <- as.matrix(dist(coord))

dim(baseDataEurope) <- c(dim(baseDataEurope)[1]*
                         dim(baseDataEurope)[2],dim(baseDataEurope)[3])
baseDataEurope <- t(baseDataEurope)
colnames(baseDataEurope) <- "1":"60"
baseData <- data.frame(time=1:2772,baseDataEurope)
baseData <- melt(baseData,id.vars="time",variable.name=
                 "location",value.name="data")
baseData$location <- gsub("X","",baseData$location)

system.time(out <- pairwiseSNHT(baseData,d,k=3,period=200,
                                crit=qchisq(1-0.05/2372,df=1),returnStat=F))
#    user  system elapsed
# 63.673   0.000  63.559      #---> runtime is within reason.
# > out$breaks
#   time location       shift
# 1  707       51  0.20400579
# 2  660       42 -0.01161282
# 3  911       51 -0.13019694
# 4  209       41 -0.05971085
# 5 1694       19  0.14797760
# 6  561       30  0.16337336
```

`out$breaks` gives information on the inhomogeneities that have been found by the `pairwiseSNHT()`. Overall, six spatial pixels with shifts have been detected. The shift heights are, nonetheless, of relatively low magnitude. One may notice that the location of these series form two spatial patches (both at the boundary of the spatial domain) of close-by neighbor series in Figure 4.11. In Figure 4.12, the difference series are depicted that have caused the threshold-exceeding values of the SNHT statistics.

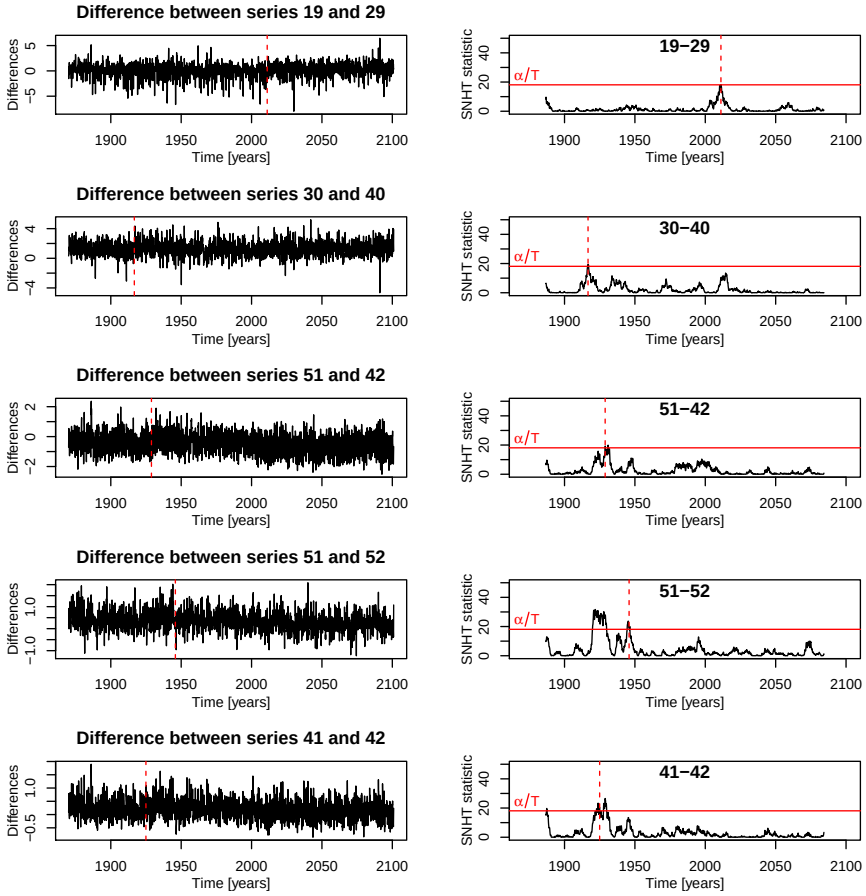

Figure 4.12: The difference series whose SNHT statistic exceeds the threshold. The red dashed line on the left corresponds to the detected break point. Left: Difference series itself, right: SNHT statistic over time of the corresponding difference series.

There are exceeding values in exactly 6 neighbor difference series ("41-42", "51-52", "51-42", "30-40" and "29-19") depicted in Figure 4.12. Whether or not the found location series really are erroneous would need to be investigated more closely, e.g., by analyzing other Near Surface Temperature model projections for the same period of time.

Monthly Surface Upwelling Longwave Radiation over South Africa (RCP45): CSIRO-Mk3L-1-2

Another example investigates the projection of the CSIRO-Mk3L-1-2 model for the Surface Upwelling Longwave Radiation under the RCP45 scenario on a larger spatial region (see Figure 4.13) over South Africa.

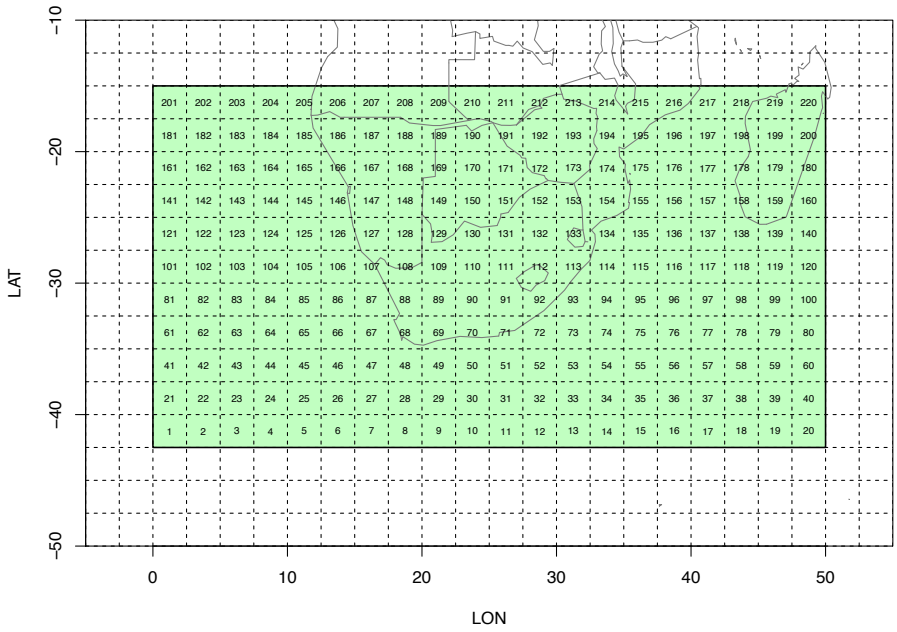

Figure 4.13: 220 2.5° × 2.5° pixels over South Africa.

The code that needs to be compiled is very similar to the previous example.

```
library(ncdf)
library(reshape2)
install.packages("/.../snht_1.0.4.tar.gz",type="src",repo=NULL)
library(snht)
source('/.../getCoord.R')

file <- '/.../rlus_mon_CSIRO-Mk3L-1-2_rcp45_r1i2p1_g025.nc'
nc <- open.ncdf(file)
data <- get.var.ncdf(nc)
close.ncdf(nc)

baseDataAfrica <- data[c(1:20),c(20:30),]
coord <- getCoordinates(c(1:20),c(20:30))
#create coordinates
d <- as.matrix(dist(coord))

dim(baseDataAfrica) <- c(dim(baseDataAfrica)[1]*
                         dim(baseDataAfrica)[2],dim(baseDataAfrica)[3])
baseDataAfrica <- t(baseDataAfrica)
colnames(baseDataAfrica) <- "1":"220"
baseData <- data.frame(time=1:2772,baseDataAfrica)
baseData <- melt(baseData,id.vars="time",variable.name=
                 "location",value.name="data")
baseData$location <- gsub("X","",baseData$location)

system.time(out <- pairwiseSNHT(baseData,d,k=3,period=200,
                                crit=qchisq(1-0.05/2372,df=1),returnStat=F))
```

```
# user     system elapsed
# 234.433   0.325 234.625 --->runtime is still within reason.
# > out$breaks
#     time location         shift
# 1   2060       45 -0.7836156845
# 2   2410      183 -0.0326845169
# 3   1272      182 -0.0170727921
# 4   1280      163 -0.1401939011
# 5   2471      123  0.0883020782
# 6   2267      143  0.0183229828
# 7   2477      142 -0.0264778519
# 8   2387      164  0.0058427048
...
# 383 2473       85  0.0609847641
# 384  491      163  0.0411592865
# 385 2497      156  0.2348451614
# 386 1343      176  0.1233944702
# 387  630       22 -0.0563951111
# 388 1548       22  0.0916710281
```

Unlike in the Near Surface Temperature example, considerably more inhomogeneities have been found by the `pairwiseSNHT()`, namely 388 shifts at 133 different locations. Figure 4.14 gives intriguing insight into which series are declared as inhomogeneous and the summed up absolute heights of the shifts are displayed.

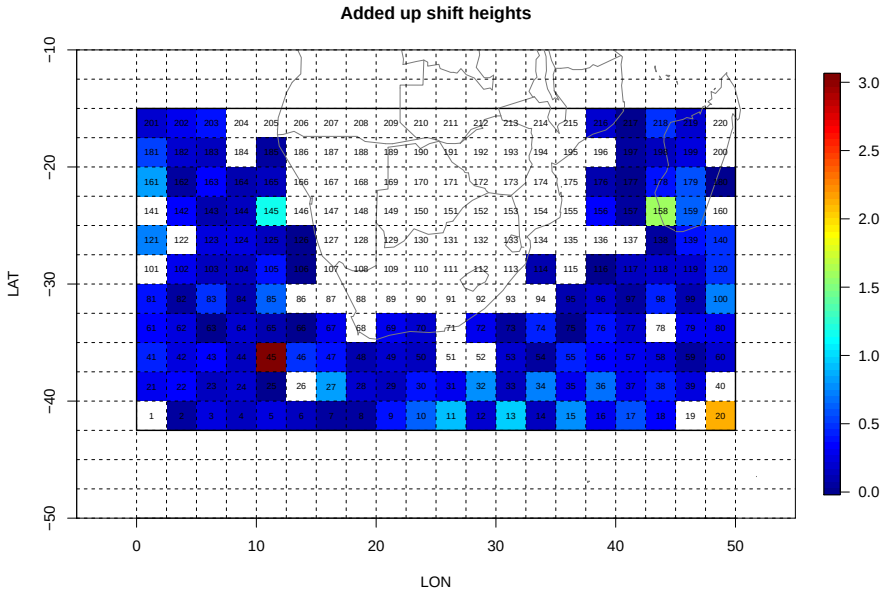

Figure 4.14: The added up absolute shift heights estimated by `pairwiseSNHT()` over Africa.

The plot reveals that only series over regions of sea have been declared as inhomogeneous. Reasons for this behavior would need to be given by climate model developers. A vague

presumption might be that existent measurements of Surface Upwelling Longwave Radiaton are imprecise over regions of sea and, therefore, simulations based on measurements come along with inherited errors.
The inhomogeneities, which have been detected by `pairwiseSNHT()` over the 220-pixel region over Africa, can also be investigated in a temporal dimension. The following histogram shows in which years the most inhomogeneities have been detected. All detected inhomogeneities have been weighted equally regardless of the estimated shift height.

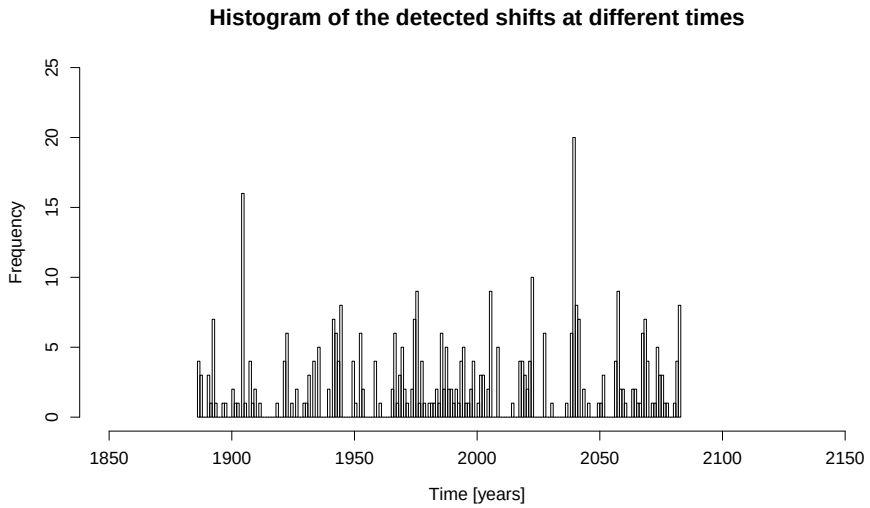

Figure 4.15: Frequency of inhomogeneities detected by `pairwiseSNHT()` over the spatial region depicted in Figure 4.13 from 1870 until 2100.

Figure 4.15 illustrates that the 388 inhomogeneities have been found across the whole interval from 1870 until 2100. Thereby, shortly after 1900 and before 2050 most inhomogeneities have been detected. These temporal regions may want to be further investigated. At this point, it might be interesting to see if smaller regions on the Southern Atlantic ocean show the same distribution as in Figure 4.15.
25 pixels in the South Atlantic Ocean are chosen as an example as depicted in Figure 4.16.

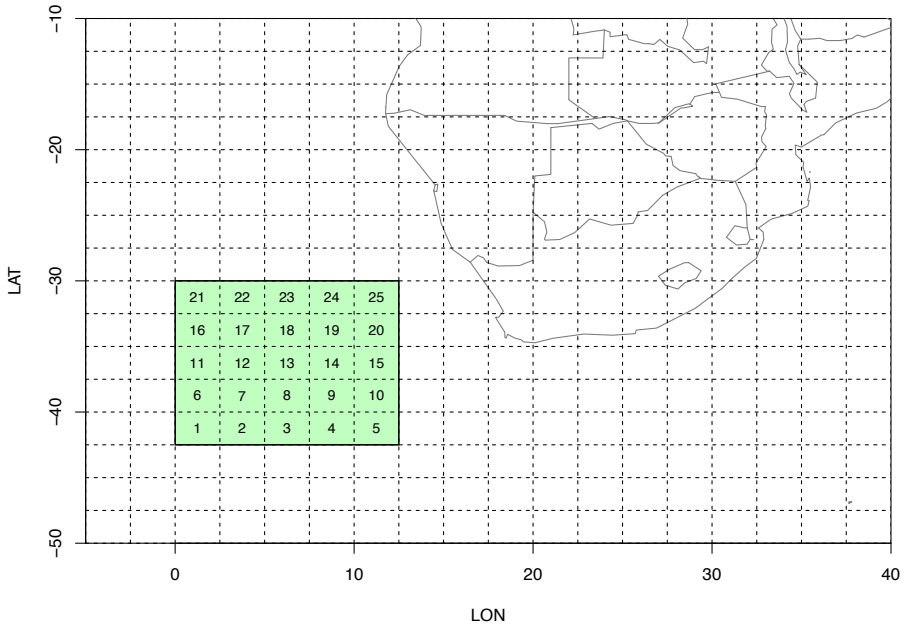

Figure 4.16: 25 2.5° × 2.5° pixels close to South Africa.

The histogram of the inhomogeneneities over the years 1870-2100 has again been plotted over this smaller spatial region.

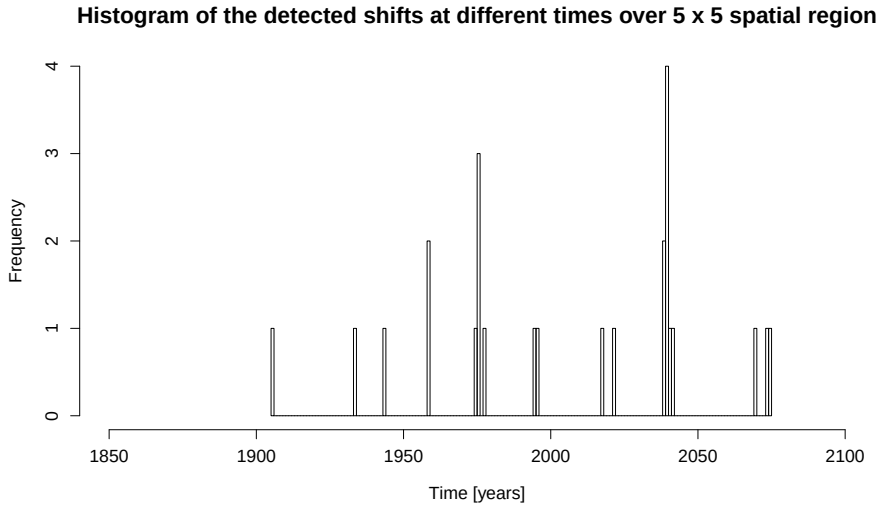

Figure 4.17: Frequency of inhomogeneities detected by `pairwiseSNHT()` over the spatial region depicted in Figure 4.16 from 1870-2100.

Compared to Figure 4.15, Figure 4.17 does not contribute largely to the 16 inhomogeneities detected shortly after 1900 but it contributes all the more to the 20 inhomogeneities detected shortly before 2050. In conclusion, even though the `pairwiseSNHT()` is not a suitable tool for global shift detection, histograms can be used to extract information about overall temporal inhomogeneities.

Chapter 5

GMRF

As mentioned in the previous chapter, the `pairwiseSNHT()` function cannot detect global shift inhomogeneities. In this chapter, homogenization tests are introduced using GMRFs which compensate for this deficiency. A MGMRF model includes the spatio-temporal structure of the data in a sparse precision matrix. The sparseness is especially useful for fast calculations of the determinant via the Cholesky factorization, which is, for instance, needed to determine the likelihood, find the MLEs and to do hypothesis testing.

5.1 Theory

The following theorems and definitions are based on the book by Rue and Held [2005].

Definition 5.1.1. *Two multivariate random vectors $\vec{x}, \vec{y}$ are called conditionally independent given $\vec{z}$, if $\pi(\vec{x}, \vec{y}|\vec{z}) = \pi(\vec{x}|\vec{z}) \cdot \pi(\vec{y}|\vec{z})$ for the generic density π.*

Definition 5.1.2. *An undirected graph $\mathcal{G}$ is a tuple $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes of the graph and $\mathcal{E}$ is the set of edges $\{i, j\}$, where $i, j \in \mathcal{V}, i \neq j$.*
If $\mathcal{V} = \{1, \ldots, n\}$ then the undirected graph is called labeled. Labeled and undirected graphs are used for the conditional independence structure in a GMRF.

5.1.1 Univariate GMRF

Definition 5.1.3. *A random vector $\vec{x} = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ is called a **GMRF** with respect to a labeled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, mean $\vec{\mu}$ and a SPD precision matrix Q, if its density has the form:*
$\pi(\vec{x}) = (2\pi)^{-n/2}|Q|^{1/2} \cdot \exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu})^T Q(\vec{x} - \vec{\mu})\right)$ *and*
$Q_{ij} \neq 0 \Leftrightarrow \{i, j\} \in \mathcal{E}, \forall i \neq j$

Definition 5.1.4. *The **precision** matrix Q is defined to be the inverse of the covariance matrix.*

Definition 5.1.5. *The neighbors of node i are all nodes in $\mathcal{G}$ having an edge connected with node i. One writes: $j \sim i$ if node i is directly connected through one edge to node j.*

Lemma 5.1.1. *For random variables x, y: $x \perp y|z \Leftrightarrow \pi(x, y, z) = f(x, z)g(y, z)$, for some functions f, g and $\forall z$ with $\pi(z) > 0$.*

Theorem 5.1.1. *Let $\vec{x}$ be a normally distributed random vector with mean $\vec{\mu}$ and positive definite precision matrix Q, then for $i \neq j$, $x_i \perp x_j|\vec{x}_{-ij} \Leftrightarrow Q_{ij} = 0$.*

Proof. Without loss of generality let $\vec{\mu} = 0$. By definition:
$\pi(x_i, x_j, \vec{x}_{-ij}) \propto \exp\left(-\frac{1}{2}\sum_{k,l} x_k Q_{kl} x_l\right)$
$\propto \exp\left(-\frac{1}{2}x_i x_j (Q_{ij} + Q_{ji}) - \frac{1}{2}\sum_{\{k,l\}\neq\{i,j\}} x_k Q_{kl} x_l\right)$
This term depends on $x_i, x_j \Leftrightarrow Q_{ij} \neq 0$, i.e.,
$\exists f, g : \pi(x_i, x_j, \vec{x}_{-ij}) = f(x_i, \vec{x}_{-ij}) g(x_j, \vec{x}_{-ij}) \Leftrightarrow Q_{ij} = 0$
Applying the previous Lemma 5.1.1 proves the Theorem. □

Hence, the (i, j)th entry of the precision matrix is nonzero if and only if x_i, x_j are conditionally dependent, i.e., the spatial structure is visually apparent in the precision matrix, which is useful for interpretation. For grid data as the CMIP5-ng, every data point has at most 4 spatial neighbors. Therefore, for a fixed time t_0, Q is a sparse matrix with at most 4 non-zero off-diagonals on each side. To calculate the determinant of Q, it is recommended to perform a Cholesky factorization. Thereby, it needs to be remembered that Q has been defined as a positive definite matrix, which guarantees that the Cholesky factor exists.

Theorem 5.1.2. *Let x be a GMRF with respect to the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with mean $\vec{\mu}$ and positive definite precision matrix Q, then:*

$$\begin{aligned} E(x_i|\vec{x}_{-i}) &= \mu_i - \frac{1}{Q_{ii}} \sum_{j:j\sim i} Q_{ij}(x_j - \mu_j) \\ Prec(x_i|\vec{x}_{-i}) &= Q_{ii} \\ Cor(x_i, x_j|\vec{x}_{-ij}) &= -\frac{Q_{ij}}{\sqrt{Q_{ii}Q_{jj}}}, i \neq j \end{aligned} \tag{5.1}$$

Remark 5.1.1. *This theorem allows to interprete the elements of the precision matrix as conditional correlations of x_i and x_j and conditional precisions of x_i.*

Often, the precision is also implicitly specified using full conditionals $\{\pi(x_i|\vec{x}_{-i})\}$.

$$\begin{aligned} E(x_i|\vec{x}_{-i}) &= \mu_i + \sum_{j:j\sim i} \beta_{ij}(x_j - \mu_j), \\ Prec(x_i|\vec{x}_{-i}) &= \kappa_i, \end{aligned}$$

with $\kappa_i > 0$, $\kappa_i \beta_{ij} = \kappa_j \beta_{ji}$, s.t.

$$Q_{ij} = \begin{cases} -\kappa_i \beta_{ij}, & \text{if } i \neq j, i \sim j, \\ \kappa_i, & \text{if } i = j, \\ 0, & \text{else,} \end{cases}$$

is SPD.
In practice, however, it would not make sense to specify the precision in such an over-parameterized manner, yielding high computational cost for estimation. E.g., κ_i and β_{ij} can be parameterized in the following way:

$$\begin{aligned} \kappa &\equiv \kappa_i, \forall i, \\ \beta &\equiv \beta_{ij}, \forall i \sim j. \end{aligned}$$

5.1.2 Multivariate GMRF

In climate data sets, not only the spatial neighbors are conditionally correlated but also temporal conditional correlation exists (illustrated in Figure 5.1).
Thus, it is useful to work with **multivariate** instead of univariate GMRFs.

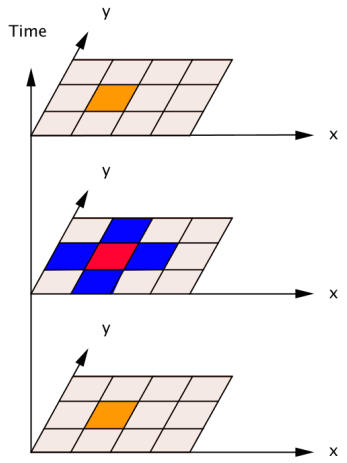

Figure 5.1: Spatio-temporal structure of grid data.

Definition 5.1.6. *A Multivariate GMRF is a generalization of the univariate GMRF from above.*
A random vector $\vec{x} = (\vec{x_1}^T, \ldots, \vec{x_n}^T)^T \in \mathbb{R}^{np}$ *where* $\dim(\vec{x_i}) = p$ *is called a* ***Multivariate Gaussian Markov Field*** *(*$MGMRF_p$*) wrt* $\mathcal{G} = (\mathcal{V} = \{1, \ldots, n\}, \mathcal{E})$ *with mean vector* $\vec{\mu} = (\vec{\mu_1}^T, \ldots, \vec{\mu_n}^T)^T \in \mathbb{R}^{np}$ *and positive definite precision matrix* $\tilde{Q}$*, iff its density has the form:*
$\pi(x) = \left(\frac{1}{2\pi}\right)^{\frac{np}{2}} |\tilde{Q}|^{\frac{1}{2}} \exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu})^T \tilde{Q}(\vec{x} - \vec{\mu})\right)$ *and* $\tilde{Q}_{ij} \neq 0_{p \times p} \Leftrightarrow \{i, j\} \in \mathcal{E}, \forall i \neq j$

Remark 5.1.2. *Every* $MGMRF_p$ *is also a GMRF with dimension* np*.*

Theorem 5.1.3. *Again* $x_i \perp x_j | \vec{x}_{-ij} \Leftrightarrow \tilde{Q}_{ij} = 0$

Proof. Follows immediately from Theorem 5.1.1. □

In climate data, temporal and spatial correlation should be taken into account, therefore, the MGMRF is suitable. In the above definition, n is chosen as the number of time units T and p is chosen to be n, where n is the number of locations (spatial $2.5° \times 2.5°$ pixels in CMIP5-ng).
Hence, the investigated MGMRF is the random vector
$\vec{x} = (x_{11}, \ldots, x_{n1}, x_{12}, \ldots, x_{n2}, \ldots, x_{1T}, \ldots, x_{nT})^T$ and its precision matrix should have the following block matrix form:

$$\tilde{Q}_{Tn \times Tn} = \begin{pmatrix} \tilde{Q}_{11} & \tilde{Q}_{12} & 0 & \ldots & 0 \\ \tilde{Q}_{21} & \tilde{Q}_{22} & \tilde{Q}_{23} & \ldots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \tilde{Q}_{T-1,T-2} & \tilde{Q}_{T-1,T-1} & \tilde{Q}_{T-1,T} \\ 0 & \cdots & 0 & \tilde{Q}_{T,T-1} & \tilde{Q}_{TT} \end{pmatrix},$$

where each $\tilde{Q}_{ij}$ is an $n \times n$ matrix (for $i, j \in \{1, \ldots, T\}$).

Remark 5.1.3. *$\tilde{Q}_{ij} = 0_{n\times n}$ with $|i-j|> 1$ for $i,j \in \{1,\ldots,T\}$ are then not directly adjacent in time and therefore thought to be conditionally independent.*

Definition 5.1.7. *$\tilde{Q}_{ii}$ is called the **spatial block** since it contains the spatial conditional correlations.*
*$\tilde{Q}_{ij}, i \neq j, |i-j|= 1$ is called the **temporal block** since it contains the temporal conditional correlations.*

Again, one can specify the precision matrix with full conditionals:

$$\begin{aligned}
E(\vec{x}_t|\vec{x}_{-t}) &= \vec{\mu}_t + \sum_{u:u\sim t}(\beta_{tu})_{n\times n}(\vec{x}_u - \vec{\mu}_u), t,u \in \{1,\ldots,T\},\\
E(x_{it}|\vec{x}_{-it}) &= \mu_{it} + \sum_{j:j\sim i}\beta^{sp}_{ij}(x_{jt}-\mu_{jt}) + \sum_{u:u\sim t}\beta^{te}_{tu}(x_{iu}-\mu_{iu}),\\
&\quad i,j \in \{1,\ldots n\}, t,u \in \{1,\ldots,T\},\\
Prec(\vec{x}_t|\vec{x}_{-t}) &= (\kappa_t)_{n\times n}, \text{call } diag(\kappa_t) = (\kappa^{(t)}_1,\ldots,\kappa^{(t)}_n).
\end{aligned}$$

Then, the spatial block of the precision matrix has the following form:

$$\tilde{Q}_{ii} = \begin{pmatrix}
\kappa^{(i)}_1 & -\kappa^{(i)}_1\beta^{sp}_{12} & 0 & \ldots & 0 & -\kappa^{(i)}_1\beta^{sp}_{1l} & 0 & \ldots & 0\\
-\kappa^{(i)}_2\beta^{sp}_{21} & \ddots & \ddots & 0 & \ddots & 0 & \ddots & \ddots & \vdots\\
0 & \ddots & \ddots & \ddots & 0 & \ddots & \ddots & \ddots & 0\\
\vdots & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & -\kappa^{(i)}_k\beta^{sp}_{kn}\\
0 & \ddots & 0 & \ddots & \ddots & \ddots & 0 & \ddots & 0\\
-\kappa^{(i)}_m\beta^{sp}_{m1} & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & \vdots\\
0 & \ddots & \ddots & \ddots & 0 & \ddots & \ddots & \ddots & 0\\
\vdots & \ddots & \ddots & 0 & \ddots & 0 & \ddots & \ddots & -\kappa^{(i)}_{n-1}\beta^{sp}_{n-1,n}\\
0 & \ldots & 0 & -\kappa^{(i)}_n\beta^{sp,i}_{n,p} & 0 & \ldots & 0 & -\kappa^{(i)}_n\beta^{sp}_{n,n-1} & \kappa^{(i)}_n
\end{pmatrix}.$$

The temporal block looks as follows:

$$\tilde{Q}_{ij} = \begin{pmatrix}
-\kappa^{(i)}_1\beta^{te}_{ij} & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0\\
0 & \ddots & \ddots & 0 & \ddots & 0 & \ddots & \ddots & \vdots\\
0 & \ddots & \ddots & \ddots & 0 & \ddots & \ddots & \ddots & 0\\
\vdots & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & 0\\
0 & \ddots & 0 & \ddots & \ddots & \ddots & 0 & \ddots & 0\\
0 & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & \vdots\\
0 & \ddots & \ddots & \ddots & 0 & \ddots & \ddots & \ddots & 0\\
\vdots & \ddots & \ddots & 0 & \ddots & 0 & \ddots & \ddots & 0\\
0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 & -\kappa^{(i)}_n\beta^{te}_{ij}
\end{pmatrix}.$$

For the sake of simplicity $\kappa^{(i)}_j, \beta^{sp}_{ij}$ and β^{te}_{ij} can again be parameterized as follows:

$$\begin{aligned}
\kappa &\equiv \kappa^{(i)}_j,\\
\beta^{sp} &\equiv \beta^{sp}_{ij},\\
\beta^{te} &\equiv \beta^{te}_{ij}.
\end{aligned}$$

Then, the temporal and spatial block can be expressed as:

$$
\begin{aligned}
\tilde{Q}_{ij} &= \kappa I_n, \\
\tilde{Q}_{ii} &= (I_n - \beta^{sp} A)\kappa, \text{ where} \\
A_{ij} &= \begin{cases} 1, \text{if } i \sim j \text{ in the spatial field} \\ 0, \text{else} \end{cases} \\
&\quad \text{A is called the adjacency matrix since} \\
&\quad \text{it gives information on the neighboring structure of the spatial grid.}
\end{aligned}
$$

The whole precision matrix $\tilde{Q}$ can be calculated using the Kronecker product [Van Loan, 2000]:
$\tilde{Q} = (I_T \otimes \tilde{Q}_{ii} + C \otimes \tilde{Q}_{ij})$, with $j : |i - j| = 1$, where
$C_{ij} = \begin{cases} 1, \text{if } |i - j| = 1, \\ 0, \text{else.} \end{cases}$

5.2 MGMRF model

GMRF models can be set up by different parametrizations of the precision matrix. Schibli [2011] has introduced the following "Model 1" parametrization for $\tilde{Q}$:

Model 1

$$
\begin{aligned}
\kappa_i^{(t)} &= 1, \forall i \in \{1, \ldots, n\}, t \in \{1, \ldots, T\} \\
\beta_{ij}^{te} \kappa_i^{(t)} &= \beta_{ij}^{te} = f \\
\beta_{ij}^{sp} \kappa_i^{(t)} &= \beta_{ij}^{sp} = b \\
c &\quad \text{a scaling parameter for the precision}
\end{aligned} \tag{5.2}
$$

Schibli [2011] approximated the valid parameter space for this Model 1. A valid parameter space is thought to be the set of parameters that result in a SPD precision matrix. Schibli [2011] also produced the Figure 5.2, which depicts her approximation of the valid parameter space depending on the parameters b and f.

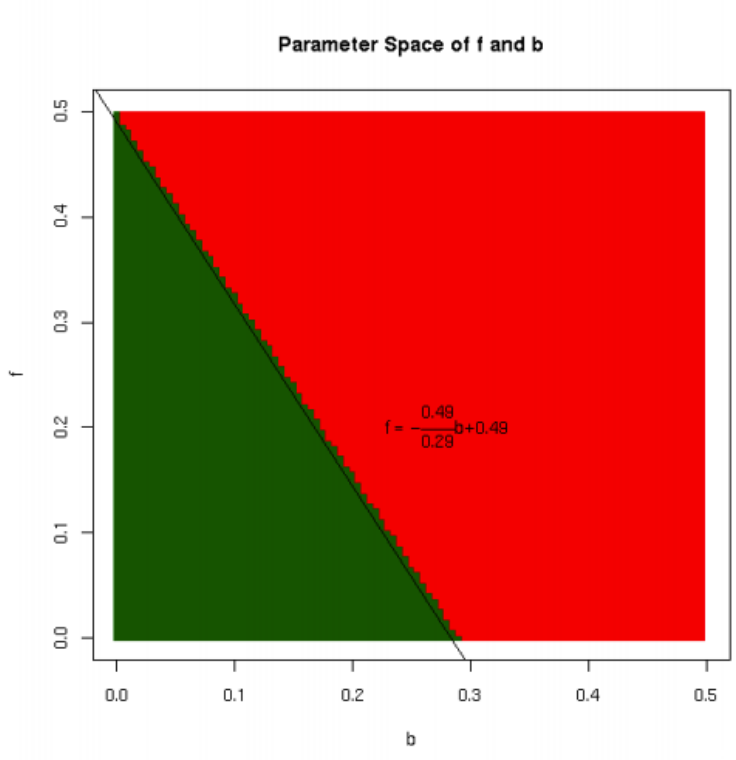

Figure 5.2: Parameters b and f that result in a SPD precision matrix (green) and parameters resulting in a symmetric but non-positive-definite precision matrix (red) [Schibli, 2011].

One can observe that the valid parameter space does not allow high conditional temporal or spatial correlations. Yet, Schibli [2011] has shown that the model is still practicable and should not be condemned, even though in climate data (with high temporal and spatial correlation) the true parameters tend to lie within the non-valid parameter space.

5.2.1 Model based hypothesis testing with MGMRF

Finally, the null and alternative hypotheses for inhomogeneity testing can be stated based on Model 1 parametrization, which has been introduced above. Thereby, $i, j \in \{1, \ldots, n\}$ represent spatial locations.

$$
\begin{aligned}
H_0 \quad &: \quad \mu_{it} \equiv \mu_0, \forall t \in \{1, \ldots, T\} \\
&\qquad \text{But: } \mu_{it} \neq \mu_{jt}, \text{ for } i \neq j \\
H_1 \quad &: \quad \begin{cases} \mu_{it} \equiv \mu_0, \forall t \in \{1, \ldots, t_0\} \\ \mu_{it} = \mu_0 + a(t), \forall t \in \{t_0 + 1, \ldots, T\} \\ \text{But for other series } \mu_{jt_0} = \mu_{jt}, t \in \{t_0 = 1, \ldots, T\}, \\ \text{i.e., other series may be homogeneous.} \end{cases}
\end{aligned}
$$

Hereby, t_0 is the time at which the change point occurred.
If $a(t) \equiv a_0, \forall t \in \{t_0 + 1, \ldots, T\}$ then it is called a "shift", if it varies with time, then one calls it a "drift".

Distribution under H_0 and H_1
The vector $\vec{x} := (x_{11}, \ldots, x_{n1}, \ldots, x_{1T}, \ldots, x_{nT})^T \in \mathbb{R}^{nT}$ as a MGMRF is clearly multivariate normally distributed, i.e.,

$$\vec{x} \sim \mathcal{N}_{Tn}\left(\begin{bmatrix}\mu_{11}\\ \vdots \\ \mu_{n1}\\ \vdots \\ \vdots \\ \mu_{1T}\\ \vdots \\ \mu_{nT}\end{bmatrix}, \begin{pmatrix} \tilde{Q}_{11} & \tilde{Q}_{12} & 0 & \ldots & 0 \\ \tilde{Q}_{21} & \tilde{Q}_{22} & \tilde{Q}_{23} & \ldots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & & \tilde{Q}_{T-1,T-2} & \tilde{Q}_{T-1,T-1} & \tilde{Q}_{T-1,T} \\ 0 & \cdots & 0 & \tilde{Q}_{T,T-1} & \tilde{Q}_{TT} \end{pmatrix}^{-1}\right).$$

Under H_0:
$\mu_{i,1} = \mu_{i,2} = \cdots = \mu_{i,T}, \forall i \in \{1, \ldots, n\}$

Under H_1:
$\mu_{i,1} = \mu_{i,2} = \cdots = \mu_{i,t_0-1}, \forall i$ and
$\mu_{i,t} = \mu_{i,t_0-1} + a(t), \forall t \in \{t_0, \ldots, T\}$ and for some $i \in \{1, \ldots, n\}, t_0 \in \{1, \ldots, T\}$

Test statistic
The likelihood ratio test statistic W provides a mean to compare the likelihood under the alternative hypothesis with the likelihood under the null hypothesis (see (5.3)). The statistic follows an approximate χ^2_1- distribution since there is only one additional parameter to be estimated under H_1 than under H_0. In order to find the MLEs of the parameters under H_1 and H_0, the minimization R function

```
optim()
```

has been employed. `optim()` – i.e., minimization – is applied to twice the negative log-likelihood. The $\vec{\theta}$ which minimizes $-2l_{H_0}(\vec{\theta})$ or $-2l_{H_1}(\vec{\theta})$, maximizes $2l_{H_0}(\vec{\theta})$ or $2l_{H_1}(\vec{\theta})$ respectively, i.e., one can find the MLE by applying `optim()`. The following transformations illustrate what the likelihood and likelihood ratio statistics look like and what to expect regarding their distributions:

$$
\begin{aligned}
W &:= 2\log\left(\frac{L(\hat{\vec{\theta}}_{\mathrm{ML}}|H_1)}{L(\hat{\vec{\theta}}_{\mathrm{ML}}|H_0)}\right) \qquad (5.3)\\
&= 2\cdot(l_{H_1}(\hat{\vec{\theta}}_{\mathrm{ML}}) - l_{H_0}(\hat{\vec{\theta}}_{\mathrm{ML}}))\\
&= (-2l_{H_0}(\hat{\vec{\theta}}_{\mathrm{ML}})) - (-2l_{H_1}(\hat{\vec{\theta}}_{\mathrm{ML}})) \overset{approx}{\sim} \chi^2_1, \text{ where}\\
L(\vec{\theta}) = L(\vec{\mu}, b, c, f) &= \left(\frac{1}{2\pi}\right)^{\frac{Tn}{2}} |\tilde{Q}|^{1/2}\exp\left(-\frac{1}{2}(\vec{x}-\vec{\mu})^T\tilde{Q}(\vec{x}-\vec{\mu})\right) \text{ and}\\
-2\cdot l_{H_0}(\vec{\mu}, b, c, f) &= nT\log(2\pi) - \log(|\tilde{Q}|) + (\vec{x}-\vec{\mu})^T\tilde{Q}(\vec{x}-\vec{\mu})\\
-2\cdot l_{H_1}(\vec{\mu}, b, c, f, a) &= nT\log(2\pi) - \log(|\tilde{Q}|) +\\
&\quad (\vec{x}-\vec{\mu}-a\vec{d})^T\tilde{Q}(\vec{x}-\vec{\mu}-a\vec{d})\\
\vec{d} &= (0,0,\ldots,0,\ \underbrace{1,\ldots,1}_{(T-t_0+1)\text{ times}},0,\ldots,0) \in \mathbb{R}^{nT}
\end{aligned}
$$

5.2.2 Validation of the MGMRF model through simulation runs

Before applying the R functions to the CMIP5-ng data, it might be interesting to analyze the estimated parameters $\hat{b}_{\mathrm{ML}}, \hat{c}_{\mathrm{ML}}, \hat{f}_{\mathrm{ML}}$ and the inhomogeneity detection behavior in general. In this section, the performance of the convergence as well as the sensitivity of the homogeneity test is analyzed with a set of samples drawn from a MGMRF. Simulated data sets are useful in this setting since true parameter values are known and, thus, one can examine the performance of the test based on the amount of deviations from the estimates to the true values of the parameters.
The MGMRF sampling has been done with the `rmvnorm.prec()` R function from the `spam` R package [Furrer, 2015]. The samples are thereby generated on a 3×3 spatial grid and $T = 50$ time values in each time series. A single sample (of dimension $3 \times 3 \times 50$) from the MGMRF can then be generated as follows:

```
library(spam)
source('/.../mgmrfPrec.R')
b <- 0.2
c <- 1
f <- 0.1
T <- 50 #time steps
N <- 9 #9 spatial locations
mu <- rep(1,9)
Q <- mgmrf.prec(b,c,f,row,col,T)
#the code of the function mgmrf.prec()
#can be found in the appendix
set.seed(1)
x <- rmvnorm.prec(1, mu = mu, Q) #x has the dimension 1 x 450
y <- x
dim(y) <- c(N,T) #bringing into right format
#.....................
#or alternatively by the function dataGenerator.R (see appendix)
source('/.../dataGenerator.R')
y <- dataGenerator(0.2,1,0.1,3,3,50)
```

The MGMRF precision matrix thereby has dimension 450×450 (9 locations $\times$ 50 time units) and looks as follows:

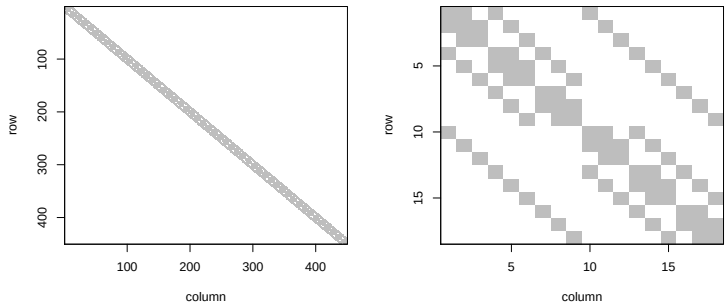

Figure 5.3: MGMRF precision matrix of dimension 450×450. Left: Whole precision matrix, right: 2 temporal and 2 spatial blocks enlarged.

Parameter estimation of Model 1 under H_0

Firstly, the parameter estimation of Model 1 under the null hypothesis is analyzed. This is done with 200 samples from the MGMRF without introducing inhomogeneities. Since one knows the true value of the parameters $\vec{\theta} = (\vec{\mu}^T, b, c, f)^T$, the performance of the estimation can be analyzed. A histogram and scatter plot show how the `optim()` R function estimates $\vec{\theta} = (\vec{\mu}^T, b, c, f)^T$ under H_0 with 200 replicates generated as above using `rmvnorm.prec()`. The starting values for `optim()` are thereby chosen as described in Figure 5.4.

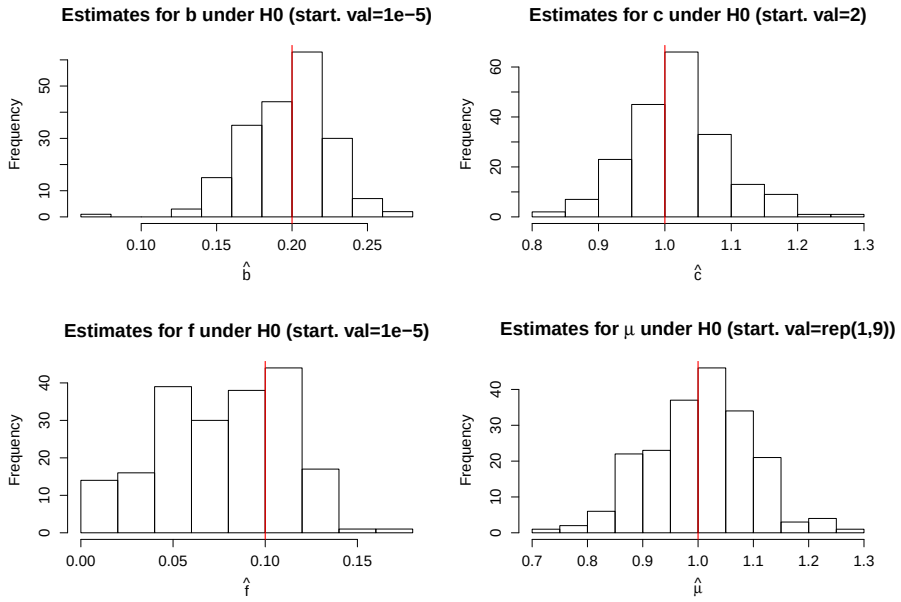

Figure 5.4: Estimates for $\vec{\theta}$ under H_0 and non-perfect starting values for `optim()` for 200 samples. The red line corresponds to the true value. For the vector $\hat{\vec{\mu}}$, the average of its components are depicted in the last histogram.

Figure 5.4 shows that the estimates approach the real value closely. Only the estimates for f show a larger spread around the true value. Apart from a histogram, it might be interesting to look at the distribution of the estimates in a scatter plot in order to exclude the possibility of correlation among the parameters.

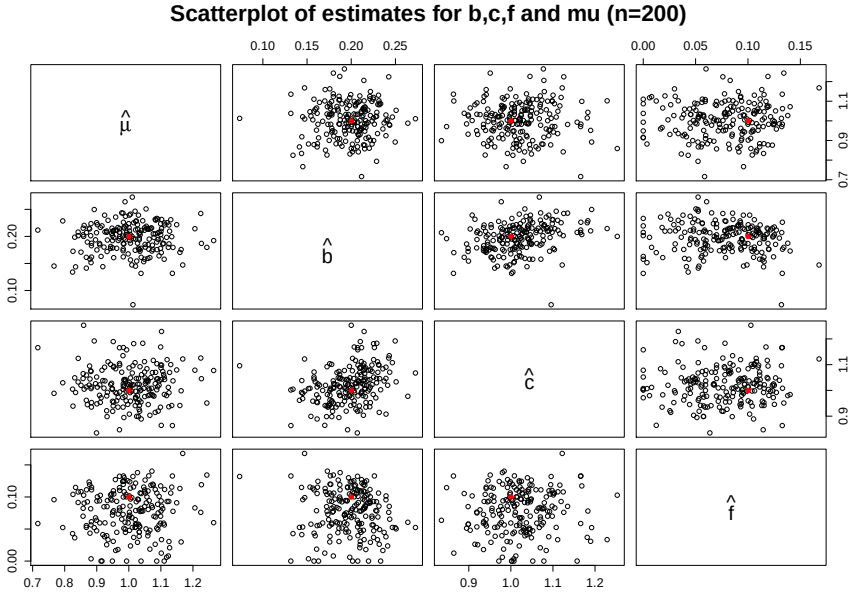

Figure 5.5: Estimates for $\vec{\theta}$ under H_0 and non-perfect starting values and a sample size of $n = 200$. The red point corresponds to the real value.

Figure 5.5 shows satisfying results since most of the estimates lie close to the real value with weak correlations among the parameters.

Parameter estimation of Model 1 under H_1

Similar checks as for the null hypothesis can be done for the MGMRF model under the alternative hypothesis. The alternative hypothesis states that there is some type of an inhomogeneity in the data. Below, different scenarios of inhomogeneities are simulated and the performance of the functions `test.H1Glob()` and `test.H1Loc()` (see appendix) are put to the test.

Global shifts

In order to simulate a global shift, the data matrix $y \in \mathbb{R}^{N \times T}$ (N: number of locations, T: number of time units) (y is again produced as illustrated on page 46) has been modified in the following manner.

```
y[,t0:T] <- y[,t0:T]+a
#a: amount of global shift at time t0
```

If one chooses $a = 0.5$ as true magnitude for the global shift, then the estimates for $\vec{\theta} = (\vec{\mu}^T, b, c, f, a)^T$ under H_1 for 200 replicates look as depicted in Figure 5.6.

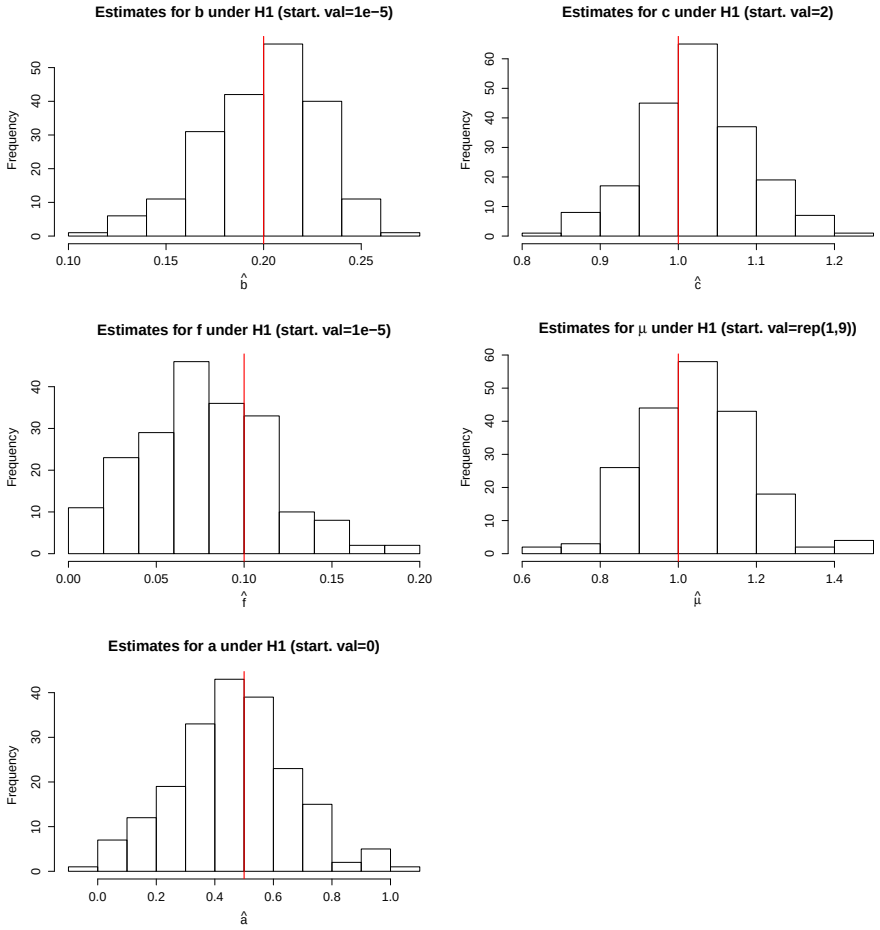

Figure 5.6: Estimates for $\vec{\theta}$ under H_1 with non-perfect starting values for `optim()` and a sample size of $n = 200$. The red lines correspond to the true value of the parameters.

Apart from histograms, one might again be interested in the distribution of the parameter estimates, which is shown in the scatter plot Figure 5.7:

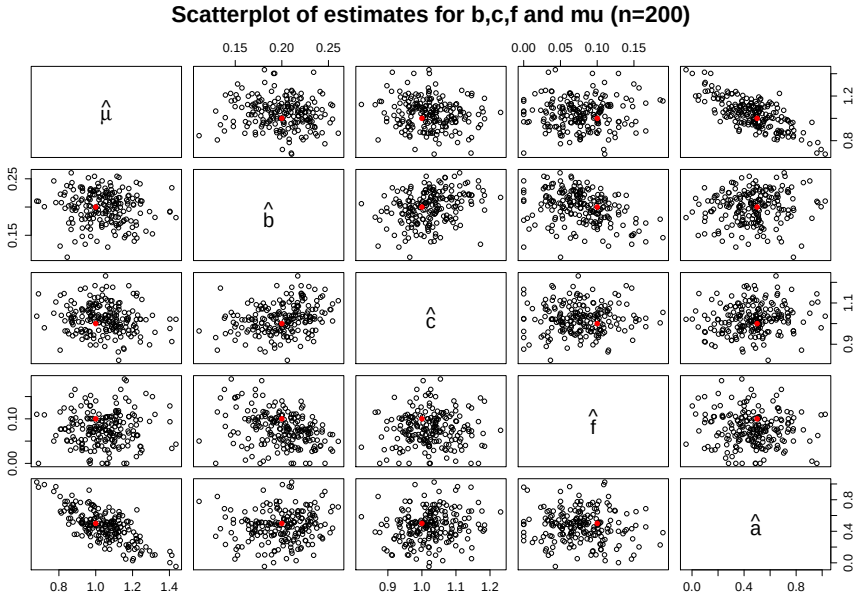

Figure 5.7: Estimates for $\vec{\theta}$ under H_1 and non-perfect starting values and a sample size of $n = 200$. The red point corresponds to the real value.

The scatter plot shows similar low correlations among $\vec{\mu}, b, c, f$ as before. Unlike before, $\hat{a}$, i.e., the shift height, is included in the analysis and shows a strong negative correlation with $\hat{\mu}$. This high correlation is desired and valid since the higher μ is, (i.e., the mean level of the time series before the global shift) the lower a needs to be in order to reach the same mean level after the global shift.
Apart from estimating the parameters, it might be interesting to look at the likelihood ratio statistic for different heights of global shifts and times $t \in \{1, \ldots, T\}$ in order to estimate the sensitivity of the GMRF homogeneity test. One should again remember that multiple testing is conducted. Therefore, the significance level has been Bonferroni adjusted, i.e., α/T is chosen as a significance level.

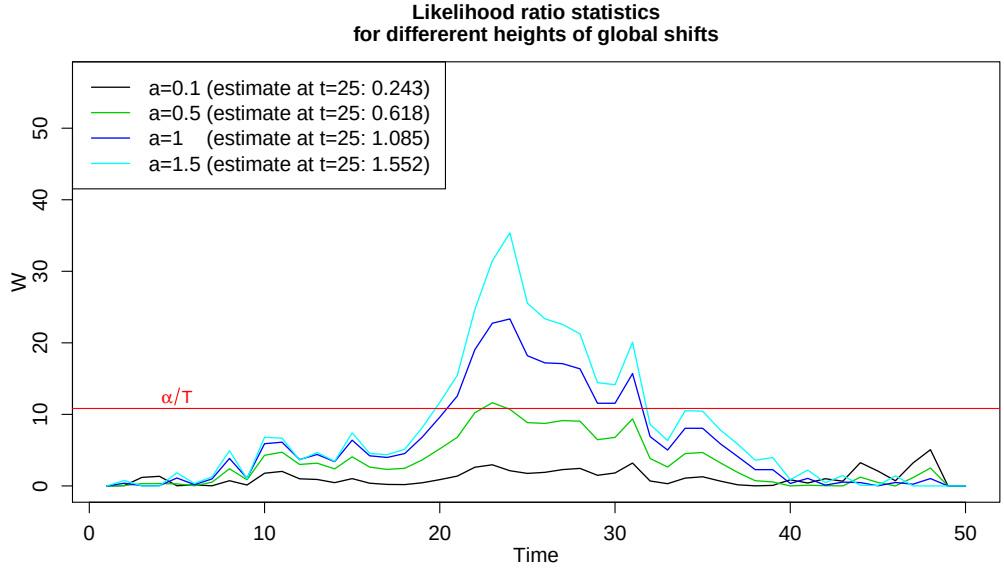

Figure 5.8: Likelihood ratio statistic for different heights a of global shifts at time $t_0 = 25$. $\hat{a}$ is the estimated height of the inhomogeneity at $t_0 = 25$.

Figure 5.8 indicates that the estimation of the shift magnitudes is fairly accurate over all simulation runs. The null hypothesis, however, is only rejected for the rather large shift height of $a = 1$, i.e., the test is not extremely sensitive since the deviation makes up approximately 0.8 (slightly deviates depending on the location of the series) of the standard deviation of each of the 9 series. The location of the shift heights a has been estimated well for a shift height of 0.5 or larger.

Local shifts

Apart from global shift detection, the `test.H1Loc()` function (see appendix) also makes local shift detection possible with the GMRF model developed. For this purpose, the likelihood ratio statistic is not only being evaluated at every point in time but also every spatial location.

Similarly as before, the performance of `test.H1Loc()` can be analyzed through a simulation example with 9 locations (on a 3×3 grid) and 50 time units. The same basis data has been used as in the global shifts simulation above, in order to allow comparison. This time, shifts of different magnitudes have been introduced in series 5, which is the location in the middle of the 3×3 spatial grid, i.e., the data matrix $y \in \mathbb{R}^{N \times T}$ of MGMRF samples have been adjusted as illustrated in this pseudo code:

```
l0 <- 5
t0 <- 25
changes <- c(0.1,0.5,1,1.5,2)
ynew <- y
for(i in 1:length(changes)){
  ynew[l0,t0:T] <- y[l0,t0:T]+changes[i]
}
```

Again multiple testing is conducted over space and time and the significance level is Bonferroni corrected. The likelihood ratio statistic values are depicted in Figure 5.9.

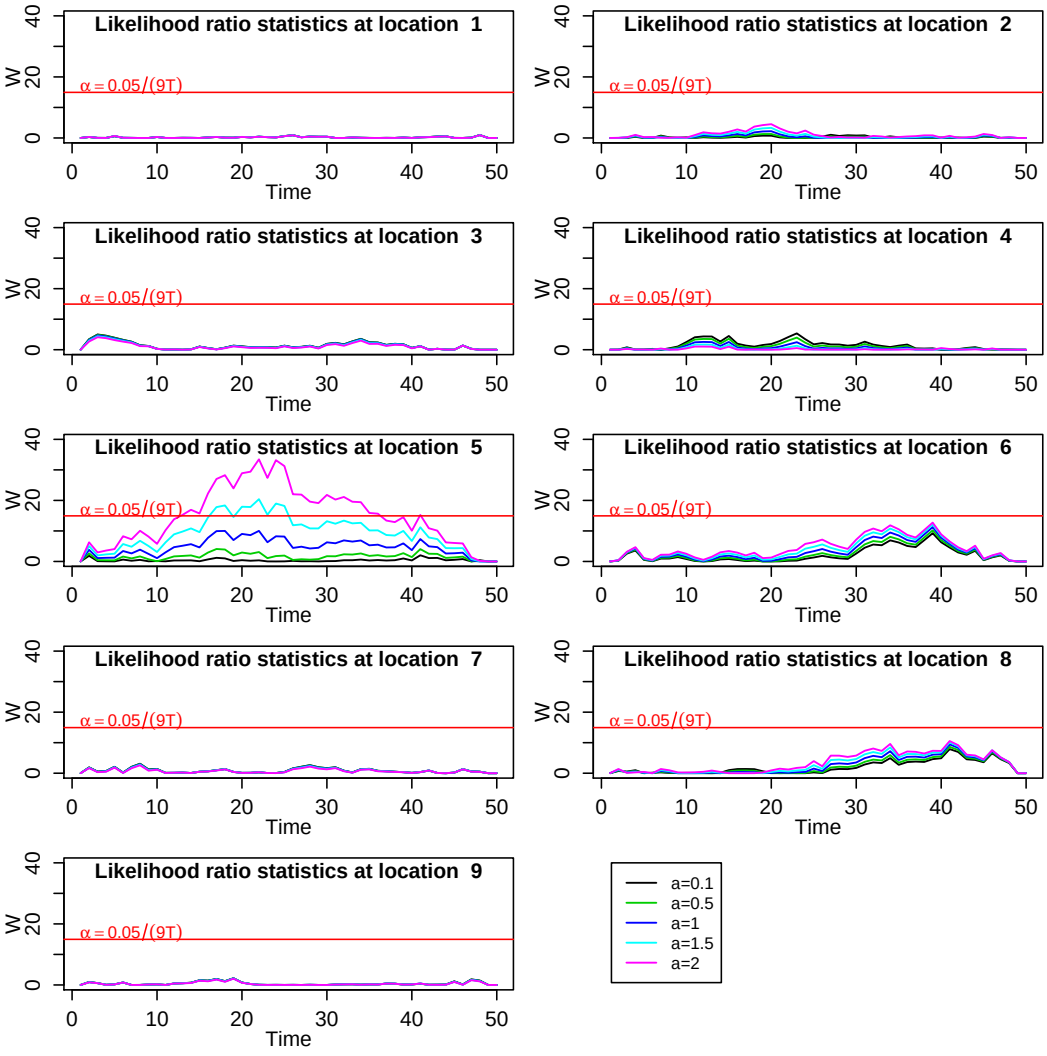

Figure 5.9: Likelihood ratio statistic for local shifts $a \in \{0.1, 0.5, 1, 1.5, 2\}$ at location 5 and time $t_0 = 25$.

The plots in Figure 5.9 show that local shifts are less likely to be detected compared to global shifts as depicted in Figure 5.8. In Figure 5.8, the global shift of $a = 0.5$ has been detected whereas in the Figure above, the green line corresponding to $a = 0.5$ does not show any significant values of the likelihood ratio statistic. This comparison is admissible since the same underlying data has been used for the simulations in the *Global shifts* and *Local Shifts* sections above.
Apart from the difference in sensitivity, the above Figure 5.9 illustrates the GMRF properties of conditionally dependent neighbors. Even though the series at locations 2, 4, 6 and 8 have been left unchanged over the course of increasing shift heights at location 5, their likelihood ratio statistics increase.

5.3 Inhomogeneity detection performance

5.3.1 Local and global shifts of the mean

Section 5.2.2 has shown that the MGMRF model succeeds in detecting local and global shifts for some samples of simulated data with large enough shifts introduced. One might be curious if the MGMRF model also succeeds at detecting local drifts or negatively correlated series (as depicted in Section 1.3.3).

5.3.2 Local drifts and negatively correlated neighbor series

For this purpose, one can again simulate data with a local drift in one pixel and see if `gmrfHomogeneityTestComp()` can detect this pixel. Below, data is generated using the function `dataGenerator()` (see appendix) on a 3×3 spatial grid with $T = 50$ time steps and a precision matrix with parameters set to $b = 0.2$, $c = 1$ and $f = 0.1$. A linear drift of slope $2/25$ is introduced in the time series at the location in the middle that starts at time $t_0 = 25$. The result is the following:

```
data <- dataGenerator(rep(0,9),0.2,5,0.1,3,3,50)
dataNewDrift <- data
dataNewDrift[2,2,25:50] <- 2/25*c(25:50)-2+data[2,2,25:50]
driftout <- gmrfHomogeneityTestComp(dataNewDrift,"local",
          muStart=rep(0,9),0.2,1,0.1,0.05,L=1)

inhomoFound timeOfInhomo heightInhomo locInhomo
       TRUE           34     1.176874         5
```

Similar as in the SNHT case, the time is not estimated accurately which is due to the construction of the `test.H1Loc()`, but the location is detected correctly.

Negatively correlated series are usually not detected by the MGMRF by defintion.

5.3.3 Summary of inhomogeneity detection performance

Based on the previous sections, the performance of the MGMRF or the `gmrfHomogeneity TestComp()` R function can be briefly summarized in Table 5.1:

Inhomogeneity type	Performance of MGMRF model `gmrfHomogeneityTestComp()`
local shifts in mean of single time series	++
local drifts	+ (good at finding the location, bad at detecting the time)
global shifts in mean	++
negatively correlated neighbor series	−−

Table 5.1: Summary of the inhomogeneity detection performance of `gmrfHomogeneityTestComp()`. ++: Very good performance, +: reasonable performance, −: poor performance, −−: extremely poor performance.

Remark 5.3.1. *Table 5.1 only takes into account the inhomogeneity detection performance based on simulation runs. Runtime, stability etc. are not considered up to this point.*

5.4 MGMRF inhomogeneity testing in R

Hypothesis testing with the MGMRF model has been implemented in the `gmrfHomogeneityTestComp()` function (details in appendix). Using perfect samples from a MGMRF on `gmrfHomogeneity TestComp()` does obviously not yield issues, mainly if the starting values for the `optim()` minimization function can be chosen perfectly. If the `gmrfHomogeneityTestComp()` function is applied to real climate data, conditions for the `optim()` convergence are often not perfect and `optim()` usually does not find the MLE of the parameters within a certain number of iterations. The following section serves to resolve uncertainties regarding the usage of the `gmrfHomogeneityTestComp()` function by making suggestions on how to avoid obtaining non-convergence. Furthermore, details are given regarding the runtime of the `gmrfHomogeneityTestComp()` function, which might be interesting for users of the framework developed.

5.4.1 Convergence of `optim()`

As already mentioned on page 45, the `optim()` R function is used to find the ML-estimates for the parameters $\vec{\mu}, b, c, f$ and a by finding the minimum of twice the negative log-likelihood function under the null and alternative hypotheses (see Section 5.2.1), i.e., under the null hypothesis, the minimum over $n+3$ parameters needs to be found and under the alternative hypothesis even $n+4$ ML-estimates need to be approximated for each time $t \in \{1, \ldots, T\}$ (and location, if local inhomogeneity detection is applied), where n is the number of locations and T is the total number of months, years or seasons depending on the temporal resolution of one's CMIP5-ng file.
`optim()` itself, or more precisely the method "L-BFGS-B", which has been used in all of the `optim()` calls in this thesis, is a limited-memory quasi-Newton algorithm with specified lower and upper bounds for each parameter as constraints for optimization [Held and Bové, 2013]. The Newton – Raphson algorithm for optimization is a numerical algorithm for a function $g : \mathbb{R}^n \to \mathbb{R}$, where the update of each iteration is defined in the following way:

$$\vec{\theta}^{(t+1)} = \vec{\theta}^{(t)} - (H_g(\vec{\theta}^{(t)}))^{-1} \cdot \nabla_g(\vec{\theta}^{(t)}) \tag{5.4}$$

Whereas the Newton – Raphson method for optimization uses the exact Hessian $H_g(\vec{\theta}^{(t)})$ and gradient $\nabla_g(\vec{\theta}^{(t)})$ at every iteration step t, "quasi-Newton" methods use positive definite approximations of the Hessian based on the successive approximations of the gradients. Thereby, the gradient can be approximated by:

$$\begin{aligned} \frac{\partial g(\vec{\theta})}{\partial \vec{\theta}_i} &\approx \frac{g(\vec{\theta} + \epsilon e_i) - g(\vec{\theta} - \epsilon e_i)}{2\epsilon}, \text{ where} \\ e_i &= (0, \ldots, \underbrace{1}_{i\text{th entry}}, \ldots, 0), \\ \epsilon \in \mathbb{R}_{>0} &\text{ chosen small.} \end{aligned}$$

Convergence of `optim()` with the "L-BFGS-B" method is therefore reached if the gradient is 0 and the corresponding Hessian matrix is positive definite, which are sufficient criteria

for a local minimum.
The "L-BFGS-B" method, however, might not necessarily converge to the desired minimum, if starting values are far away from the actual minimum or if there are discontinuities in the likelihood function. 10^6 has been assigned to twice the negative log-likelihood in the non-valid parameter space. On the valid parameter space, the negative log-likelihood is significantly smaller. Therefore, `optim()` convergence issues appear at the boundary of the valid and non-valid parameter space.
In this thesis, `optim()` has been used with the maximal number of iterations set to 200. This can be done with the command

```
control=list(maxit=200).
```

If `optim()` does not converge within 200 iterations an exception is thrown by the `gmrfHomogeneityTestComp()` saying: "quasi – Newton method did not converge under H0/H1". If this is the case, a user might want to know how to proceed. Hence, operating instructions are given in these few steps:

1. Apply the `gmrfHomogeneityTestComp()` to the data.
2. The quasi-Newton method does not converge under H_0:
   (a) If `optim$convergence=52`: then one can extract the latest valid parameters b, c and f and pass them to `gmrfHomogeneityTestComp_VR()`. The b, f parameters should be at the bounds of the valid and non-valid parameter space (see Figure 5.2).
   (b) If `optim$convergence=1`: The iterations limit `maxit` has been reached. One can try another starting value or proceed as in the `optim$convergence=52` case.
3. `gmrfHomogeneityTestCom_VR()` fixes the parameter b under H_0 and H_1 to the input value and fixes f and c under H_1 to the converged value under H_0. This procedure, with almost no exception, leads to convergence of the `optim()` but increases runtime tremendously.

It is obvious that fixing a parameter has an effect on the likelihood under H_0 and H_1 since the parameters affect the determinant of the precision matrix, which is an important component of the likelihood function. Fixing parameters, however, also has an influence on the sensitivity of the significance test. Two aspects of the fixation influence the behavior and are briefly elucidated.
First, one should be aware of the fact that fixing parameters, which have been obtained by the `optim()` under H_0, leads to suppressing the following effect: Under H_1, the parameter estimates for b, c and f are usually smaller if there is an inhomogeneity in the data compared to the estimates under H_0. This is due to the fact that a shift in a time series leads to higher temporal and spatial conditional correlation. If the parameters are then fixed to the same value under H_0 and H_1, this effect is then suppressed and could potentially lead to a biased likelihood ratio statistics.
Second, if a parameter is fixed to a value that is further away from the truth, the model bias tends to be reinforced.
In order to examine these two aspects, the same global shift experiment (with the same underlying data) has been repeated as in Section 5.2.2. This time, the `gmrfHomogeneity-`

`TestComp_VR()` has been used and b has been fixed to 0.2 under H_0 and H_1 and then fixed to 10^{-5} under H_0 and H_1. The true value of b is 0.2 under H_1.

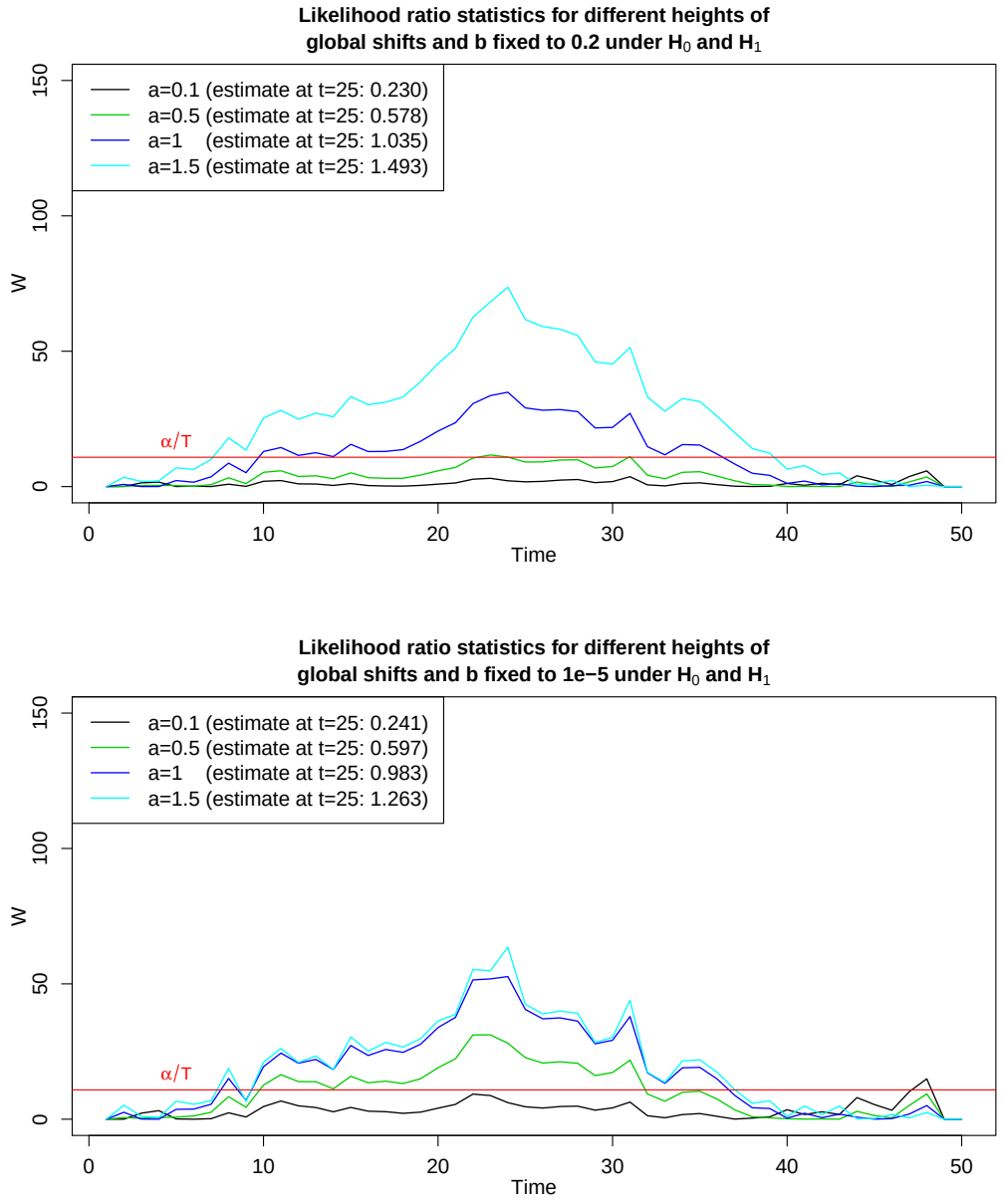

Figure 5.10: Likelihood ratio statistics in global shift detection with varying shift heights introduced at $t_0 = 25$. Top: b has been fixed to 0.2 (true value under H_1). Bottom: b has been fixed to 10^{-5}.

Comparing Figure 5.10 to Figure 5.8 of Section 5.2.2 reveals that essentially nothing changes if b is fixed to 0.2 and f, c are fixed to MLEs as obtained by the null hypothesis. However, if b is fixed to a value that is approximately 0.2 away from the true value, changes are apparent. It seems as if the estimates for the shift height get more inaccurate the larger the introduced shift in the mean level is. Simultaneously, the test statistic increases and the test in general is more sensitive with an increased probability of committing a type I error (e.g., shortly before $t = 50$, a type I error has been committed for b set to

10^{-5}). Overall, the impact of fixing b is, however, less extreme as probably suspected. Through extracting the b to a realistic value as obtained under H_0, it is more probable to be in a situation as depicted in the top plot than the bottom plot of Figure 5.10.

5.4.2 Empirical runtime estimation of the `gmrfHomogeneityTestComp()` function

A user of the `gmrfHomogeneityTestComp()` function might be interested in a runtime estimation. One may remember that the `gmrfHomogeneityTestComp()` function includes the optimization of the likelihood function under the null and alternative hypotheses using a quasi-Newton method. Runtime, therefore, partly depends on the speed of convergence of this numerical method. The speed of convergence, on the other hand, depends on further variables such as the chosen starting values, the upper and lower bounds or scaling factors set in the option "parscale" in the `optim()` call. The maximal number of iterations in `optim()` can, however, be controlled by the option `maxit`. All the code produced for this thesis works with 200 as an upper bound for the number of iterations. In a single iteration the precision and the likelihood needs to be estimated based on the `optim()` updated values of the parameters. Furthermore, `optim()` needs to estimate the gradient, the Hessian matrix and calculates a new vector of parameters $\vec{\theta}^{(i+1)}$ from the previously parameter vector $\vec{\theta}^{(i)}$.
Apart from optimizing the likelihood, the size of the data itself has a large effect on runtime. The data is passed to the `gmrfHomogeneityTestComp()` function as a three dimensional array with dimensions longitude, latitude and time. The time component has an approximately linear effect on the runtime if $L = 1$ as input parameter in `gmrfHomogeneityTestComp()`. Thereby, L is an integer parameter and determines in what temporal distances the likelihood ratio statistics are evaluated. E.g., if $L = 1$, then the likelihood ratio statistics are evaluated at every time unit, if $L = 5$, then it is evaluated only every 5th time etc. Thus, choosing L large accelerates runtime by a factor of L but simultaneously information on the likelihood ratio statistic is lost and the time at which a temporal shift of the mean level occurred, might not be detected accurately. Apart from the temporal dimension, runtime of the `gmrfHomogeneityTestComp()` function obviously increases by an increasing number of spatial locations. More parameters need to be estimated with a larger spatial field. The runtime also depends on the "type" as an input parameter of the `gmrfHomogeneityTestComp()` function. If it is chosen as "local", a loop over all locations is performed which would then lead to a more than linear increase of runtime with an increasing number of locations. The question is, if it is also more than linear, if the "type" is chosen as global. For this reason, the "space vs. time" comparison has been conducted.

Space vs. time
Is it more expensive to apply a narrow spatial field with many time units or a large spatial field with a few time units to the `gmrfHomogeneityTestComp()` function? An experiment has indicated that the latter is the case, i.e., it usually takes longer to estimate the parameters for many spatial pixels compared to only estimate them for every time on a small spatial domain. The experiment has been conducted on a $3 \times 3 \times 100$ and a $10 \times 10 \times 9$ data array over Europe, i.e., in both cases the same amount of data values have been analyzed. The code of the experiment can be found in the appendix. A 5 times faster runtime has been measured for the data set with large temporal dimension on a

narrow spatial domain compared to the large spatial domain with only a few time units. Overall, runtime of the `gmrfHomogeneityTestComp()` function is dependent on the speed of the likelihood optimization, but also on large parts on the input parameters that are chosen as well as the size of the data.

5.5 MGMRF methods on the CMIP5-ng data

The previous sections have illustrated that the `gmrfHomogeneityTestComp()` R function can be used as a tool to detect global and local inhomogeneities in spatio-temporal data sets. This section finally presents applications of `gmrfHomogeneityTestComp()` on CMIP5-ng data. Preliminary, the CMIP5-ng data needs to be "transformed" as close as possible to realizations of a MGMRF, in order to provide good conditions for convergence of the `optim()` and inhomogeneity detection performance in general. Raw CMIP5-ng data contains seasonality and trends (e.g., induced by RCP-scenarios) which can affect the performance of inhomogeneity detection and optimization. Therefore, these climatic occurrences are recommended to be removed. In Section 5.5.1, methods are presented that use the residuals of different representative models, which can then be passed to `gmrfHomogeneityTestComp()` in order to provide more stability in the code compilation process.

5.5.1 Removing seasonality and trends

The challenge in removing seasonality and trends from the residuals is to find a model that does not eliminate the inhomogeneities that need to be detected. Two possible approaches are discussed in this section. The first one fits a Generalized Additive Mixed Models (GAMM) for every time series individually, whereas the second one makes use of the weighted mean over different CMIP5-ng model projection. The latter method has more bias potential, but is computationally faster. Accordingly, Section 5.5.1 provides details on how the user can avoid including strong outlier models in the weighted mean.

Decomposition with GAMM

This section focuses on the decomposition of the data into trends and seasonality with a GAMM. Thereby, the model should include predictive functions that are rather smooth with few degrees of freedom in order to not remove the inhomogeneities from the residuals that one wants to find. In a GAMM, a penalized term controls the smoothness by a penalty on the integrated and squared second derivative of the smoother. The GAMM models, which are suggested to be used, look as follows:

$$\begin{aligned} \vec{y} &= \beta_0 + s_{season}(\vec{x}_1) + s_{trend}(\vec{x}_2) + \vec{\epsilon}, \text{ for monthly data,} \\ \vec{y} &= \beta_0 + s_{trend}(\vec{x}_2) + \vec{\epsilon}, \text{ for annual data.} \end{aligned}$$

Smoothers: s_{season}, s_{trend}
If the data to be analyzed has a temporal resolution of a month, it is suggested to use a cyclic cubic spline for the seasonal smoother s_{season} on the vector of months, i.e., $x_1 = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, \dots)$. Here, the cyclic cubic spline consists of a basis of cubic splines all fitted for the same period of 12 subsequent months (January, ..., December). A weighted sum over these basis splines then form the cyclic cubic spline.

Knot points are defined at which the spline basis functions should meet to avoid discontinuity. In the `gamm()` R function of the R package `gamair` [Wood, 2015], the cyclic cubic spline type is specified as `bs=''cc''`.
For s_{trend} the `bs=''cr''` R type is suggested to be used, which refers to a basis of cubic splines with a moderate size of equidistant knots across the covariates x_2. Here, x_2 is chosen as a sequence of $1, \ldots, T$, where T is the total number of months in the investigated CMIP5-ng data set. The `gamPeriodTrendRem()` R function (see appendix) has been implemented in the course of this thesis to assist in the removal process with the above described smoothers in the GAMM.

Example 5.5.1. *The usage and results of the GAMM model is illustrated below on a CMIP5-ng Near Surface Temperature time series of the climate model ACCESS1-0 (r1i1p1) at a* $2.5° \times 2.5°$ *pixel over Switzerland. The* ***gamPeriodTrendRem()*** *function prints out a summary of the model fit with GAMM.*

```
Family: gaussian
Link function: identity

Formula:
values ~ s(month, bs = "cc", k = 12) + s(time, bs = "cr")

Parametric coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 281.53155    0.03255    8649   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Approximate significance of smooth terms:
           edf Ref.df      F p-value
s(month) 9.760 10.000 5180.6  <2e-16 ***
s(time)  7.092  7.092  276.1  <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

R-sq.(adj) =  0.951
Scale est. = 2.9359    n = 2772
```

The R summary shows highly significant p-values as one might have expected.
At this point, one may be interested in predictions of the GAMM model, which are illustrated in Figure 5.11 for a first slice of 100 months. The smooth trend part of the model is depicted as well.

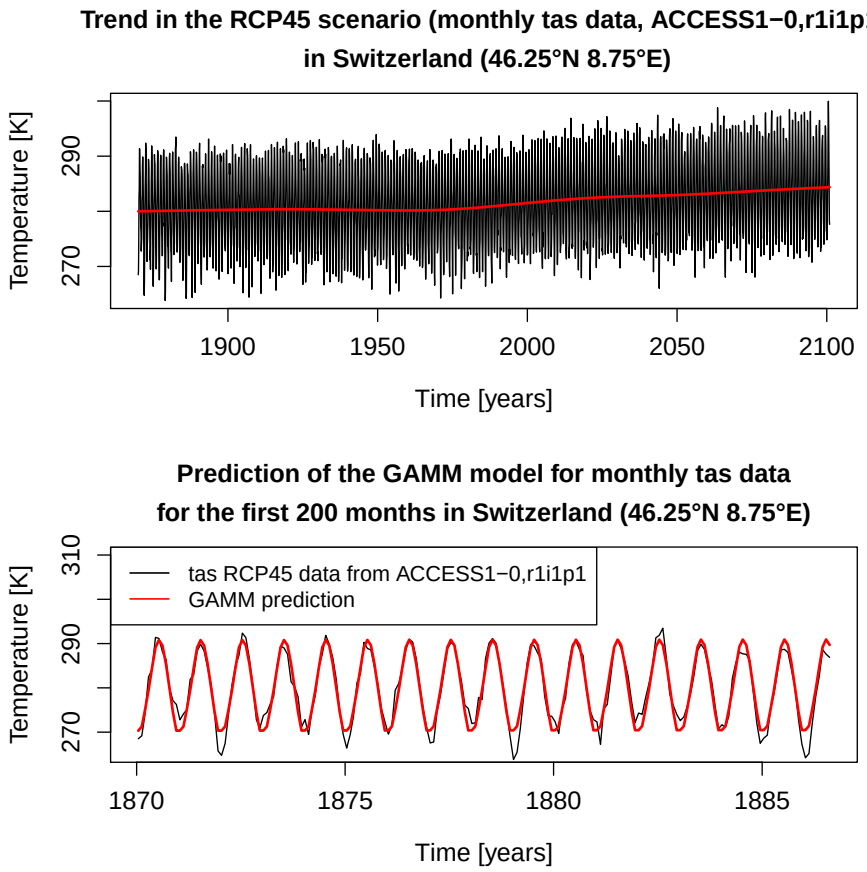

Figure 5.11: Top: Trend part of the GAMM in the context of the ACCESS1-0 data series. Bottom: Prediction of the GAMM model based on the ACCESS1-0 data series for the first 200 months since 1870.

Figure 5.11 shows a smooth trend with little variance as desired. The remaining residuals of the GAMM at the spatial pixel centered at $46.25°N\ 8.75°E$ *have Autocorrelation Function (ACF), Partial Autocorrelation Function (PACF) and Q-Q plots depicted in Figure 5.12:*

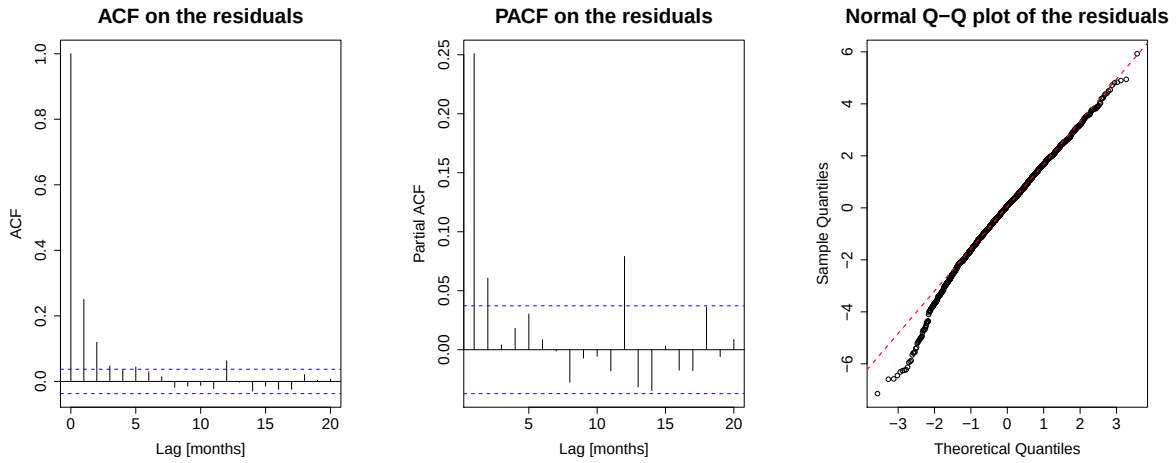

Figure 5.12: ACF, PACF and Q-Q plots of the residuals of the GAMM model applied to the ACCESS1-0 data series. The red dashed line depicts the theoretical normal quantiles.

The ACF and PACF plots show significant autocorrelation for lags of 1,2 and 3 which may be fixed by modeling and AR(2) process, nevertheless, the MGMRF is applied to the residuals in which way temporal correlations are wanted. For a lag of 12, the ACF exceeds the 95% threshold, which is most likely to seasonality that could not be removed through the process.

Decomposition with weighted model means

A more efficient way of removing seasonality and trends in one specific CMIP5-ng file would be to subtract it from the overall (weighted-) mean over all models and ensembles of a specific scenario, variable and resolution in the CMIP5-ng data pool. The underlying assumption is that the mean of different model projections of the same climate variable, scenario and resolution provide a reasonable overall representation of a specific variable under a specific scenario and resolution. If this assumption is not fulfilled then one may proceed as in Section 5.5.1.

Example 5.5.2. *One may again look at the Near Surface Temperature data under the RCP45 scenario and the ACCESS1-0 (r1i1p1) model for a monthly temporal resolution and subtract it from the overall weighted mean of all monthly Near Surface Temperature data files under the RCP45 scenarios. Figure 5.13 illustrates what the difference series and the mean series as well as the original data look like at a spatial pixel located in Switzerland for the first 100 months.*

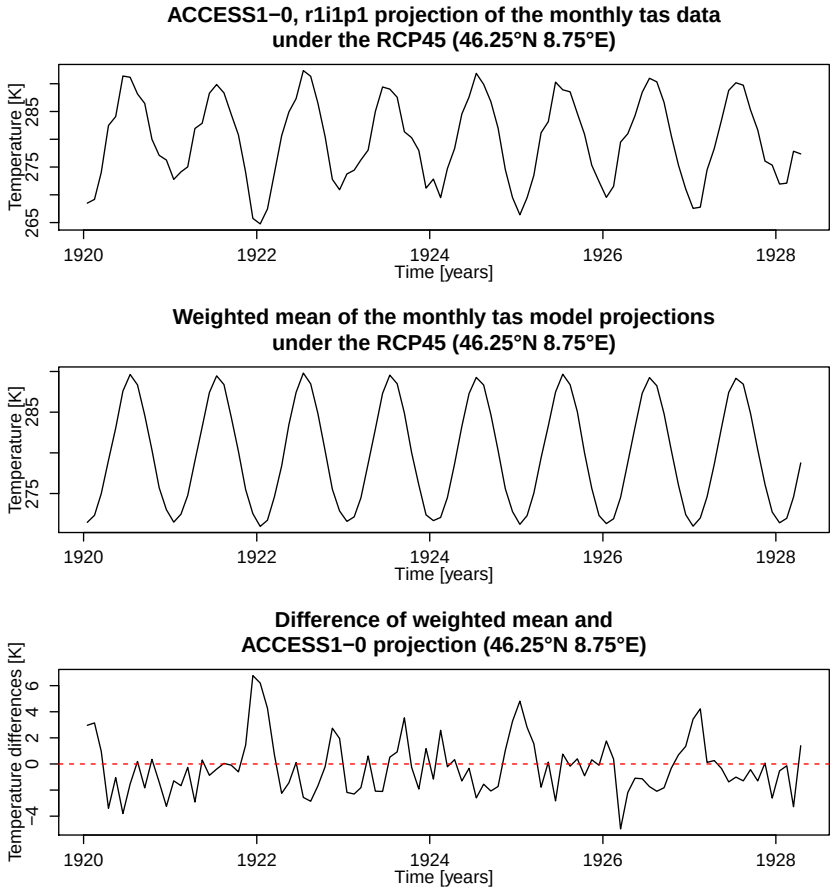

Figure 5.13: Time series at a pixel in Switzerland for the first 100 months. Top: The original ACCESS1-0, r1i1p1 data. Center: Mean series of all tas-RCP45-monthly-2.5° × 2.5° CMIP5-ng data sets. Bottom: The difference time series of ACCESS1-0 (r1i1p1) and the weighted mean of 110 model projections of the same scenario, variable and resolution.

One might also be interested in whether or not the above difference series really are normally distributed. The Q-Q plots from three sample time series in Figure 5.14 show that it very much depends on the geographical region how close the residuals resemble samples from a normal distribution.

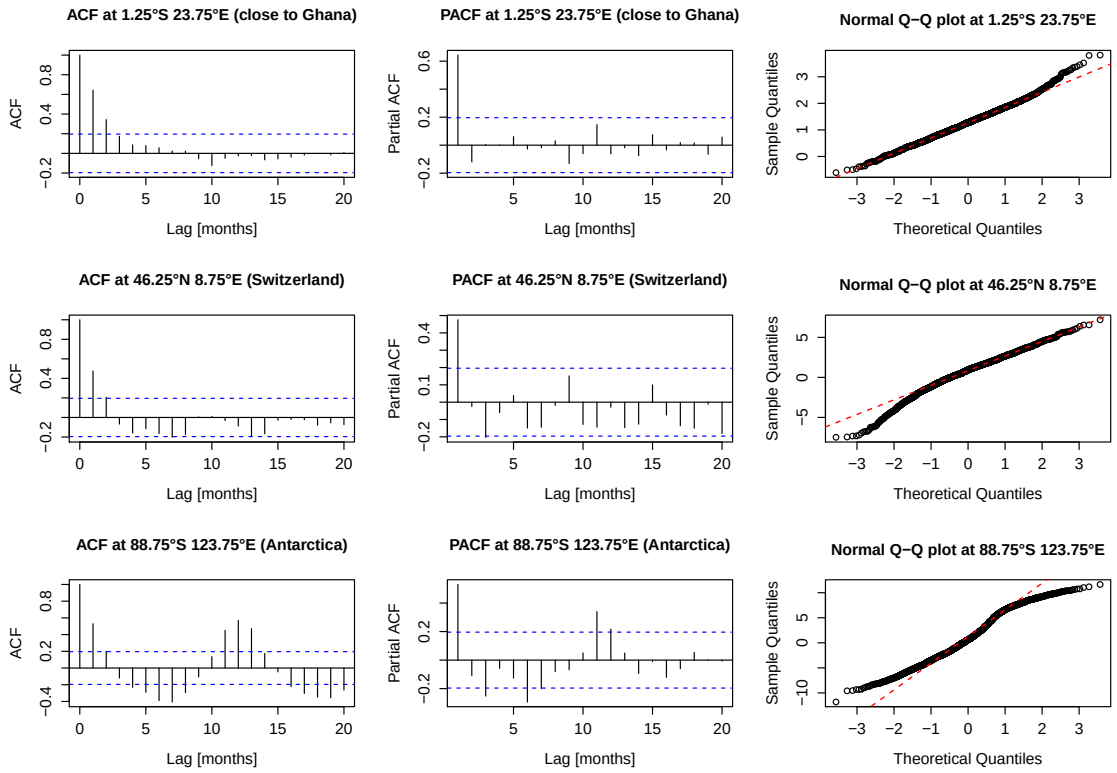

Figure 5.14: ACF, PACF and Q-Q plots for different spatial pixels (2.5° × 2.5°) based on the "deseasonalized" (trend, seasonality removed) ACCESS1-0 (r1i1p1) monthly tas data. The red dashed line in the Q-Q plots depict the theoretical normal quantiles.

The ACF and Q-Q plots for the spatial pixel over Switzerland look very similar to the one in Figure 5.12. It similarly shows a positive autocorrelation for a lag of 12 and lags 1, 2 and 3, i.e., despite a different approach to removing the seasonality and trend was used, similar effects and structure are still existent in the remaining residuals.
The time series that is located around the Equator shows better results in terms of the ACF and the Q-Q plot which might be due to the fact that Equatorial regions experience less marked seasonality as regions in the Antarctica or in Switzerland.

The procedure of building the weighted mean is automated by the `meanOfFiles()` function, which has been implemented in the course of this thesis. This method can be used as follows (illustrated for a NetCDF file containing monthly precipitation data on the RCP45 scenario):

```
source('/../meanBuilder.R')
source('/.../MeanOfFiles.R')
source('/.../difference.R')
#meanOfFiles only needs the path to the directory
#with all pr_mon_.*_rcp45_.*_g025.nc files
prMean <- meanOfFiles("/.../pr",weighted=T)

#difference subtracts the mean (here ''prMean'') from
#a specific data set and standardizes each time series.
desData <- difference(Mean = prMean,
                      path = "/.../pr/pr_mon_CCSM4_rcp45_r6i1p1_g025.nc")
```

Given a path of a directory with NetCDF files and a specific file from this directory, it calculates the **weighted mean** over all files in that directory. The weights are defined according to the number of ensembles a certain model produces. The assumption is that ensembles are more likely to be dependent whereas models are thought to be independent even though, due to shared code among the climate institutes, independence can still not always be guaranteed.

Bias in the weighted mean

One may criticize that the removal of trends and seasonality via the weighted mean is not optimal since biased models, if existent, are included in determining the weighted mean. The measurement and assessment of the overall bias, on the other hand, is non-trivial due to uncertainties of the model projections in general.
Nonetheless, one may for instance order the models according to the standardized Sum of Squared Differences (SSD) of the overall weighted mean to the mean of the model projections (mean over all ensemble projections of a certain model) in order to detect models that are "far away" from the overall mean. The SSD for a model M is defined as:

$$
\begin{aligned}
SSD^{(M)} &:= \sum_{i \in Lon \times Lat \times Time} \frac{(x_i^{(M)} - \bar{x}^{(i)})^2}{s^2}, \text{ where} \\
Lon &= \{1, \ldots, 144\}, \text{ this range only applies for a } 2.5 \times 2.5^\circ \text{ spatial resolution} \\
Lat &= \{1, \ldots, 72\}, \text{ this range only applies for a } 2.5 \times 2.5^\circ \text{ spatial resolution} \\
Time &= \{1, \ldots, T\} \\
x_i^{(M)} &: \text{data values of model } M \text{ at location and time } i \\
\bar{x}^{(i)} &: \text{value of the weighted overall mean at location and time } i \\
s &: \text{sample standard deviation across all values of } \bar{x}
\end{aligned}
$$

In R, the following code can be used to calculate the SSD:

```
weigthedMeanTas <- meanOfFiles("/.../tas",weighted = T) #code: see appendix
meansOfModels <- getMeansOfModel("/.../tas") #code: see appendix
des <- list()
for(i in 1:length(meansOfModels)){
  des[[i]] <- (weigthedMeanTas-meansOfModels[[i]])
}
names(des) <- names(meansOfModels)
ssd <- numeric(length(des))
for(i in 1:length(des)){
  ssd[i] <- sum((des[[i]]/sd(weigthedMeanTas))^2)
}
```

The SSD for the Near Surface Temperature can also be represented graphically as done in 5.15.

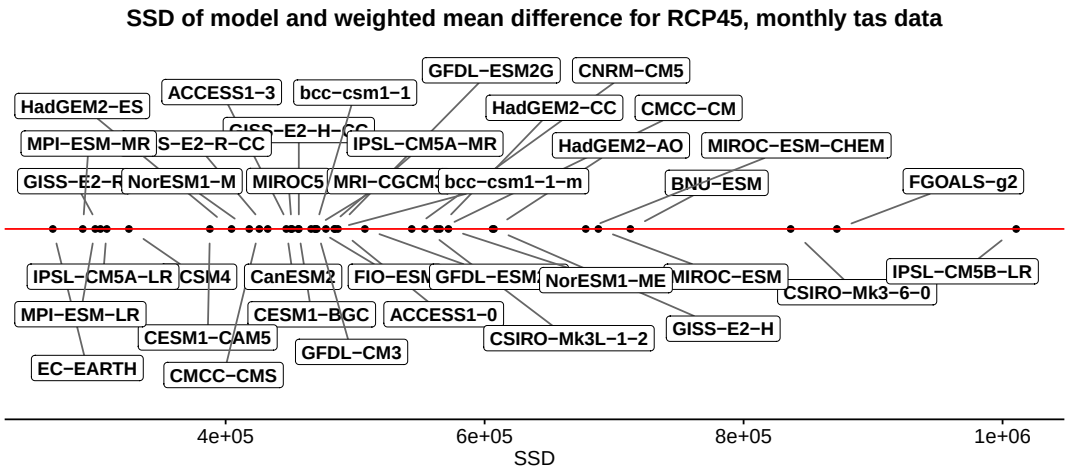

Figure 5.15: The SSD the weighted over all mean to the model mean, illustrated for monthly RCP45 Near Surface Temperature data.

Figure 5.15 shows that the IPSL-CM5B-LR climate model is furthest away from the overall weighted mean. That does not mean that the climate model is a "bad" climate projection since the truth is unknown. At this point, it is left to the user to decide if or not to remove the IPSL-CM5B-LR model from the calculation of the weighted mean.
For monthly precipitation data under the RCP45 the SSD looks slightly different as depicted in Figure 5.16.

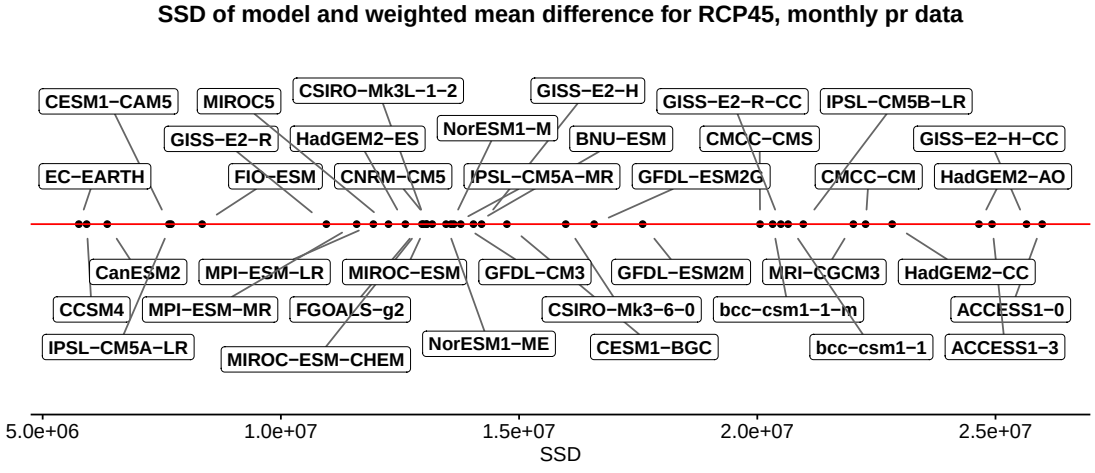

Figure 5.16: The SSD the weighted over all mean to the model mean, illustrated for monthly RCP45 precipitation data.

This procedure not only reveals the cause of a possible bias in the overall mean by looking

at outliers, but it also shows that models from the same research institution produce similar results, such as ACCESS1-0, ACCESS1-3, HadGEM2-AO or HadGEM2-CC.
In this above case, one may notice that the ACCESS models are furthest away from the overall weighted mean. That does again not mean, that the ACCESS models have produced "bad" projections of the variable and scenario since the truth is unknown. Last but not least, the same plot has been produced for the Upwelling Longwave Radiation (see Figure 5.17).

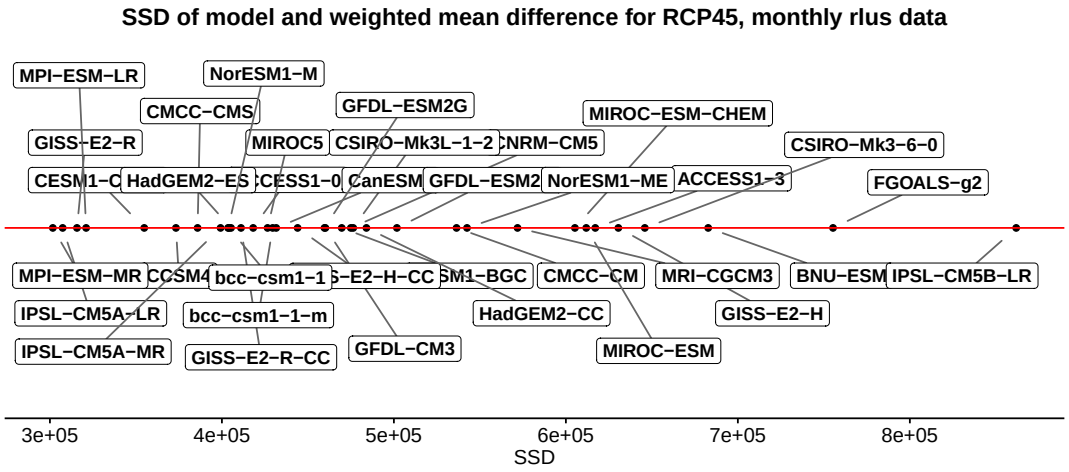

Figure 5.17: The SSD the weighted over all mean to the model mean, illustrated for monthly RCP45 Surface Upwelling Longwave Radiation data.

Figure 5.17 reveals that the same climate models, as in the Near Surface Temperature case, seem to be outlier models in the Surface Upwelling Longwave Radiation, which might be due to the fact that Near Surface Temperature and Longwave Radiation are strongly related climate variables.

5.5.2 Applying `gmrfHomogeneityTestComp()` to CMIP5-ng data sets

After introducing all the MGMRF tools that have been developed in this thesis, this section finally presents the output of the `gmrfHomogeneityTestComp()` R function on CMIP5-ng data. The same spatial regions, models and variables are chosen as in the SNHT Section 4.5 in order to allow comparison of the results.

Monthly Near Surface Temperature over Europe (RCP45): ACCESS1-0 (r1i1p1)

Again, one may look at the 60 pixels over Europe as shown in Figure 4.11.

Global inhomogeneities; decomposition with GAMM
The GAMM has been used and fit to every single time series of the ACCESS1-0 monthly

Near Surface Temperature data projection. In R, this fitting procedure can be done as follows:

```
library(ncdf)
library(ncdf.tools)
library(mgcv)
library(gamair)
source('/.../gamPeriodTrendRem.R')
file1 <- '/.../tas_mon_ACCESS1-0_rcp45_r1i1p1_g025.nc'
nc1 <- open.ncdf(file1)
data1 <- get.var.ncdf(nc1)
times <- convertDateNcdf2R(time.source=nc1$dim$time$vals,
       units="days",origin=as.POSIXct("1850-01-01",tz="UTC"),
       time.format='%Y-%m-%d')
close.ncdf(nc1)
times[1] #"1870-01-16 12:00:00 UTC"
times[length(times)] #"2100-12-15 12:00:00 UTC"
year <- rep(seq(1870,2100),each=12)
month <- rep(c(1:12),length(times)/12)
res <- matrix(0,nrow=144*72,ncol=length(year))
k <- 1
for(j in 1:72){
  for(i in 1:144){
    data <- data.frame(month=month,year=as.factor(year),
          values=data1[i,j,],time=c(1:length(year)))
    res[k,] <- gamPeriodTrendRem(data)
    k <- k+1
  }
}
```

Having the residuals, the likelihood ratio statistics can then be obtained as follows:

```
#Europe 10x6x2772
library(spam)
source('/.../gmrfHomogeneityTest_VR.R')

dim(res) <- c(144,72,2772) # res from above
desDataEurope <- res[c(1:10),c(55:60),]
av <- matrix(0,10,6)

for(i in 1:10){
  for(j in 1:6){
    av[i,j] <- mean(desDataEurope[i,j,])
  }
}

timeTaken <- system.time(
           outGlobTasVR <- gmrfHomogeneityTestComp_VR(desDataEurope,
                           type="global",mu=c(av),
                           bStart=0.14,cStart=1.2,fStart=0.2,
                           sigLevel=0.05,L=5))
```

```
#   inhomoFound timeOfInhomo heightInhomo
# 1        TRUE         2710    0.6554669

# > timeTaken
#      user    system   elapsed
# 32766.594    57.361 32835.670 --> approximately 9 h
```

`outGlobTasVR` carries the likelihood ratio statistics as well as the estimates for the global shift heights `a`. These are depicted in Figure 5.18.

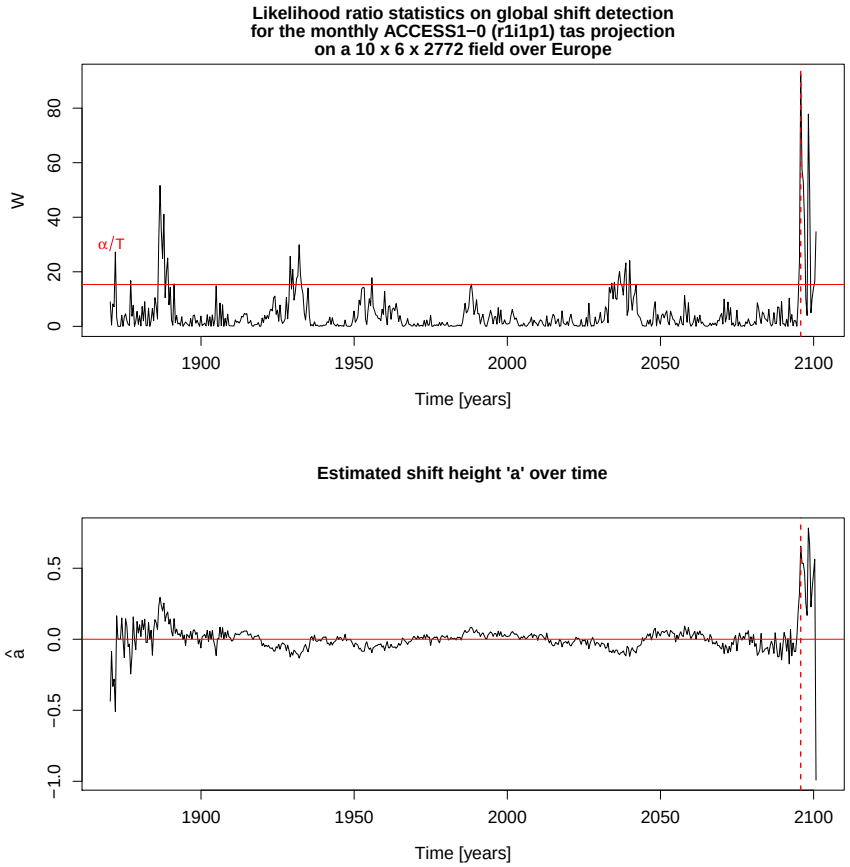

Figure 5.18: Likelihood ratio statistics and estimated shift heights $\hat{a}$ over time for the monthly Near Surface temperature ACCESS1-0 (r1i1p1) projection on a $10 \times 6 \times 2772$ field over Europe.

In general, one should pay more attention to significant likelihood ratio statistics in the middle of the time interval than at the beginning and end of the time interval. The reason for that can be found in the construction of the `test.H1Glob()` R function. If there are a few extreme values at the end of the interval, the likelihood ratio statistics will react sensitively to these values. This behavior is somewhat similar to the original Alexandersson's SNHT statistics.
More importantly, if one compares the results of the `gmrfHomogeneityTestComp()` in Figure 5.18 with the ones of the `pairwiseSNHT()` in Figure 4.12, it becomes evident that similar time regions have been declared as inhomogeneous, namely those between the years 1900 and 1950 as well as between 2000 and 2050. Thus, the two inhomogeneity

tests share some level of consistency even though they use completely different approaches to find inhomogeneities.

Global inhomogeneities; decomposition with weighted mean model
Proceeding analogously as above, the difference of the ACCESS1-0 and the weighted mean model can be used as an input for the `gmrfHomogeneityTestComp()` function.

```
library(ncdf)
source('/.../gmrfHomogeneityTest_VR.R')
source('/.../MeanOfFiles.R')
source('/../difference.R')
Mean <- meanOfFiles("/.../tas",weighted = T)
desData <- difference(Mean,'/.../tas_mon_ACCESS1-0_rcp45_r1i1p1_g025.nc',
                  standardize = TRUE)
desDataEurope <- desData[c(1:10),c(55:60),]
av <- matrix(0,10,6)

for(i in 1:10){
  for(j in 1:6){
    av[i,j] <- mean(desDataEurope[i,j,])
  }
}

timeTaken <- system.time(
           outGlobWght <- gmrfHomogeneityTestComp_VR(desDataEurope,
                        type="global",mu=c(av),
                        bStart=0.13,cStart=2,fStart=0.2,
                        sigLevel=0.05,L=5))
# > timeTaken
#      user    system   elapsed
# 47763.961    45.816 47837.277  --> approximately  13 h 17 min

#   inhomoFound timeOfInhomo heightInhomo
#          TRUE         1415   -0.1631918
```

`outGlobWght$a` and `outGlobWght$lRatioStat` can again be plotted as above, yielding likelihood ratio statistics and shift heights as depicted in Figure 5.19.

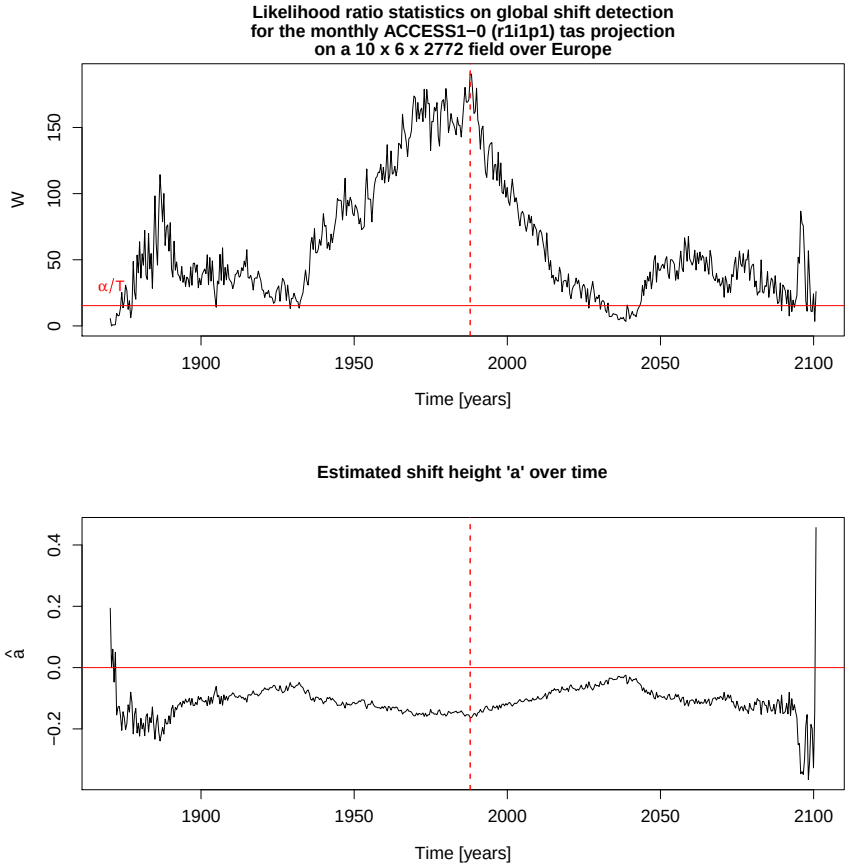

Figure 5.19: Likelihood ratio statistic and estimated shift height a produced by `gmrfHomogeneityTestComp()` on the differences of the ACCESS1-0 to the overall weighted mean of monthly Near Surface Temperature data under the RCP45 scenario.

The likelihood ratio statistic is significant almost at any point in time. The estimates for a, on the other hand, are almost all negative. Yet, the two plots do not agree on the absolute values of the maximum. The likelihood ratio statistic is maximal at $t_0 = 1415$ (months since January, 1870) whereas the estimates for a are maximal in absolute terms at $t_0 = 2770$ (months since January, 1870). Interestingly, the estimates for a look very similar in a neighborhood of $t_0 = 2770$ (months since January, 1870) as in Figure 5.18. Overall, the statistic and its associated estimates for a, however, are not reasonable. At the very end of the time interval, the statistic shows no significance, but the corresponding estimates of the shift heights are large in absolute terms. A possible reason for this behavior might be that the time series are still not normalized enough by the subtracting the weighted mean. Hence, it is suggested to use the weighted mean approach for different applications but not to remove trends and seasonality.

Monthly Surface Upwelling Longwave Radiation over South Africa (RCP45): CSIRO-Mk3L-1-2 (r1i2p1)

Analogously as above, the residuals via the GAMM can be obtained for the CSIRO-Mk3L-1-2 (r1i2p1) model projection. It has been elucidated in the previous sections that the `gmrfHomogeneityTestComp()` function performs with relatively slower runtime on large

spatial fields. Therefore, unlike in the applications section of the SNHT, it is only focused on a rather small spatial field over Africa with 25 locations instead of the original 220 locations. The spatial field looks as in Figure 4.16.

Global inhomogeneities with GAMM

The global shift detection has been applied analogously as in the Near Surface Temperature example, yielding:

```
#  inhomoFound  timeOfInhomo heightInhomo
#         TRUE          2765    -3.723992    # 2755 refers to the 2755-th
                                              # month after January 1870

# the CPU time measured of 'gmrfHomogeneityTestComp()' is the following:
# > timeTaken
#     user   system  elapsed
# 5325.478    8.420 5335.282 --> approximately 1.5 h
```

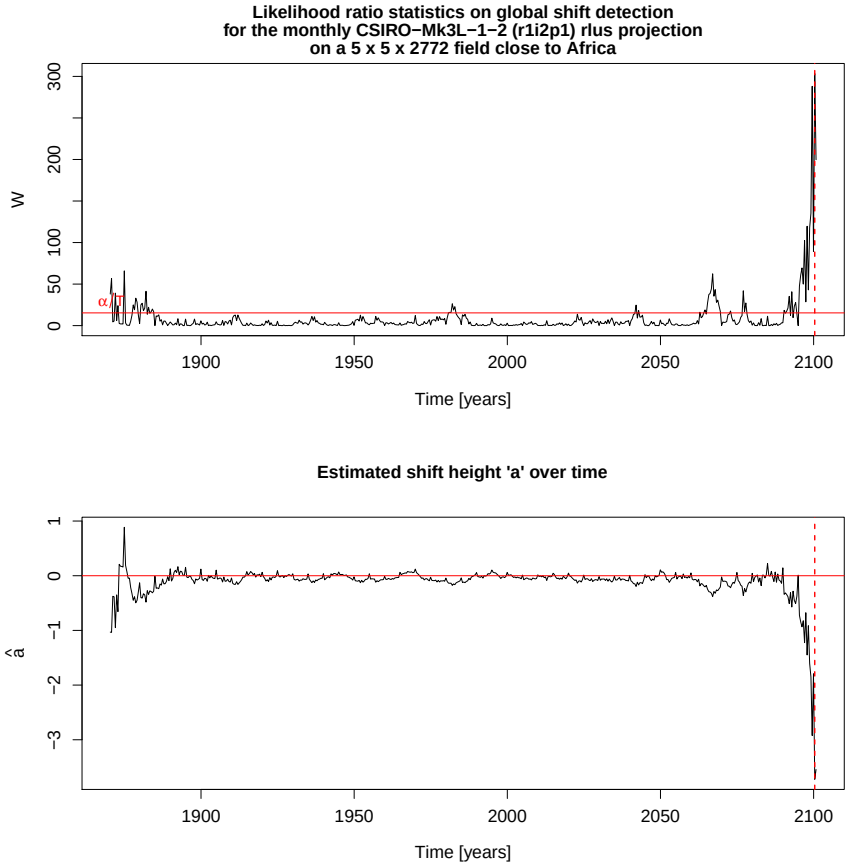

Figure 5.20: Likelihood ratio statistics and estimated shift height for `a` over time for the monthly Surface Upwelling Longwave Radiation projection on a $5 \times 5 \times 2772$ field over Africa.

Figure 5.20 as well as the `gmrfHomogeneityTestComp()` output, again, show highly significant values at the end of the time interval. These might again not be too significant as

also illustrated in the next section. Interestingly, however, is the fact that the two exceeding likelihood ratio statistics, in the middle of the time interval, lie at similar temporal regions as detected by the `pairwiseSNHT()` (see Figure 4.17).

Local Inhomogeneities with GAMM

In order to illustrate the usage of the local inhomogeneity detection, a rather small time interval from year 2050 until 2100 has been chosen, in order to keep runtime short. The code and its output are provided below:

```
#...(same data, libraries as above)
timeTaken <- system.time(
outLocAfricaSmall <- gmrfHomogeneityTestComp_VR(desDataAfrica,type="local",
                                                mu=c(av),
                                                bStart=0.1393180,
                                                cStart=1.6189352,
                                                fStart=0.2583866,
                                                sigLevel=0.05,L=5))

# >timeTaken
#     user   system  elapsed
# 7063.846   35.892 7099.973 --> approximately 1 h 58 min

#   inhomoFound timeOfInhomo heightInhomo locInhomo
#          TRUE          590    -2.564506        21
```

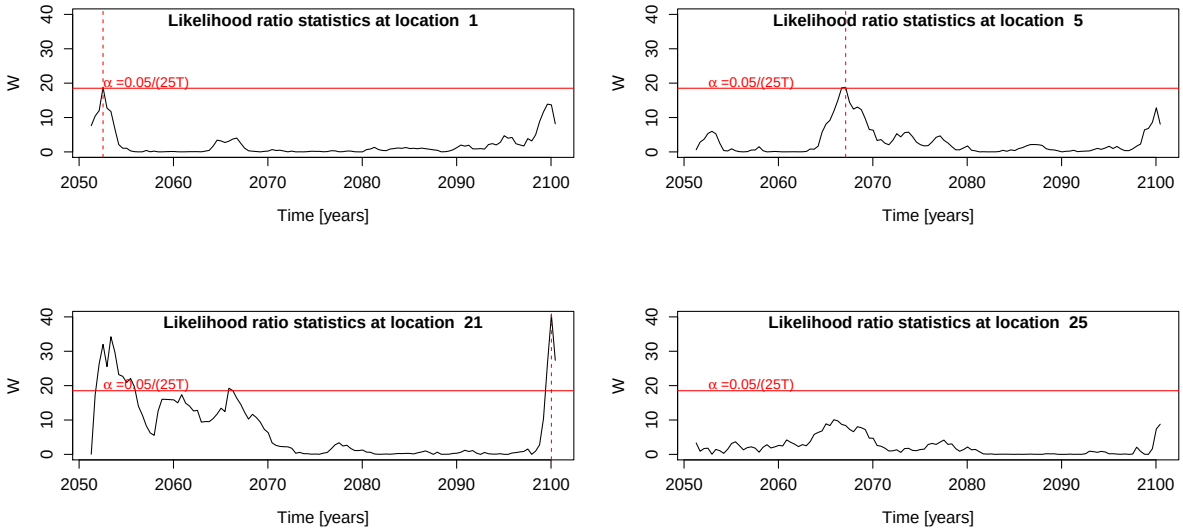

Figure 5.21: Likelihood ratio statistics at pixels at the vertices of the 5×5 spatial domain

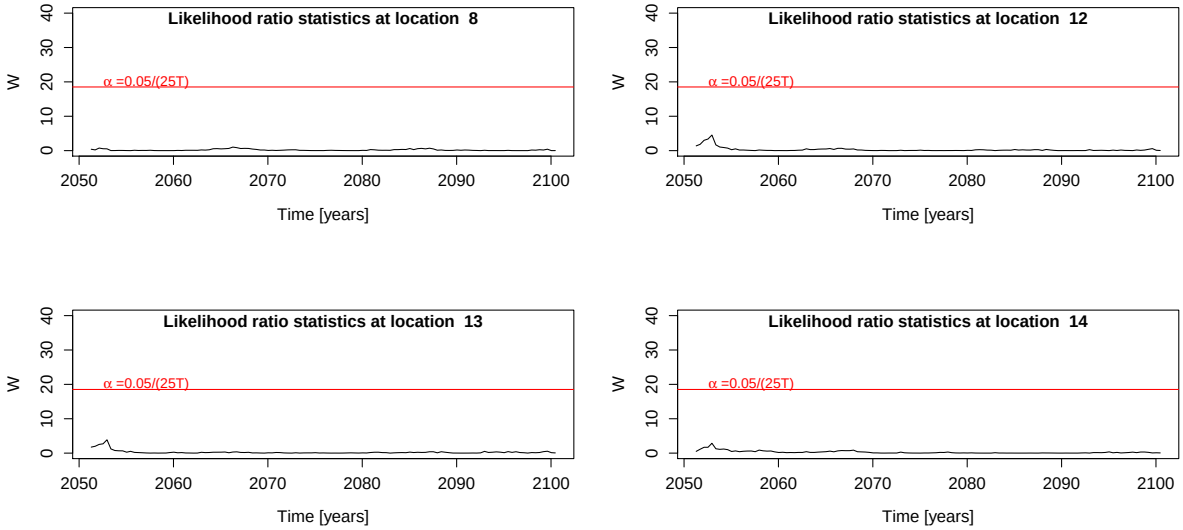

Figure 5.22: Likelihood ratio statistics at pixels in the middle of the 5×5 spatial domain.

The effect of the strongly exceeding likelihood ratio statistics is not as extreme as before, yielding exceedance of the significance threshold only at location 21.

Chapter 6

Lattice Krig

Apart from GMRF models, a multi-resolution spatial model called "Lattice Krig", which was developed by Dr. Douglas Nychka and his working group has been chosen as an analysis tool for the large CMIP5-ng data. It is applicable to large data sets since estimation and prediction of the model are computationally affordable, which is due to the GMRF methodology that is used for the stochastic coefficients of compact-support basis functions. Unlike in the previously introduced GMRF models, there is no temporal component included in the multi-resolution model Lattice Krig.

6.1 Theory

The following theory on Lattice Krig is based on Nychka et al. [2013]. It intends to illustrate the basic ideas, definitions and properties of the spatial Lattice Krig model as a preparation to the R applications Section 6.2.

6.1.1 Basic construction of the spatial model

Given are n pairs of observations $(\vec{x}_i, y_i), i \in \{1, \ldots, n\}$, then a model of the following form is considered:

$$\begin{aligned} y_i &= \vec{Z}_i^T \vec{d} + g(\vec{x}_i) + \epsilon_i, \text{ where} \\ \epsilon_i & \quad \text{are random errors and} \\ g : \mathbb{R}^p \to \mathbb{R} & \quad \text{is an unknown, smooth function,} \\ \vec{Z}_i & \quad \text{is a vector of covariates,} \\ \vec{d} & \quad \text{is a vector of linear parameters.} \end{aligned} \tag{6.1}$$

The goal of Lattice Krig is to estimate $g(\vec{x})$ based on the observations and to quantify the uncertainty of the estimates. Here, $p = 2$, i.e., one looks at longitude-latitude predictive variables $\{\vec{x}_i\}_{i \in \{1,\ldots,n\}}$, whereby the function $g : \mathbb{R}^2 \to \mathbb{R}$ is constructed in the following way:

$$
\begin{aligned}
g(\vec{x}) &= \sum_{l=1}^{L} g_l(\vec{x}), \text{ where} \\
g_l(\vec{x}) &= \sum_{j=1}^{m(l)} c_j^l \phi_{j,l}(\vec{x}) \text{ and} \qquad (6.2) \\
\{\phi_j^l\}_{j\in\{1,\ldots,m(l)\}} &\quad \text{a sequence of } \textbf{basis functions} \text{ at level } l \\
\vec{c}^{\,l} \in \mathbb{R}^{m(l)} &\quad \text{a vector of coefficients distributed as } \mathcal{N}_{m(l)}(0, \rho P_l) \\
(P_l)_{m(l)\times m(l)} &\quad \text{depends on additional parameters} \\
Z_{n\times m(l)} &\quad \text{matrix of covariates} \\
\vec{d} &\quad \text{vector of length } m(l) \text{ of linear parameters}
\end{aligned}
$$

Nychka et al. [2013] also uses the following simplified notation of the multi-resolution Lattice Krig model in descriptions of his package and other presentations that can be found online:

$$
\begin{aligned}
\vec{y} &= Z\vec{d} + \Phi\vec{c} + \vec{e}, \text{ where} \qquad (6.3) \\
\Phi_{ij} &= \phi_j(\vec{x}_i), \text{ i.e., } \Phi \text{ is an } n \times m \text{ matrix,} \\
\vec{c} &\sim \mathcal{N}_m(0, \rho P). \qquad (6.4)
\end{aligned}
$$

One should notice that in (6.4), the coefficient vectors $\{\vec{c}^{\,l}\}_{l\in\{1,\ldots,L\}}$ have been combined into a single vector $\vec{c} = (\vec{c}^{\,1}, \ldots, \vec{c}^{\,L})$. Apart from the general form, one may be interested in the explicit form of the basis functions and different levels of resolution, which is explained in more detail below.

Radial basis functions

It should be remembered that the Lattice Krig model is a spatial model with different levels of spatial resolution. At each level l, a set of basis functions $\{\phi_j^l\}_{j\in\{1,\ldots,m(l)\}}$ is arranged on a spatial rectangular grid with grid points $\{\vec{u}_j^{\,l}\}_{1\le j\le m(l)}$, where $\vec{u}_j^{\,l} \in [a_1, a_2] \times [b_1, b_2] \subset \mathbb{R}^2, \forall j \in \{1, \ldots, m(l)\}$.

Definition 6.1.1. *For a unimodal, symmetric one-dimensional radial function ϕ, the Lattice Krig basis function ϕ_j^l for the grid point $\vec{u}_j^{\,l}$ is defined and constructed as follows:*

$$
\phi_j^l(\vec{x}) := \phi\left(\frac{||\vec{x} - \vec{u}_j^{\,l}||}{\theta_l}\right), \theta_l > 0 \qquad (6.5)
$$

Remark 6.1.1. *In the* `LatticeKrig` *R package [Nychka et al., 2015] the radial function $\phi : \mathbb{R} \to \mathbb{R}$ is chosen by default as a* ***Wendland*** *function [Wendland, 1995] defined as:*

$$
\phi(d) = \begin{cases} \frac{1}{3}(1-d)^6(35d^2 + 18d + 3), & \text{if } d \in [0,1], \\ 0, & \text{else,} \end{cases} \qquad (6.6)
$$

where d stands for the scaled distances between the grid points in one up to three dimensions, i.e., ϕ is a radial function with compact support and attains the value 0 for all scaled distances larger than 1.

Geometrically speaking, the set of basis functions at each level consists of bumps that are centered at the grid points $\vec{u}_j^l$. The overlap of these functions is thereby controlled by the parameter θ_l (in (6.5)) and in the `LatticeKrig` R package [Nychka et al., 2015] this overlap is set to 2.5 times the grid spacing by default [Nychka et al., 2015].

Grid spacing and different levels of resolution

In the last paragraph, the construction of the radial functions on different levels of resolution has briefly been introduced. However, it has still not been pointed out how $m(l)$, i.e., the number of grid points and basis functions, are chosen at each resolution level. The spacing of the grid points successively halves from one level to the next, i.e., assuming that the spatial domain of one's data is $[a_1, a_2] \times [b_1, b_2]$ covered with $m_x \times m_y$ grid points, then the grid spacing among the points $\{\vec{u}_j^1\}_j$ is $\delta^{(1)} = \frac{a_2 - a_1}{m_x - 1} = \frac{b_2 - b_1}{m_y - 1}$. By definition, this means that $\delta^{(l+1)} = \delta^{(1)} 2^{-l}$, yielding a sequence of grids for each level with grid points $\{\vec{u}_j^l\}_j$ increasing in number by roughly 4 from level l to level $l+1$. $m(l)$, i.e., the number of basis functions at each level, can then be calculated as follows:

$$m(l) = (m_x - 1)(m_y - 1)4^{l-1} + m_x + m_y + 1.$$

This identity immediately follows from the "halving property" of the grid from one level to the next. In order to leave the overlap of the basis functions constant to 2.5 units of spacing at each level, θ_l needs to change as well in the course of changing levels. The width needs to shrink by the same amount as the spacing shrinks between the grid points, i.e., the parameter θ_l needs to attain the following value:

$$\theta_l = \frac{\theta_1}{2^{l-1}}$$

Property on the distribution of $\vec{y}$

In order to make estimation and prediction possible with the Lattice Krig model, it is useful to derive the following property of the distribution of the data vector $\vec{y}$.

Proposition 6.1.1. *Under the assumption that* $\vec{e} \sim \mathcal{N}_n(0, \sigma^2 W^{-1})$ *in* (6.3)*, it follows that:*

$$\vec{y} = Z\vec{d} + \Phi\vec{c} + \vec{e} \;\sim\; \mathcal{N}_n(Z\vec{d}, \rho\Phi P\Phi^T + \sigma^2 W^{-1}). \tag{6.7}$$

Proof. The proof is straightforward from the fact that $\vec{c}$ and $\vec{e}$ are independent multivariate normal random variables and so is a linear combination of the two:

$$\begin{aligned} E(\vec{y}) &= E(Z\vec{d} + \Phi\vec{c} + \vec{e}) = Z\vec{d} + \Phi E(\vec{c}) + E(\vec{e}) = Z\vec{d} \\ cov(\vec{y}) &= cov(Z\vec{d} + \Phi\vec{c} + \vec{e}) \underset{\vec{c} \perp \vec{e}}{=} \Phi cov(\vec{c})\Phi^T + cov(\vec{e}) = \Phi P \Phi^T + \sigma^2 W^{-1}. \end{aligned}$$

□

6.1.2 The role of GMRF in Lattice Krig

It still has not been discussed how P, i.e., the covariance of the coefficients $\vec{c}$, is constructed. It should be recalled that $\vec{c}^l \sim \mathcal{N}_{m(l)}(0, \rho P_l)$, where P_l may depend on additional parameters. The dependence among the c_j^l at each level l can, therefore, be modeled by a GMRF or more specifically a spatial autoregressive model of order 1 with precision $Q_l = (\rho P_l)^{-1}$. The spatial autoregressive model of order 1 looks as follows:

$$(4 + \kappa_l^2)c_j^l - \sum_{k \in N_j} c_k^l = e_j^l, \text{where } \vec{e}^l \sim \mathcal{N}_{m(l)}(0, \rho I_{m(l)\times m(l)}), \kappa_l > 0 \tag{6.8}$$

Remark 6.1.2. *$\vec{e}^l$ in (6.8), as the error terms of the spatial autoregressive model, should not be confused with $\vec{e}$ in (6.7) representing the residuals of the whole model.*

N_j stands for the 1st order neighborhood of the grid point $\vec{u}_j^l$. Additionally, it is assumed that the coefficients between different resolution levels are independent. In some references of Nychka, (6.8) is also denoted by matrix notation as:

$$B^l \vec{c}^l = \vec{e}^l, \text{ where}$$
$$B_{i,j}^l = \begin{cases} 4 + \kappa_l^2, & \text{if } i = j, \\ -1, & \text{if } i \neq j \text{ and } \vec{u}_i^l, \vec{u}_j^l \text{ are neighbor grid points at level } l, \\ 0, & \text{else.} \end{cases}$$

Therefore,

$$\begin{aligned} Q_l &= cov(\vec{c}^l)^{-1} = cov((B^l)^{-1}\vec{e}^l)^{-1} \\ &= \left((B^l)^{-1}\rho(B^l)^{-T}\right)^{-1} = \rho^{-1}(B^l)^T B^l. \end{aligned}$$

Remark 6.1.3. *Unlike in Chapter 5, the precision matrix of the Lattice Krig model is always positive definite, independent of the chosen values for κ_l or other model parameters. By definition of the symmetric matrix, B^l has four off-diagonals filled with -1's (coming from the 4 closest neighbors of a certain pixel in the spatial field) and a diagonal filled with $4 + \kappa^2$'s. The positive definiteness comes from the fact that for any $\vec{x} \in \mathbb{R}^{m(l)}$ the term $\vec{x}^T B^l \vec{x} > 0$ since, intuitively speaking, the four -1's in each column of the matrix cancel the 4 of the diagonal term $4 + \kappa^2$, but can never exceed $4 + \kappa^2$ as $\kappa^2 > 0$. It is also straight forward from the diagonal dominance criterion for symmetric matrices[1]. The positive definiteness of B^l then also implies that Q, as a block matrix consisting of positive matrices B^l, is positive definite. It should be mentioned, that due to the positive definiteness of the Lattice Krig precision Q, the expenses of calculating the Cholesky factor as a criterion on positive definiteness, as done in the previous GMRF chapter, can be avoided.*

In general, one may stack all coefficients $\vec{c}^l$ for different levels of resolution into a vector $\vec{c} = (\vec{c}^1, \ldots, \vec{c}^L)$, where L is the total number of levels. The parameters $\alpha_1, \ldots, \alpha_L$ with $\sum_{l=1}^L \alpha_l = 1$ form an additional set of weights for each level and the total precision $Q_{m\times m}$ consists of the block matrices $(\alpha_l)^{-1}(B^l)^T B^l$ for $l \in \{1, \ldots, L\}$ or more specifically:

[1] A symmetric matrix A is SPD if the following property holds: $A_{ii} - \sum_{j:j\neq i}|A_{ij}| > 0$ [Rue and Held, 2005]

$$Q_{m\times m} = \rho^{-1}\begin{pmatrix} (1/\alpha_1)(B^1)^T B^1 & 0 & \dots & 0 \\ 0 & (1/\alpha_2)(B^2)^T B^2 & \dots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & \cdots & 0 & (1/\alpha_L)(B^L)^T B^L \end{pmatrix}.$$

This block matrix form is induced by the fact that the coefficients are independent between different levels.
Hence, the above construction shows that the covariance of $\vec{c}$ depends on at least $2L$ parameters, namely: $\{\kappa_1, \dots, \kappa_L, \alpha_1, \dots, \alpha_L\}$. That means, the number of parameters, which are to be estimated, depends on the number of resolution levels needed. More levels are needed when there is a largely varying range of data values among few spatial locations.

6.1.3 Estimation and prediction

Estimation

Setting $M_\lambda = \Phi P \Phi^T + \lambda W^{-1}$ and $\lambda = \sigma^2/\rho$ in (6.7) yields $\vec{y} \sim \mathcal{N}_n(Z\vec{d}, \rho M_\lambda)$ and accordingly the log likelihood for $\vec{y}$ is:

$$l(\vec{y}|\rho, P, \lambda, \vec{d}) \;=\; -\frac{1}{2}(\vec{y} - Z\vec{d})^T(\rho M_\lambda)^{-1}(\vec{y} - Z\vec{d}) - \frac{1}{2}\log(|\rho M_\lambda|) - \frac{n}{2}\log(2\pi)$$

In the `LatticeKrig` R package [Nychka et al., 2015], estimates for $\rho, \vec{d}$ and λ are found via Maximum Likelihood estimation.
To find estimates for the coefficients $\vec{c}$, the following identity was used:

Lemma 6.1.1. *If* $\begin{pmatrix}\vec{X}^{(1)} \\ \vec{X}^{(2)}\end{pmatrix} \sim \mathcal{N}_{m+n}\left(\begin{pmatrix}\vec{\mu}^{(1)} \\ \vec{\mu}^{(2)}\end{pmatrix}, \begin{pmatrix}\Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22}\end{pmatrix}\right)$,
then $\vec{X}^{(1)}|\vec{X}^{(2)} \sim \mathcal{N}_m(\vec{\mu}^{(1)} + \Sigma_{12}\Sigma_{22}^{-1}(\vec{X}^{(2)} - \vec{\mu}^{(2)}), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21})$

Here,

$$\begin{pmatrix}\vec{c} \\ \vec{y}\end{pmatrix} \sim \mathcal{N}_{m+n}\left(\begin{pmatrix}0 \\ Z\vec{d}\end{pmatrix}, \begin{pmatrix}\rho M_\lambda & \rho P\Phi^T \\ \rho\Phi P & \rho M_\lambda\end{pmatrix}\right).$$

Lemma 6.1.1 then gives the conditional distribution of $\vec{c}|\vec{y}$:

$$\vec{c}|\vec{y} \sim \mathcal{N}_m(P\Phi^T M_\lambda^{-1}(\vec{y} - Z\vec{d}), \rho P - \rho P\Phi^T(M_\lambda)^{-1}\Phi P)$$

It is then suggested to take the expected value of $\vec{c}|\vec{y}$,

$$\hat{\vec{c}} = P\Phi^T M_\lambda^{-1}(\vec{y} - Z\vec{d}), \tag{6.9}$$

as an estimate for $\vec{c}$.

Remark 6.1.4. (6.9) *shows that* λ *is inversely related to* $\hat{\vec{c}}$.

Prediction

Lattice Krig model prediction, with the above introduced spatial Lattice Krig model, can be done quite easily as follows:

$$\begin{aligned}\hat{\vec{y}}(\vec{x}) &= Z^T(\vec{x})\hat{\vec{d}} + \hat{g}(\vec{x}), \text{ where} \\ \hat{g}(\vec{x}) &= \sum_{j=1}^{m} \phi_j(\vec{x})\hat{c}_j, \text{ where } \hat{c}_j \text{ is chosen as in (6.9).}\end{aligned}$$

6.2 Lattice Krig tests in R

This section introduces different approaches on how to use the Lattice Krig spatial model to find anomalies such as patches with negative correlation to neighbor pixels where the temporal component is fixed. The `LatticeKrig` R package [Nychka et al., 2015] is used to set up the Lattice Krig spatial model, i.e., estimate various spatial model parameters given predefined gridpoints and other parameters that are used to build the spatial model. It must be pointed out that the tests that are presented in the following sections act as indicators rather than significance tests for spatial inhomogneities. All the "LatticeKrig" R package information that is presented in this chapter is based on Nychka et al. [2015].

6.2.1 Lattice Krig setup in R

As already mentioned (see p.76), the spatial Lattice Krig model is set up on different levels of spatial resolutions, where the number of grid points and, therefore, the number of basis functions duplicates in each dimension from one level to the other.
In the R code and tests that have been developed in this thesis, the following Lattice Krig setup is used:

```
LKinfo <- LKrigSetup(x=x, nlevel=3, alpha=c(1/3,1/3,1/3),
                     a.wght=4.05, NC=36, NC.buffer=0, overlap=2.5)
```

There are three different levels of spatial resolution chosen, all levels are weighted the same. $NC = 36$ means that there are 36 node points in the direction of the longitude at the coarsest level of resolution. Furthermore, the overlap between the basis functions is set to 2.5. It should be recalled that 2.5 refers to the relative overlap in units of the spacing at each level and the support of the basis functions can be calculated as $2.5 \cdot \delta_i$ for $i \in \{1, 2, 3\}$, where δ_i represents the spacing between the grid points at level i. x is chosen to be the coordinates from the CMIP5-ng 144×72 rectangular spatial domain. `LKinfo` then has the following properties:

```
#> LKinfo
#Classes for this object are:  LKinfo LKRectangle
#The second class usually will indicate the geometry
#     e.g.,  2-d rectangle is  LKRectangle
#
#Ranges of locations in raw scale:
#        [,1]   [,2]
```

```
#[1,] -178.75 -88.75
#[2,]  178.75  88.75

#Number of levels: 3
#delta scalings: 10.21429 5.107143 2.553571
#with an overlap parameter of  2.5
#alpha:  0.3333333 0.3333333 0.3333333
#a.wght:  4.05 4.05 4.05

#Basis  type: Radial using  WendlandFunction  and Euclidean  distance.
#Basis functions will be normalized

#Total number of basis functions  13003
# Level Basis size
#     1        648  36 18
#     2       2485  71 35
#     3       9870 141 70
#
#Lambda value:  NA
```

One notices that the distances between the grid points (see `delta scalings`) are close to 2.5 at the finest level of resolution, i.e., approximately four $2.5° \times 2.5°$ spatial pixels are summarized by one spatial pixel at the coarsest level of resolution. The distances between the grid points at different levels are also depicted graphically in Figure 6.1.

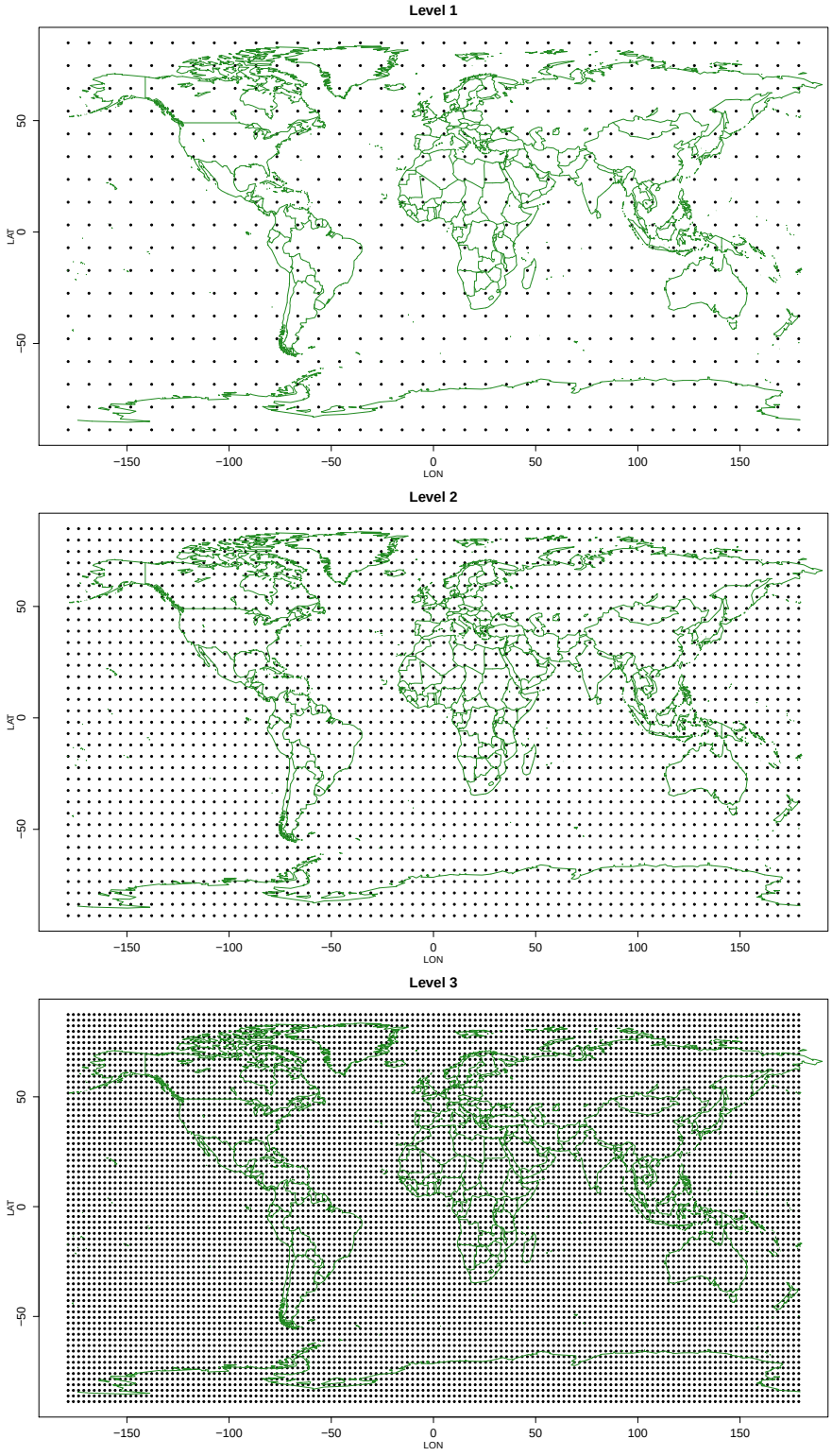

Figure 6.1: The three suggested levels of resolution for Lattice Krig.

6.2.2 The smoothing parameter λ

Proposition 6.1.1 stated that: $cov(\vec{y}) = \rho\Phi P\Phi^T + \sigma^2 W^{-1}$. One can further recall that $\lambda := \frac{\sigma^2}{\rho}$. Therefore, λ can be interpreted as a noise to signal ratio with regards to the

variance of $\vec{y}$ and acts as a smoothing parameter, similarly as in cubic spline models. The larger λ is, the less smooth is the underlying data and the more smoothing needs to be done by the spatial model. On the one hand, these estimates for λ themselves can give an indication on the structure of the spatial field. On the other hand, the smoothing factor λ can be kept constant in order to make comparisons possible among the basis function coefficients of different CMIP5-ng model projections at one specifically fixed point in time. Otherwise, inhomogeneities might not be detected, as the spatial model smooths them out. Thereby, λ should be fixed to a value that is not too small or too large since that could result in over or under fitting the spatial model. The following two sections introduce these two approaches.

λ as an indicator for spatial structure

The estimates for λ, themselves, may be useful as indicators for inhomogeneities. The larger λ is, the more smoothing needs to be done by the spatial model. Large λ estimates can therefore either indicate a model projection with a high resolution or a model projection with suspicious spatial patches. These two phenomena are illustrated on the basis of CMIP5-ng data sets.

Order the model projections by λ
Again, the monthly RCP45 model projections of the Near Surface temperature are chosen. Time is fixed to the 100-th month. For each model projection, the Lattice Krig parameter λ is estimated, yielding the following values for λ:

```
source('/.../getCoord.R')
coord <- getCoordinates(lonInd,latInd)
xNew <- cbind(coord$coordLon,coord$coordLat)
tasLambda <- lambdaLatTest(pathToDir = "/.../tas",
                           T=100,xNew)
> tasLambda
       file                                                lambda
  [1,] "tas_mon_BNU-ESM_rcp45_r1i1p1_g025.nc"        "0.000123653610613044"
  [2,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r10i1p1_g025.nc" "0.000123653610613044"
  [3,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r1i1p1_g025.nc"  "0.000123653610613044"
  [4,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r2i1p1_g025.nc"  "0.000123653610613044"
  [5,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r3i1p1_g025.nc"  "0.000123653610613044"
  [6,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r4i1p1_g025.nc"  "0.000123653610613044"
  [7,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r5i1p1_g025.nc"  "0.000123653610613044"
  [8,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r6i1p1_g025.nc"  "0.000123653610613044"
  [9,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r7i1p1_g025.nc"  "0.000123653610613044"
 [10,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r8i1p1_g025.nc"  "0.000123653610613044"
# ...  (see appendix, Section 8.2.2)
[101,] "tas_mon_GISS-E2-R_rcp45_r1i1p3_g025.nc"      "0.00173972508449444"
[102,] "tas_mon_GISS-E2-H_rcp45_r5i1p2_g025.nc"      "0.00174753604222548"
[103,] "tas_mon_GISS-E2-R_rcp45_r2i1p3_g025.nc"      "0.00175980502931455"
[104,] "tas_mon_GISS-E2-H_rcp45_r4i1p1_g025.nc"      "0.00184457351420222"
[105,] "tas_mon_GISS-E2-H_rcp45_r2i1p3_g025.nc"      "0.00188735119085388"
[106,] "tas_mon_GISS-E2-H_rcp45_r5i1p1_g025.nc"      "0.00188814929359308"
[107,] "tas_mon_GISS-E2-R_rcp45_r3i1p1_g025.nc"      "0.00190870937729969"
[108,] "tas_mon_GISS-E2-R_rcp45_r6i1p1_g025.nc"      "0.00192168723825333"
```

```
[109,] "tas_mon_GISS-E2-R_rcp45_r4i1p2_g025.nc"      "0.00202220248937382"
[110,] "tas_mon_CMCC-CM_rcp45_r1i1p1_g025.nc"        "0.00448169866503329"
```

Once again, the dependence among the ensembles of the same climate model are evident in the estimates for λ. The λ estimates can also be graphically displayed in a histogram (see Figure 6.2):

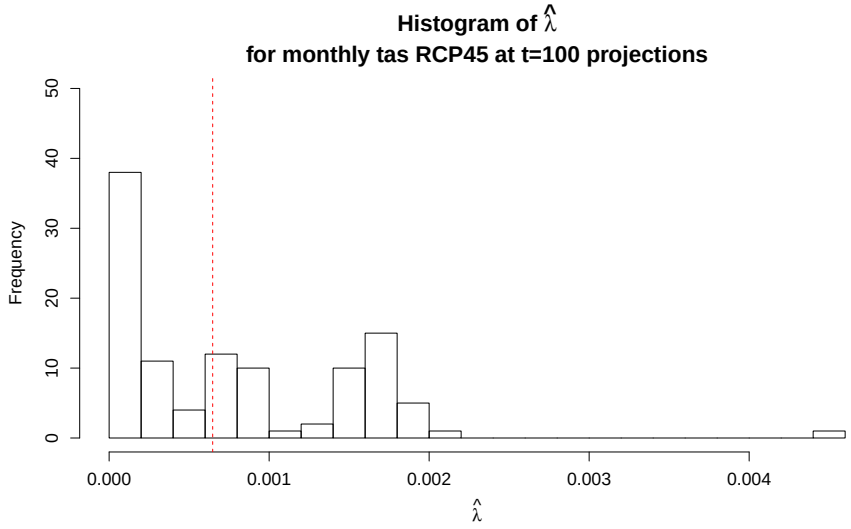

Figure 6.2: Histogram of the Lattice Krig λ estimates of all monthly Near Surface Temperature projections at time $t = 100$ under the RCP45 scenario. The red dashed line corresponds to the median of the λ estimates.

The maximal λ is attained by the CMCC-CM model projection whereas the minimal λ is, for instance, attained by the BNU-ESM projection. What these model projections look like is graphically depicted in Figure 6.3.

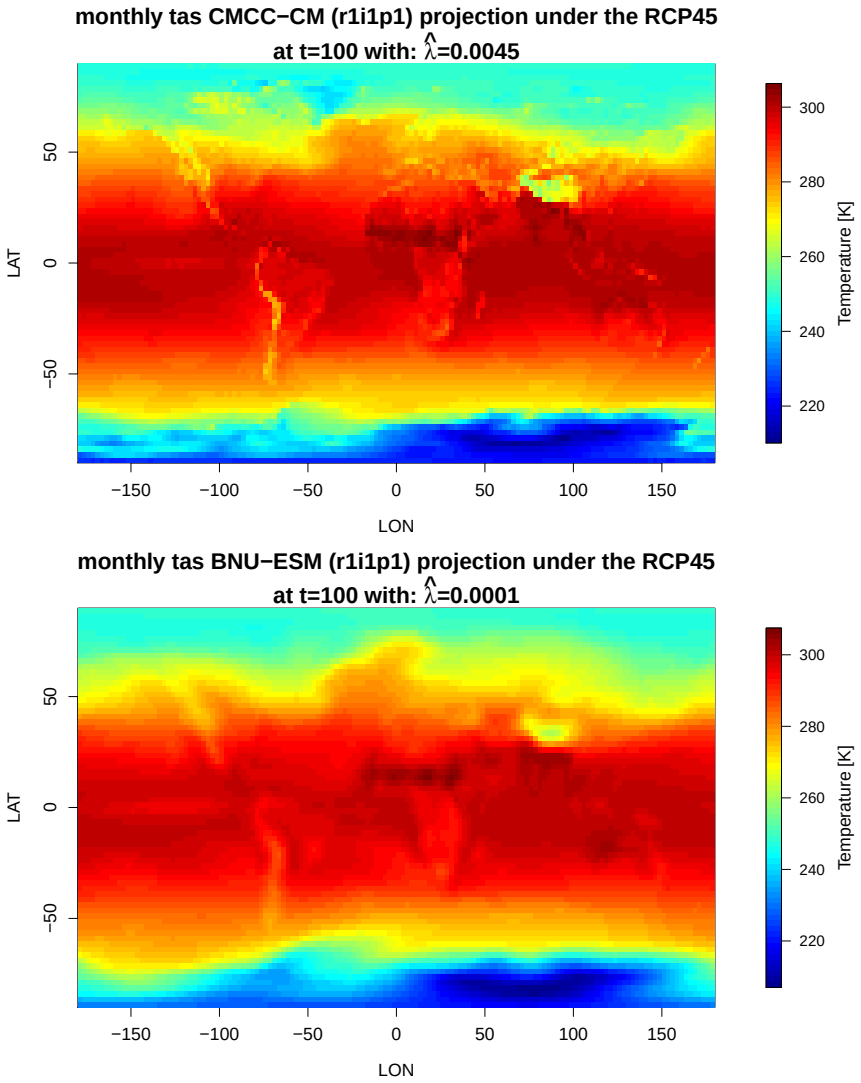

Figure 6.3: Model projections attaining the maximal and minimal λ estimates.

The CMCC-CM projection reveals more details on the Near Surface Temperature whereas the BNU-ESM projection seems almost blurred. The reason for this behavior may be due to different native spatial resolutions of the two models. Nevertheless, visually speaking, both model projections seem to be homogeneous.

λ as an indicator for inhomogeneity

The other possibility is that a large value for λ is induced by an abnormal spatial patch or stripes as illustrated in Figure 6.4.

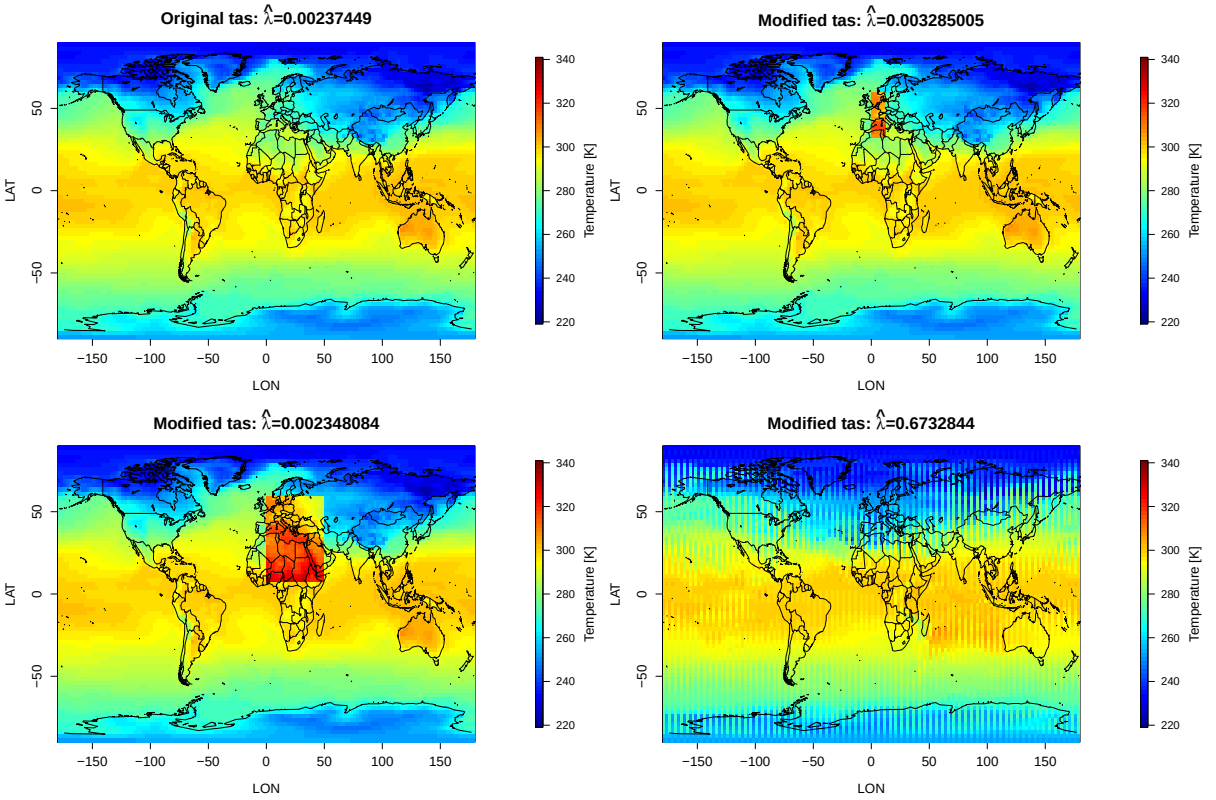

Figure 6.4: Original and modified Near Surface Temperature at time 1. Top left: The original ACCESS1-0 projection at time $t = 1$. Top right: Modified projection with 50 pixels shifted by 30 Kelvin over Europe. Bottom left: Modified projection with 400 pixels shifted by 30 Kelvin over Europe and Africa. Bottom right: Modified projection with mixed up longitude coordinates.

Figure 6.4 shows how λ changes if different sorts of modifications are carried out. From Figure 6.4 it is also apparent that λ does not always increase if inhomogeneities are existent, i.e., some inhomogeneities might remain undetected by λ. The estimates for λ should, therefore, be conceived as a first indicator of the spatial structure of climate data but is not an extremely powerful inhomogeneity detection tool.

Fix λ

As mentioned above, λ should be fixed to a certain value when comparisons of the Lattice Krig basis function coefficients among different model projections are conducted. The difficulty remains in finding the optimal λ. On the one hand, if λ is chosen too large, then the spatial model will give small variability to the fitted values, i.e., it will perform under fitting. On the other hand, if λ is chosen too small, then the spatial model comes close to interpolating the data with large variability among the fitted values, i.e., over fitting is performed. Both cases are illustrated in Figure 6.5 by an example on the Near Surface Temperature data based on the ACCESS1-0 (r1i1p1) projection at time $t = 1$ under the RCP45 scenario.

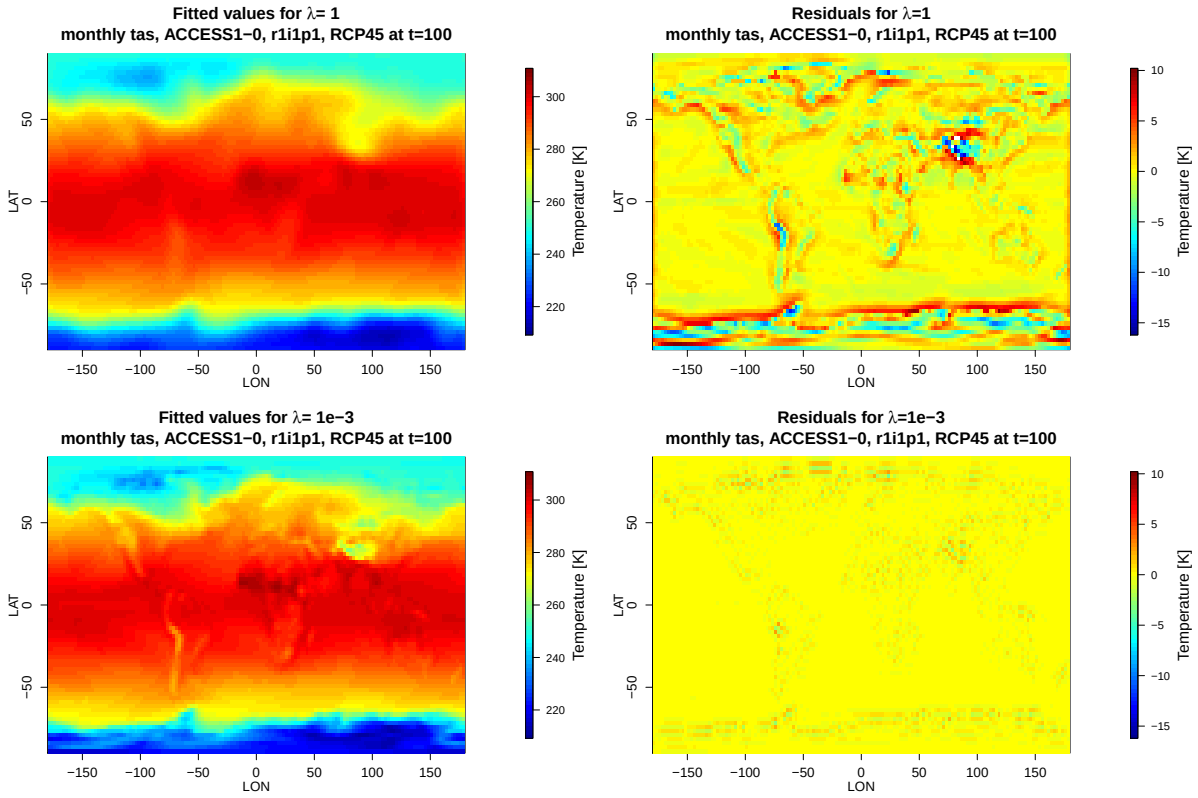

Figure 6.5: Fitted values and residuals of the Lattice Krig model for fixed $\lambda \in \{10^{-3}, 1\}$. Both cases were obtained with the same underlying ACCESS1-0 model projection data of the Near Surface Temperature at time $t = 100$.

The plots show correlated residuals in both cases as there is still some structure apparent at coastal areas. It further illustrates that fixing a large λ yields a blurred picture with large absolute values of the residuals whereas a small λ reveals more details on the fitted values and smaller absolute values of the residuals. Both, over and under fit, are not desirable as they have low predictive power. In the following paragraphs, two approaches are presented on how one can proceed in finding λ systematically.

Weighted mean estimate for λ

A very simple and practical approach is to fix λ to the estimate of the weighted mean representation over all CMIP5-ng model projections under the same scenario, resolution and at the same specific time. This procedure has low computational cost since λ only needs to be estimated once. By that, the tendency, however, is to fix λ to a relatively low value as the weighted mean generally provides a smoother representation than one single model projection, i.e., the danger that comes with this procedure is to produce an under fit with large residuals.

Median of estimates for λ

Another possibility is to take the median over all estimates for λ obtained by all CMIP5-ng model projections under the same scenario, resolution and at the specific time of analysis. The computational cost increases through this method compared to the weighted mean approach since λ has to be estimated for each model projection separately, but it might

lead to a more accurate λ.

Bootstrapping
The most sophisticated approach is to construct a bootstrap confidence interval for λ by resampling the residuals of a specific model projection. This is, however, computationally expensive as λ needs to be estimated for each of the $B = 1,000 - 10,000$ bootstrap samples that are needed to get a reasonable interval. Generating one bootstrap sample with the `boot` R package [Canty and Ripley, 2015] and estimating the λ for this new sample of 144×72 CMIP5-ng spatial values takes approximately one minute, i.e., for 1,000 it would take about 17 hours and for 10,000 bootstrap samples, it would take approximately a week of computation to obtain a reasonable confidence interval. Therefore, it is focused on the weighted-mean and median approaches for the rest of the chapter.

The estimates for λ are illustrated under the weighted mean and median approaches on the basis of the ACCESS1-0 model projection of the monthly Near Surface Temperature under the RCP45 scenario.

Weighted mean estimate for λ:

```
#get weighted mean estimate for lambda
library(ncdf)
library(LatticeKrig)
source('/.../MeanOfFiles.R')
source('/.../getCoord.R')
lonInd <- c(1:144)
latInd <- c(1:72)
coord <- getCoordinates(lonInd,latInd)
xNew <- cbind(coord$coordLon,coord$coordLat)
LKinfo <- LKrigSetup(x=xNew,nlevel=3, alpha=c(1/3,1/3,1/3),
                  a.wght=4.05,NC=36,NC.buffer=0, overlap=2.5)
weightedMean <- meanOfFiles(path ="/.../tas")

meanObj <- LatticeKrig(x=xNew,c(weightedMean[,,100]),LKinfo=LKinfo)
meanObj$lambda.fixed
# > meanObj$lambda.fixed
# [1] 0.0001423206 #--->lambda estimated from the weighted mean.
```

Median of estimates for λ:

```
#...same coordinates (xNew), LKinfo and libraries as above
files <- list.files("/.../tas",full.names = TRUE)
lambda <- numeric(length(files))
for(i in 1:length(files)){
  nc <- open.ncdf(files[i])
  data <- get.var.ncdf(nc)
  obj <- LatticeKrig(x=xNew,c(data[,,100]),LKinfo=LKinfo)
  lambda[i] <- obj$lambda.fixed
}
median(lambda)
#0.0006472452 --->median of the lambdas
```

One sees that the median approach gives a λ that is approximately 6 times larger than obtained by the weighted mean approach, meaning that the weighted mean model provides a smoother representation as expected. Yet, both values for λ are relatively small, i.e., they yield relatively smooth representations of the model projections. Comparison with Figure 6.3 also reveals that both estimates lie within the maximal and minimal range of λ estimates over all monthly Near Surface Temperature RCP45 model projections.
In the next two sections, tests are presented where λ is fixed to a user defined value, which can be fixed to either the median or weighted mean value or any other preferable value.

6.2.3 Lattice Krig test with $\hat{\sigma}_{\mathbf{ML}}$

If one suspects that there are only a few suspicious pixels or regions that abruptly change their pattern, $\hat{\sigma}_{\mathrm{ML}}$, as a control parameter of the covariance of the residuals of the spatial model, may reveal the existence of such patterns if they are severe enough. A few examples of the modified Near Surface Temperature data and their estimates are presented below:

Example 6.2.1. *The following examples illustrate what one can expect with respect to the sensitivity and range of $\hat{\sigma}_{ML}$ for the Near Surface Temperature, with λ fixed to the value* 0.0006472452 *(as obtained by the median approach). The plots, in Figure 6.6, present single spatial* $2.5^\circ \times 2.5^\circ$ *pixels and regions with shifted values. Furthermore, plots of rearranged spatial patches are presented (e.g., obtained by interchanging the Northern and Southern hemisphere values or reversing the longitudes on the Southern hemisphere). The modifications are again made on the basis of the ACCESS1-0 (r1i1p1) projection of the monthly Near Surface Temperature at time* $t = 100$.

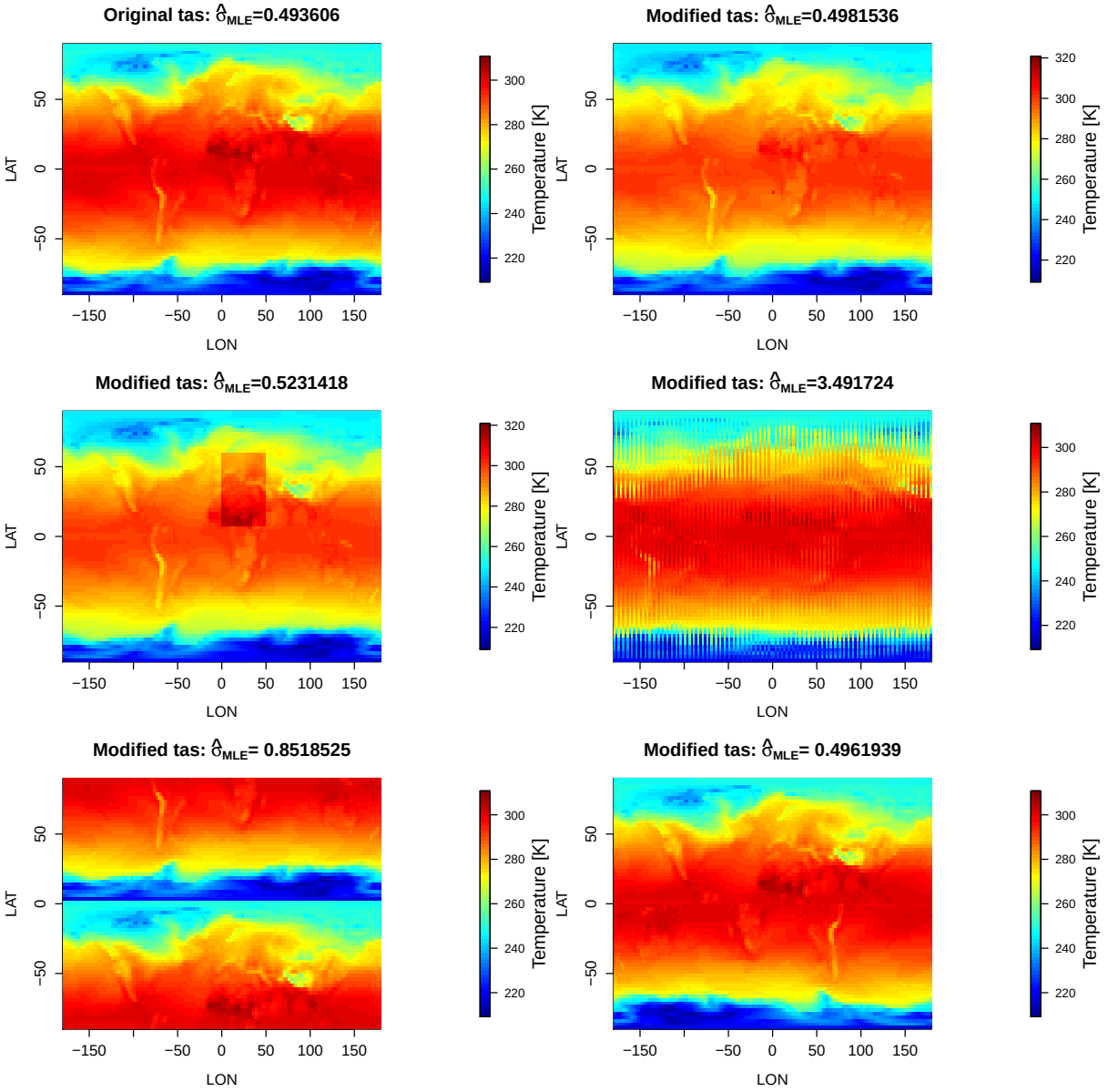

Figure 6.6: Original and modified Near Surface Temperature data and their $\hat{\sigma}_{\mathrm{ML}}$ estimates. Top left: Original tas data obtained from the monthly ACCESS1-0 (r1i1p1) projection at time $t = 100$ under the RCP45 scenario. Top right: One pixel at 16.25°S 1.25°E shifted by 10 Kelvin. Center left: 420 pixels over Europe and Africa shifted by 10 Kelvin, Center right: Stripes with disarranged longitudes. Bottom left: Interchanged Northern and Southern hemisphere values. Bottom right: Reversed longitudes on the Southern hemisphere.

Figure 6.6 shows that $\hat{\sigma}_{ML}$ generally increases with the introduced modifications. Only in the last spatial field a smaller $\hat{\sigma}_{ML}$ is estimated even though the last plot shows unreasonable values for the Near Surface Temperature on the Southern Hemisphere. Section 6.2.4, gives more details on how to detect cases such as depicted in the last plot of Figure 6.6.

overall, $\hat{\sigma}_{\mathrm{ML}}$ might, indeed, be a useful indicator for inhomogeneities such as abnormal spatial patches and, e.g., severe dis-arrangements of longitudes. However, the values of $\hat{\sigma}_{\mathrm{ML}}$ should only be used within the same class of climate model projections, where λ is

fixed to the same value since $\hat{\sigma}_{\mathrm{ML}}$ depends on λ.
For λ and time T fixed, the `sigmaLatTest()` R function provides the ordered estimates of σ for all CMIP5-ng files of the same class (i.e., same variable, scenario and resolution). More application examples of `sigmaLatTest()` can be found in Section 6.5.

6.2.4 Lattice Krig test with a reference model

In contrast to the last section, this section does not focus on the error term, but on the signal of the spatial model which is stored in the GMRF coefficients c. In cases of smooth inhomogeneities such as spatial drifts, $\hat{\sigma}_{\mathrm{ML}}$ might not necessarily show an extreme value since drifts, similar to normal climatic occurrences, can be fit smoothly by the spatial model without producing large error terms. The coefficients c of the Lattice Krig model contains the drift information to some extent, but since the spatial autoregressive model is in between, interpreting the values for c can be difficult. Therefore, this chapter introduces a test using a homogeneous reference spatial field. It is suggested to once again use the weighted model mean as reference, but any other reference is admissible and compatible with the developed `refLatTest()` R function. `refLatTest()` takes a reference spatial field and a candidate field and compares the differences of the estimated coefficients. A summary of the maximum, median and mean is printed out for each Lattice Krig level of resolution. Thereby, major mistakes such as dis-arrangements of longitudes or interchanges of the Northern and Southern hemisphere values should manifest in the coarsest level of resolution, as long as a reasonable reference is chosen.
Again, one can look at some examples of modified Near Surface Temperature data and their `refLatTest()` output.

Example 6.2.2. *For the following experiments, λ is again fixed to* 0.0006472452 *as obtained by the median approach (see p.86). In order to allow comparison, the same experiments and modifications are chosen as in Section 6.2.3 and again the ACCESS1-0 projection at the 100-th month is chosen as a basis. Examples for different models and variables as well as instructions on how to call the* `refLatTest()` *function can be found in Section 6.5. The ACCESS1-0 projection at time $t = 100$ as well as modified versions of the projection are applied to* `refLatTest()` *giving the following results:*

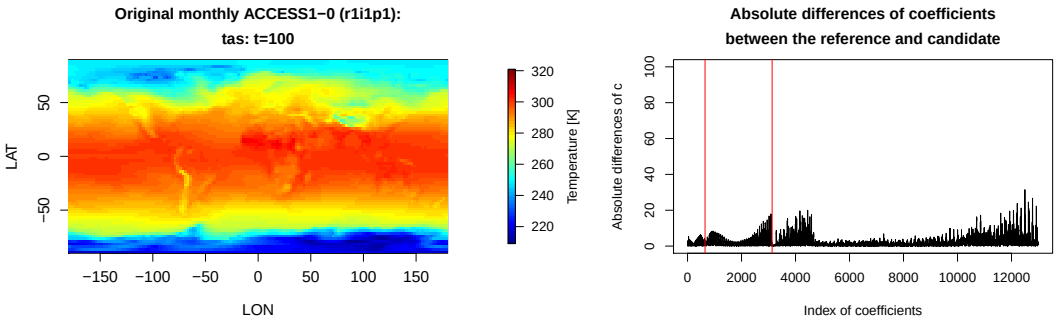

Figure 6.7: Left: Near Surface Temperature at time $t = 100$ of the original ACCESS1-0 projection. Right: Absolute values of the differences of the coefficients obtained by the reference (weighted mean) and the candidate (ACCESS1-0 (r1i1p1) projection). The red lines partition the 3 levels of spatial resolution.

```
# refLatTest() prints out the following statistics
```

```
# on the absolute differences of the coefficients
#              max   median     mean
# level 1  6.42073 1.335006 1.698518
# level 2 17.93001 2.021090 2.706038
# level 3 31.48010 1.104305 2.182321
```

The right plot in Figure 6.7 shows two bumps at each level of resolution. These bumps are induced by the disparity of the coefficients at the poles of the Earth. It seems as if the ACCESS1-0 projection does not completely agree on the Near Surface Temperature with the weighted mean reference in the Arctic and Antarctica. In the third level these disparities become more pronounced as the differences of the coefficients reach a level of 30.
Similarly as in the last section, one can shift an arbitrary pixel value by 10 Kelvin in order to see if that has an effect on the difference of the coefficients.

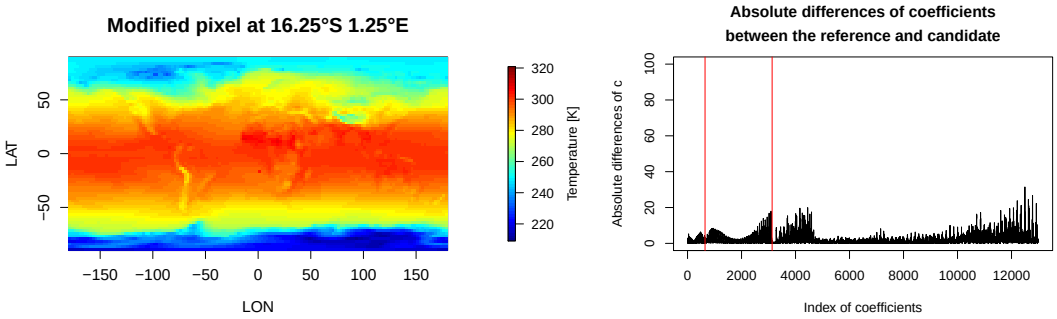

Figure 6.8: Left: Near Surface Temperature at time $t = 100$ of the modified ACCESS1-0 projection. Right: Absolute values of the differences of the coefficients obtained by the reference (weighted mean) and the candidate (ACCESS1-0 projection with 1 pixel at $16.25°S 1.25°E$ shifted by 10 Kelvin). The red lines partition the levels of resolution.

```
#              max   median     mean
# level 1  6.435539 1.376235 1.716851
# level 2 17.954300 1.981378 2.714559
# level 3 31.473078 1.184928 2.22458
```

The differences of the coefficients do not change in the first two levels of resolution, but in the third resolution level (between index 6,000 and 8,000) there are a few increased differences apparent. The printed out maximum, median and mean statistics change only slightly in the third level as well.
The next experiment shows shifted regional values as in the previous section.

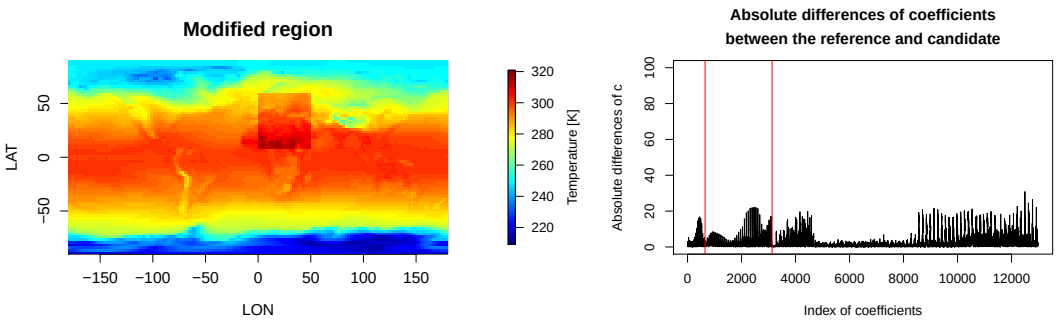

Figure 6.9: Left: Near Surface Temperature at time $t = 100$ of the modified ACCESS1-0 projection. Right: Absolute values of the differences of the coefficients obtained by the reference (weighted mean) and the candidate (ACCESS1-0 projection with 10 K shifted region over Europe). The red lines partition the levels of resolution.

```
#               max   median     mean
# level 1 16.80272 1.612330 2.347272
# level 2 22.16072 2.178299 3.195220
# level 3 30.86285 1.340263 2.786685
```

The regional shift becomes apparent in the printed out summary of the maximum, median and mean at all levels of resolution as well as the absolute difference plot of the coefficients.
The next example shows the most pronounced effect on the coefficients c. The disarrangement of the longitudes leads to extremely high values of the differences of the coefficients at all resolution levels.

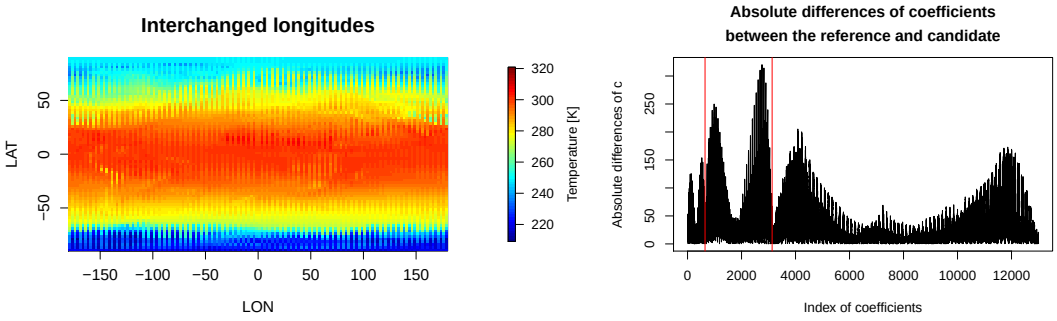

Figure 6.10: Left: Near Surface Temperature at time $t = 100$ of the modified ACCESS1-0 projection. Right: Absolute values of the differences of the coefficients obtained by the reference (weighted mean) and the candidate (interchanged longitudes with underlying ACCESS1-0 data). The red lines partition the levels of resolution.

```
#               max   median     mean
# level 1 153.9044 33.40924 41.82434
# level 2 320.2716 55.30007 79.37474
# level 3 205.2572 22.02993 33.21319
```

The dis-arrangement of the longitudes leads to enormously high differences of the coefficients at all levels of resolution. This is coherent with the $\hat{\sigma}_{ML}$ estimates in the last section.

At last, the spatial field with the interchanged longitudes on the Southern hemisphere is applied to ***refLatTest()****.*

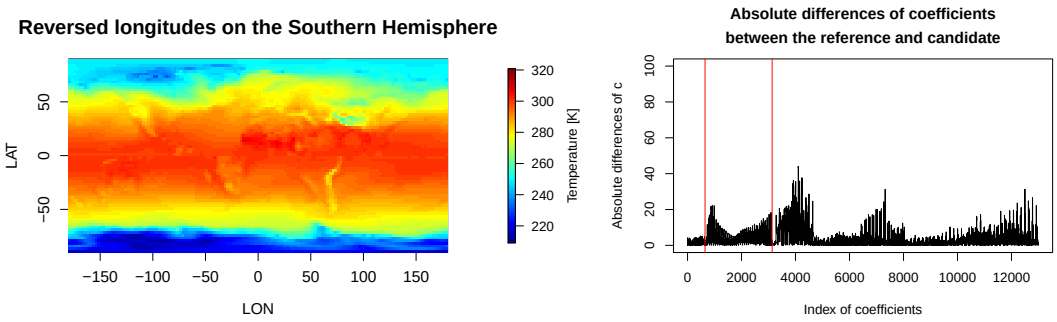

Figure 6.11: Left: Near Surface Temperature at time $t = 100$ of the modified ACCESS1-0 projection. Right: Absolute values of the differences of the coefficients obtained by the reference (weighted mean) and the candidate (reversed longitudes on the Southern Hemisphere based on the ACCESS1-0 (r1i1p1) projection). The red lines partition the levels of resolution.

```
#                max    median     mean
# level 1  5.230335 1.695255 1.839952
# level 2 22.235646 4.846252 5.474112
# level 3 43.969815 1.666811 3.655357
```

The modification mainly reveals slightly higher differences in the finest resolution level (apparent between index 6000 and 8000, i.e., the Equatorial region). There are also changes apparent in the first two levels, these seem to be rather insignificant.

`refLatTest()` does again not provide a significance level, but it gives an idea whether certain model projections appear suspicious or not in their overall structure in comparison to a reference series. Since there are different levels of resolution, the printed out max, median and mean summaries for each level, give an idea on whether and at what scale or level a certain projection appears suspicious.
Again, it needs to be mentioned that the choice of λ also influences these differences of the coefficients. This is due to the dependence of the estimates of the coefficients and λ, see equation (6.9). Therefore, it is wrong to set a certain bound as a threshold without considering the value for λ. This relation of the absolute difference and λ is inverse, i.e., if λ is chosen relatively large, the differences of the coefficients will get smaller whereas for λ chosen relatively small, the differences of the coefficients will become larger.

6.3 Runtime

The expensive part of the analysis with `refLatTest()` and `sigmaLatTest()` is mainly the estimation of the λ if a bootstrap or the median approach is chosen. Estimating the λ under the median approach takes approximately 100 minutes. Applying the `sigmaLatTest()` is less expensive, if λ is fixed. It takes approximately 20 minutes. `refLatTest()` is then the fastest with a runtime of approximately 20 seconds. However, `refLatTest()` also only gives information on one specific model projection whereas

`sigmaLatTest()` gives information on every projection of a whole class of climate projections. The exact CPU time that has been measured can be found in the examples in Section 6.5.

6.4 Inhomogeneity detection performance

The developed Lattice Krig methods, unlike the MGMRF and SNHT tests, only investigate a spatial patch at a fixed point in time. Therefore, any inhomogeneities that evolve over time cannot be detected. Nevertheless, the Lattice Krig methods, might be useful as an indicator for the general structure of the projection such as the smoothness (represented by an estimate for λ) or the amount of abrupt changes in the spatial patch (represented by $\hat{\sigma}_{\mathrm{ML}}$). `refLatTest()` further provides some measure of distance of the spatial patch to a reference spatial structure at different levels of spatial resolution.

6.5 Lattice Krig methods on the CMIP5-ng data

This section provides results and interpretations of the application of the methods `sigmaLatTest()` and `refLatTest()` on CMIP5-ng data sets. Since these Lattice Krig methods operate on climate data with a fixed time component, the results cannot be compared one-to-one with the results in the other CMIP5-ng application Sections 4.5 and 5.5. Since the time component is fixed, it can be looked at the whole Earth as a spatial domain.

6.5.1 Monthly Near Surface Temperature at time $t = 100$

Analogously as in the previous application sections, the monthly Near Surface Temperature is applied to the function `sigmaLatTest()` and `refLatTest()`.

`sigmaLatTest()`

Below, one sees how a user can call `sigmaLatTest()` as well as its output:

```
library(ncdf)
library(LatticeKrig)
source('/.../sigmaLatTest.R')
source('/.../getCoord.R')
source('/.../getLambda.R')
lonInd <- c(1:144)
latInd <- c(1:72)
coord <- getCoordinates(lonInd,latInd)
xNew <- cbind(coord$coordLon,coord$coordLat)
system.time(lambda <- getLambda(pathToDir = "/.../tas",xNew = xNew,
                    type = "median",T = 100))
# system.time output:
#     user   system  elapsed
# 5828.201  100.733 6021.918 --> approximately 100 minutes
system.time(sigmaTas <- sigmaLatTest(pathToDir = "/.../tas",
                      lambda =0.0006472452,T = 100,xNew))
# sytem.time output:
#     user   system  elapsed
```

```
# 1310.319   22.809 1332.466  --> approximately 22 minutes

> sigmaTas
       file                                              sigma
  [1,] "tas_mon_FIO-ESM_rcp45_r3i1p1_g025.nc"        "0.11006772222915"
  [2,] "tas_mon_FIO-ESM_rcp45_r1i1p1_g025.nc"        "0.11366495539389"
  [3,] "tas_mon_FIO-ESM_rcp45_r2i1p1_g025.nc"        "0.115593458668153"
  [4,] "tas_mon_BNU-ESM_rcp45_r1i1p1_g025.nc"        "0.122867289754893"
  [5,] "tas_mon_FGOALS-g2_rcp45_r1i1p1_g025.nc"      "0.142121953939162"
  [6,] "tas_mon_CSIRO-Mk3L-1-2_rcp45_r1i2p1_g025.nc" "0.152685116458957"
  [7,] "tas_mon_CSIRO-Mk3L-1-2_rcp45_r3i2p1_g025.nc" "0.153301490720799"
  [8,] "tas_mon_CSIRO-Mk3L-1-2_rcp45_r2i2p1_g025.nc" "0.157096157213068"
  [9,] "tas_mon_bcc-csm1-1_rcp45_r1i1p1_g025.nc"     "0.163111081940897"
  #... (see appendix, Section 8.2.2)
[104,] "tas_mon_GISS-E2-H_rcp45_r2i1p3_g025.nc"      "0.572225686262292"
[105,] "tas_mon_GISS-E2-H_rcp45_r5i1p3_g025.nc"      "0.572325813631521"
[106,] "tas_mon_GISS-E2-H-CC_rcp45_r1i1p1_g025.nc"   "0.57613901948118"
[107,] "tas_mon_GISS-E2-H_rcp45_r2i1p2_g025.nc"      "0.577021593947984"
[108,] "tas_mon_GISS-E2-H_rcp45_r3i1p1_g025.nc"      "0.578371145557919"
[109,] "tas_mon_GISS-E2-H_rcp45_r1i1p1_g025.nc"      "0.579337834948866"
[110,] "tas_mon_CMCC-CM_rcp45_r1i1p1_g025.nc"        "0.681343428572589"
```

If compared to Section 6.2.2, the ordering of the model projections by $\hat{\sigma}_{\mathrm{ML}}$ with a fixed λ is more or less the same as if ordered by $\hat{\lambda}$. One shall recall that λ depends on the values for σ since λ was defined as: $\lambda = \frac{\sigma^2}{\rho}$. If ρ as the variance of the signal in the data is more or less constant over all model projections then the estimates for λ and σ are similar.

`refLatTest()`

According to $\hat{\sigma}_{\mathrm{ML}}$, the most suspicious projection was produced by the CMCC-CM (r1i1p1) model. Therefore, this projection is further analyzed with the `refLatTest()`:

```
library(ncdf)
library(LatticeKrig)
source('/...refLatTest.R')
source('/.../getCoord.R')
source('/.../MeanOfFiles.R')
weightedMeanTas <- meanOfFiles(path ="/.../tas")
time <- 100
lonInd <- c(1:144)
latInd <- c(1:72)
coord <- getCoordinates(lonInd,latInd)
xNew <- cbind(coord$coordLon,coord$coordLat)
file <- '/.../tas_mon_CMCC-CM_rcp45_r1i1p1_g025.nc'
nc <- open.ncdf(file)
data <- get.var.ncdf(nc)
data <- data[,,time]
close.ncdf(nc)
system.time(outTas <- refLatTest(candidate= data,
                reference=weightedMeanTas[,,time],xNew,lambda = 0.0006472452))
# system.time() gives
#   user  system elapsed
```

```
# 22.752   0.190  22.927 --> approximately 20 seconds


# printout by refLatTest():
#               max   median     mean
# level 1  4.510782 1.911667 1.920838
# level 2 16.554764 1.808701 2.326461
# level 3 26.879113 1.116658 2.144481
```

In absolute terms, the maximum, median and mean do not seem to be extremely high. Yet, one should be cautious with interpretations based on these simple statistics (max, median and mean) as mentioned before. To gain more information, the absolute difference of the coefficients have been plotted (see Figure 6.12).

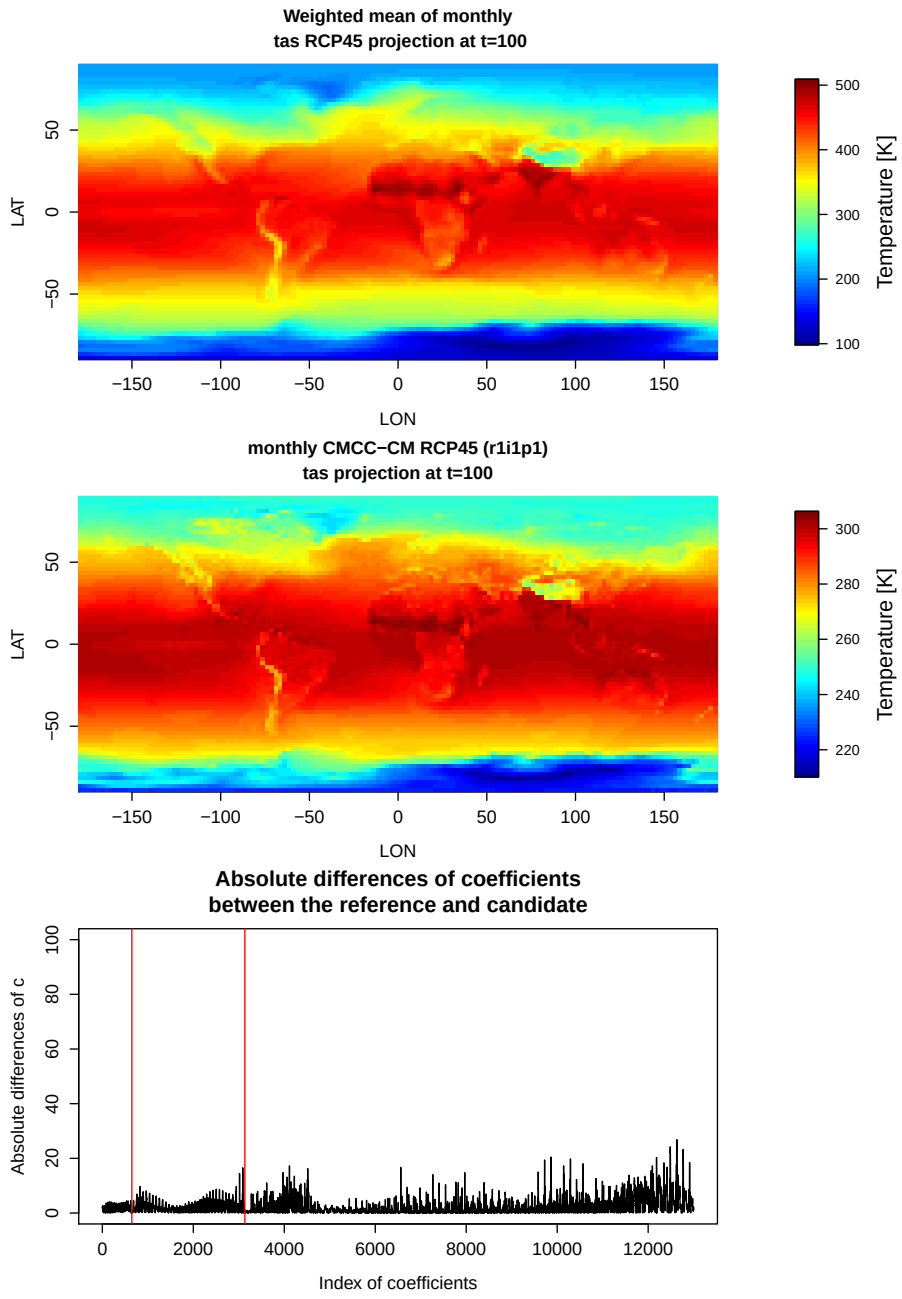

Figure 6.12: Top: Weighted mean of monthly Near Surface Temperature under the RCP45 scenario at time $t = 100$. Center: Near Surface Temperature Projection of the CMCC-CM (r1i1p1) model under the RCP45 scenario at time $t = 100$. Bottom: Differences of the Lattice Krig coefficients obtained by the function `refLatTest()`.

Figure 6.12 shows that the disparity is mainly evident in the finest level of resolution which means that the model projection generally agrees with the weighted mean projection. The largest disparities are apparent in Antarctica, followed by the Arctic and Equatorial regions.

6.5.2 Monthly Surface Upwelling Longwave Radiation at $t = 100$

Similarly as in the last application sections, the Monthly Surface Upwelling Longwave Radiation is investigated under the RCP45 scenario. For the analysis with Lattice Krig, the time component is again fixed to $t = 100$.

`sigmaLatTest()`

The `sigmaLatTest()` function can be called as illustrated above. The median approach has again been applied to calculate the λ, giving the following values for $\hat{\sigma}_{\text{ML}}$

```
# lambda$med
# [1] 0.001336948
# xNew as obtained before.
system.time(sigmaRlus <- sigmaLatTest(pathToDir = "/.../rlus",
                       lambda=0.001336948,T=100,xNew))
#     user   system  elapsed
# 1192.573   35.530 1227.627
> sigmaRlus
       file                                                sigma
  [1,] "rlus_mon_bcc-csm1-1_rcp45_r1i1p1_g025.nc"     "1.12257069943875"
  [2,] "rlus_mon_FGOALS-g2_rcp45_r1i1p1_g025.nc"      "1.17189204415031"
  [3,] "rlus_mon_BNU-ESM_rcp45_r1i1p1_g025.nc"        "1.22784408801046"
  [4,] "rlus_mon_CSIRO-Mk3L-1-2_rcp45_r3i2p1_g025.nc" "1.34032649858499"
  [5,] "rlus_mon_MIROC-ESM_rcp45_r1i1p1_g025.nc"      "1.34964408479685"
  [6,] "rlus_mon_CSIRO-Mk3L-1-2_rcp45_r1i2p1_g025.nc" "1.35192757013706"
  [7,] "rlus_mon_MIROC-ESM-CHEM_rcp45_r1i1p1_g025.nc" "1.35490311136217"
  [8,] "rlus_mon_CSIRO-Mk3L-1-2_rcp45_r2i2p1_g025.nc" "1.35559110276213"
  #... (see appendix, Section 8.2.2)
 [92,] "rlus_mon_GISS-E2-R_rcp45_r6i1p3_g025.nc"      "3.4046325011337"
 [93,] "rlus_mon_GISS-E2-R_rcp45_r6i1p1_g025.nc"      "3.4063820054556"
 [94,] "rlus_mon_GISS-E2-H_rcp45_r3i1p2_g025.nc"      "3.41856638908484"
 [95,] "rlus_mon_GISS-E2-H_rcp45_r2i1p2_g025.nc"      "3.44570652269342"
 [96,] "rlus_mon_CESM1-CAM5_rcp45_r1i1p1_g025.nc"     "3.53904229858017"
 [97,] "rlus_mon_CESM1-CAM5_rcp45_r3i1p1_g025.nc"     "3.63363356344099"
 [98,] "rlus_mon_CESM1-CAM5_rcp45_r2i1p1_g025.nc"     "3.65789195162577"
 [99,] "rlus_mon_CNRM-CM5_rcp45_r1i1p1_g025.nc"       "3.82696568740668"
[100,] "rlus_mon_CMCC-CM_rcp45_r1i1p1_g025.nc"        "4.68354930141825
```

Interestingly, the same model (CMCC-CM) attains the maximum value as in the Near Surface Temperature case.

`refLatTest()`

Again, one can look at the CMCC-CM projection of the rlus and analyze it further with the function `refLatTest()` giving the following ouput:

```
#               max   median     mean
# level 1  14.26672 4.964832 5.425929
# level 2  43.39104 6.700173 8.509525
# level 3 108.44810 4.671224 8.322322
```

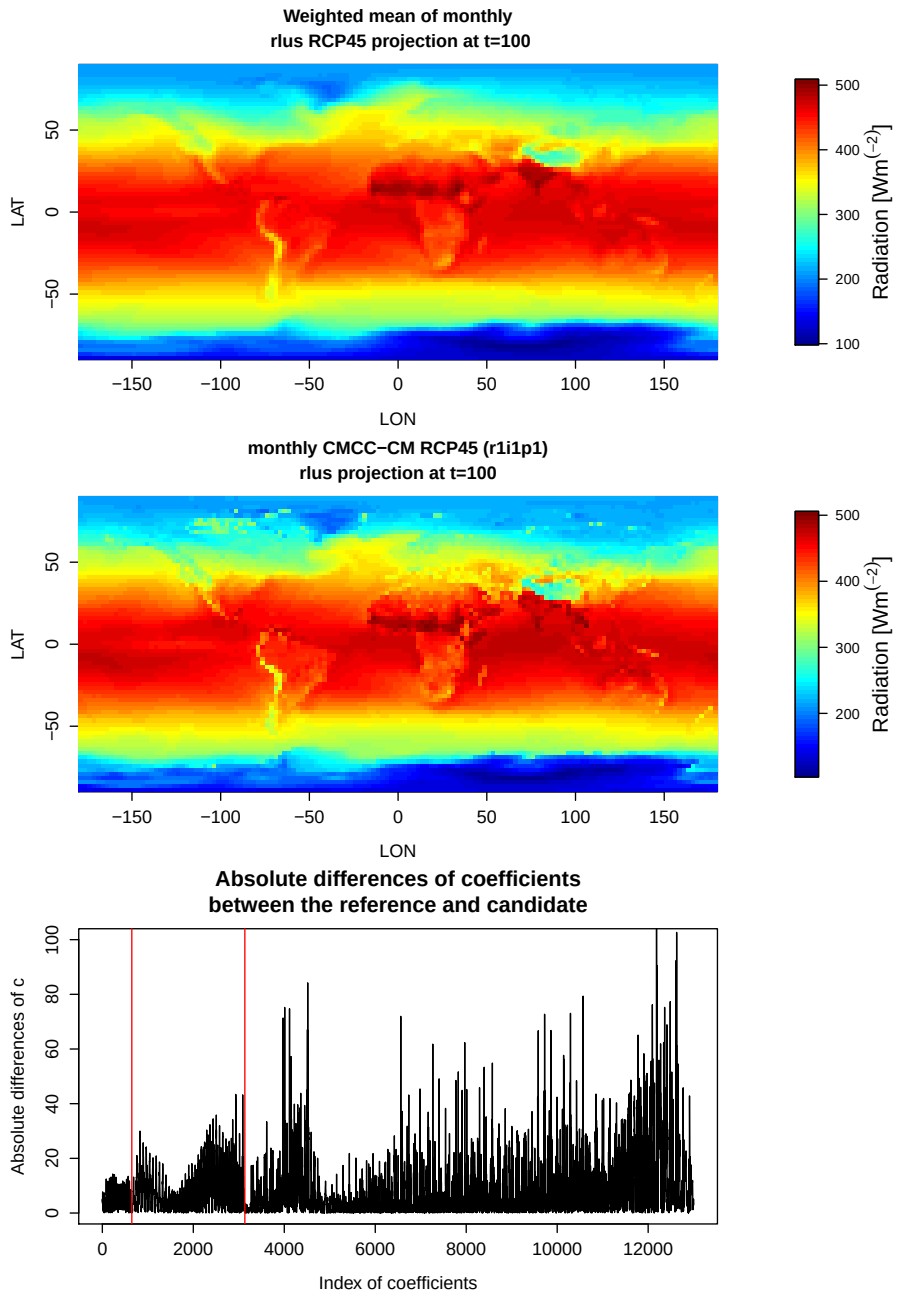

Figure 6.13: Top: Weighted mean of monthly Surface Upwelling Longwave Radiation under the RCP45 scenario at time $t = 100$. Center: Surface Upwelling Longwave Radiation projection of the CMCC-CM (r1i1p1) model under the RCP45 scenario at time $t = 100$. Bottom: Differences of the Lattice Krig coefficients obtained by the function `refLatTest()`.

This time, $\hat{\sigma}_{\mathrm{ML}}$ attains much larger values than in the Near Surface Temperature case. This might be due to badly chosen fixed λ or due to the lack of overall quality of the monthly Surface Upwelling Longwave Radiation projections at time $t = 100$. One may remember that the `pairwiseSNHT()` has detected relatively more inhomogeneities in the variable Surface Upwelling Longwave Radiation than in the Near Surface Temperature projection. Therefore, it might also be the case that the high differences of the coefficients, which are displayed in Figure 6.13, come from actual inhomogeneities.

Chapter 7

Conclusion and outlook

After presenting the CMIP5-ng analysis framework that has been developed in this thesis, this section serves to discuss some of the most important findings that have been made across all chapters and give an outlook on further research. This is done in two sections, whereby the first one focuses on the setup of the R framework and the second one provides the findings obtained by its applications on the CMIP5-ng data sets respectively.

7.1 Setup of the statistical framework

Statistical tools based on the SNHT, GMRF and Lattice Krig have been set up in order to find anomalies in the CMIP5-ng data pool. The methods have been selected due to their complementary nature in finding different types of inhomogeneities as well as their potential to ensure affordable computational cost if applied to large data sets.
The `pairwiseSNHT()` function, presented in Chapter 4, has comparatively fast runtime, which is partly due to the modified SNHT test statistic that does not take into account T values (as in Alexandersson's version) but only $2N << T$. Applications have suggested that the `pairwiseSNHT()` is more efficient than the `gmrfHomogeneityTestComp()` in finding local inhomogeneities (see application/runtime Sections 4.5 and 5.5.2/4.4 and 5.4.2). Yet, global shifts cannot be detected by the `pairwiseSNHT()`. In this respect, `gmrfHomogeneityTestComp()` still gives the best results. Nevertheless, the runtime and stability of the `gmrfHomogeneityTestComp()` are still non-optimal, which is predominantly due to the discontinuity in the likelihood function at the boundaries of the valid parameter space which often results in non-convergence of the `optim()` function in R. This issue can be avoided by an analysis of the valid parameter space and fixing parameters. A user, however, must be aware of the fact that fixing parameters under the likelihood of the null and alternative hypotheses can lead to inaccuracies and type I errors if the parameter is fixed to a value that is too distant from the true value. However, it can be added that an experiment (see Section 5.4.1) has shown that fixing the parameter b in a distance of approximately 0.2 from the true value did only yield slight deviations of the $\hat{a}_{\mathrm{ML}}$ estimates when compared to the reference simulations where b was not fixed.
Regarding the runtime, Sections 4.4 and 5.4.2 have suggested that the `gmrfHomogeneity-TestComp()` and the `pairwiseSNHT()` prefer data in a narrow spatial but large temporal region. Restricting the field of analysis to a narrow spatial region, however, yields to large undetected spatial regions that have been shifted or rearranged. To compensate for this deficiency, the Lattice Krig methods `refLatTest()` and `sigmaLatTest()` have been implemented. Even though they only act as indicators and do not give information on significance, they can be useful to find spatially suspicious regions anywhere on Earth at a fixed specific time. Another advantage in the usage of the Lattice Krig spatial model is its stability as the diagonally dominant blocks of the precision matrix always yield positive

definiteness.
Overall, the complementary aspect of the framework's methods has been achieved, whereas for runtime optimization, one may want to implement the framework or merely the `gmrfHomogeneityTest()` function in the C programming language. C is a lower level programming language and, hence, runtime is less influenced by elements of abstraction to allow an easy to use programming surface as in R.
In conclusion, the existing framework as presented in this thesis could be improved by further research on the following aspects:

1. GMRF: Can the relationship between the differences $\hat{b}_{ML,H_0} - \hat{b}_{ML,H_1}$, $\hat{c}_{ML,H_0} - \hat{c}_{ML,H_1}$, $\hat{f}_{ML,H_1} - \hat{f}_{ML,H_0}$ and the global shift height $\hat{a}_{\mathrm{ML}}$ be quantified?

2. GMRF: What is the quantitative effect on the likelihood ratio statistic when fixing the parameters b, c, f to estimates that are a certain distance away from the true parameter values?

3. GMRF: Is there a good parametrization of the spatial model yielding a diagonally dominant precision matrix that guarantees more overall stability when finding the MLEs?

4. GMRF: Is there a computationally efficient and useful alternative to GAMM to remove seasonality and trends in climate data projections?

5. Lattice Krig: Is it possible to develop a significance test on the basis of Lattice Krig for detecting spatial inhomogeneities?

7.2 Application of the framework on CMIP5-ng data

This thesis has focused on the development of an R framework. Its usage has been illustrated on the basis of a few model projections on the Near Surface Temperature as well as the Surface Upwelling Longwave Radiation under the RCP45 scenario. Some of the observations and conjectures, which one may want to analyze further, are provided in this section.
The application of the SNHT on the Surface Upwelling Longwave Radiation in Section 4.5 suggests that ocean regions may have been less thoroughly analyzed with more detected inhomogeneities than in regions of land. Thus, it is suggested to further analyze ocean regions for less commonly used model projections of variables such as the Surface Upwelling Longwave Radiaton.
The application of the GMRF with the `gmrfHomogeneityTestComp()` function has given similar results as the SNHT when trends and seasonality were removed using the GAMM. The weighted mean approach has resulted in a less accurate removal of these climatic occurrences and, hence, in impractical output of the `gmrfHomogeneityTestComp()`. Nonetheless, the weighted mean is useful for building the SSD in order to reveal outlier models.
The application of Lattice Krig to the Near Surface Temperature sample data suggested that, though the spatial smoothness of model projections (estimated by λ) vary between different climate variables, model projections can be ordered according to the spatial smoothness, which can be generated by the λ estimates of the Lattice Krig model. The order of the model projections according to the λ's only differs slightly between the Near Surface Temperature and Surface Upwelling Longwave radiation climate variable projections. At this point, it should be noticed that the Near Surface Temperature and Surface

Upwelling Longwave radiation are strongly dependent variables. Hence, one may want to analyze if the phenomenon of spatial smoothness as an attribute of a climate model holds for other climate variables as well. Through the application of `refLatTest()`, it has become evident that certain model projections of the CMIP5-ng diverge in their values at polar regions. Thus, further analysis may want to be conducted in these spatial regions.

Overall, the framework developed is a working analysis tool and a contribution to finding erroneous simulation runs in the CMIP5-ng data pool. Homogenization and further investigation of non-climatic anomalies in the climate data, which can potentially be detected by statistical tools and frameworks as provided in this thesis, can yield more representative climate simulation runs. Finally, this has the potential to reduce the uncertainty in climate scenario projections and, among other things, is a first step in generating more accurate future climate predictions.

Acronyms

ACF Autocorrelation Function. 60, 61, 63

AR Assessment Reports. 5

CMIP Coupled Model Intercomparison Project. I, 1, 5, 6, 11, 12

CMIP5 CMIP Phase 5. IV, 1, 2, 6, 8–11, 13, 14, 19

CMIP5-ng CMIP Phase 5 next generation. I, II, IV, 1, 2, 5–7, 13–15, 19, 20, 22–24, 29–32, 40, 41, 46, 54, 58, 59, 61, 62, 66, 74, 79, 82, 86, 87, 90, 94, 100–102

GAMM Generalized Additive Mixed Models. 58–61, 66, 70–72, 101

GMRF Gaussian Markov Random Field. II–IV, 1–4, 39–41, 43, 50–52, 74, 77, 90, 100, 101, 113

IPCC Intergovernmental Panel on Climate Change. 5, 6

MGMRF Multivariate Gaussian Markov Random Field. II, 39, 41, 43–47, 49, 51, 53, 54, 58, 61, 66, 94

MLE Maximum Likelihood Estimate. 4, 45, 54, 56, 101

PACF Partial Autocorrelation Function. 60, 61, 63

RCP Representative Concentration Pathways. 9–11, 58

rlus Surface Upwelling Longwave Radiation. IV, 98

SNHT Standard Normal Homogeneity Test. I, II, IV, 1–4, 20–23, 25–28, 32–34, 53, 66, 68, 71, 94, 100, 101, 113

SPD Symmetric and Positive Definite. 4, 39, 40, 43, 44, 77

SSD standardized Sum of Squared Differences. 64–66, 101

tas Near Surface Temperature. IV, 62, 63, 89

tos Sea Surface Temperature. 16, 17

Chapter 8

Appendix

8.1 Runtime experiments

8.1.1 SNHT: Runtime experiment (Space vs. time)

```
library(reshape2)
install.packages("/.../snht_1.0.4.tar.gz",type="src",repo=NULL)
library(snht)
library(ncdf)
source('/.../getCoord.R')
file <- '/.../tas_mon_ACCESS1-0_rcp45_r1i1p1_g025.nc'
nc <- open.ncdf(file)
data <- get.var.ncdf(nc)
close.ncdf(nc)
baseDataEurope <- data[c(1:3),c(55:57),c(1:100)]
coord <- getCoordinates(c(1:3),c(55:57))
#create coordinates
dist <- (as.matrix(dist(coord)))

dim(baseDataEurope) <- c(dim(baseDataEurope)[1]*
                         dim(baseDataEurope)[2],dim(baseDataEurope)[3])
baseDataEurope <- t(baseDataEurope)
colnames(baseDataEurope) <- "1":"9"
baseData <- data.frame(time=1:100,baseDataEurope)
baseData <- melt(baseData,id.vars="time",variable.name=
                 "location",value.name="data")
baseData$location <- gsub("X","",baseData$location)

system.time(out <- pairwiseSNHT(baseData,dist,k=3,period=10,
                                crit=qchisq(1-0.05/80,df=1),returnStat=F))

# user  system elapsed
# 0.414   0.008   0.431

#vs.  more locations... 10x10x9
baseDataEurope <- data[c(1:10),c(55:64),c(1:9)]
coord <- getCoordinates(c(1:10),c(55:64))
#create coordinates
dist <- (as.matrix(dist(coord)))

dim(baseDataEurope) <- c(dim(baseDataEurope)[1]*
                         dim(baseDataEurope)[2],dim(baseDataEurope)[3])
```

```
baseDataEurope <- t(baseDataEurope)
colnames(baseDataEurope) <- "1":"100"
baseData <- data.frame(time=1:9,baseDataEurope)
baseData <- melt(baseData,id.vars="time",variable.name=
                "location",value.name="data")
baseData$location <- gsub("X","",baseData$location)

system.time(out <- pairwiseSNHT(baseData,dist,k=3,period=2,
                            crit=qchisq(1-0.05/5,df=1),returnStat=F))

# user  system elapsed
# 2.035   0.000   2.029
```

8.1.2 GMRF: Runtime experiment (Space vs. time)

```
library(ncdf)
library(spam)
source('/.../gmrfHomogeneityTest_VR.R')
source('/.../mgmrfPrec.R')


desData <- load("/.../resRCP45ACCESS1") #obtained by GAMM
dim(res) <- c(144,72,2772)

desDataEurope <- res[c(1:3),c(55:57),c(1:100)]
# image.plot(desDataEurope[,,1])
av <- matrix(0,3,3)

for(i in 1:3){
  for(j in 1:3){
    av[i,j] <- mean(desDataEurope[i,j,c(1:100)])
  }
}
timeTaken1 <- system.time(
            out1 <- gmrfHomogeneityTestComp_VR(desDataEurope,
                    type="global",mu=c(av),
                    bStart=0.14,cStart=1,fStart=0.1,
                    sigLevel=0.05,L=1))
# user  system elapsed
# 8.803   0.129   8.879
desDataEurope <- res[c(1:10),c(55:64),c(1:9)]
av <- matrix(0,10,10)
for(i in 1:10){
  for(j in 1:10){

    av[i,j] <- mean(desDataEurope[i,j,c(1:9)])
  }
}

timeTaken2 <- system.time(
            out2 <- gmrfHomogeneityTestComp_VR(desDataEurope,
```

```
                    type="global",mu=c(av),
                    bStart=0.14,cStart=1,fStart=0.1,
                    sigLevel=0.05,L=1))
# > timeTaken2
#    user  system elapsed
# 40.073   0.047  40.091
```

8.2 Output

8.2.1 Preanalysis of data output

```
#timeDim:   number of time units (months, years, seasons)
#max:       maximum of the data values of a specific projection
#min:       minimum of the data values of a specific projection
#mean:      mean of the data data values of a specific projection
#susp:      projection has too many missing values
#           (over 40% for sea-type variables,
#           over 80% for land-type variables,
#           more than 0% for global variables) or values that are outside of
#           a predefined interval.
#ok:        projection is non-suspicious wrt missing values and range of values.
> out
      name                                                timeDim max        min        mean       susp/ok
 [1,] "tos_mon_ACCESS1-0_piControl_r1i1p1_g025.nc"        "6000"  "307.71 *" "271.22 "  "286.95 "  "ok"
 [2,] "tos_mon_ACCESS1-3_piControl_r1i1p1_g025.nc"        "6000"  "305.86 "  "271.24 "  "286.99 "  "ok"
 [3,] "tos_mon_BNU-ESM_piControl_r1i1p1_g025.nc"          "6708"  "306.28 "  "271.36 "  "286.47 "  "ok"
 [4,] "tos_mon_CCSM4_piControl_r1i1p1_g025.nc"            "12612" "305.64 "  "271.12 "  "286.73 "  "ok"
 [5,] "tos_mon_CCSM4_piControl_r2i1p1_g025.nc"            "1872"  "305.28 "  "271.13 "  "286.71 "  "ok"
 [6,] "tos_mon_CCSM4_piControl_r4i1p1_g025.nc"            "600"   "305.23 "  "271.16 "  "286.74 "  "ok"
 [7,] "tos_mon_CESM1-BGC_piControl_r1i1p1_g025.nc"        "6000"  "305.29 "  "271.08 "  "286.76 "  "ok"
 [8,] "tos_mon_CESM1-CAM5-1-FV2_piControl_r1i1p1_g025.nc" "600"   "307.7 *"  "271.22 "  "287 "     "ok"
 [9,] "tos_mon_CESM1-CAM5_piControl_r1i1p1_g025.nc"       "3828"  "306.34 "  "271.24 "  "286.69 "  "ok"
[10,] "tos_mon_CESM1-FASTCHEM_piControl_r1i1p1_g025.nc"   "2664"  "305.51 "  "271.14 "  "286.74 "  "ok"
[11,] "tos_mon_CMCC-CESM_piControl_r1i1p1_g025.nc"        "3324"  "307.64 *" "271.05 *" "286.74 "  "susp"
[12,] "tos_mon_CMCC-CMS_piControl_r1i1p1_g025.nc"         "6000"  "306.93 "  "271.05 *" "286.85 "  "susp"
[13,] "tos_mon_CMCC-CM_piControl_r1i1p1_g025.nc"          "3960"  "305.85 "  "271.11 "  "286.6 "   "susp"
[14,] "tos_mon_CNRM-CM5-2_piControl_r1i1p1_g025.nc"       "4920"  "306.08 "  "270.1 *"  "286.49 "  "ok"
[15,] "tos_mon_CNRM-CM5-2_piControl_r1i1p2_g025.nc"       "1680"  "305.84 "  "270.23 *" "286.43 "  "ok"
[16,] "tos_mon_CNRM-CM5-2_piControl_r1i1p3_g025.nc"       "1680"  "305.9 "   "270.74 *" "286.24 "  "ok"
[17,] "tos_mon_CNRM-CM5-2_piControl_r1i1p4_g025.nc"       "840"   "305.55 "  "270.78 *" "286.26 "  "ok"
[18,] "tos_mon_CNRM-CM5_piControl_r1i1p1_g025.nc"         "10200" "306.07 "  "270.18 *" "286.72 "  "ok"
[19,] "tos_mon_CSIRO-Mk3-6-0_piControl_r1i1p1_g025.nc"    "6000"  "305.95 "  "271.32 "  "290.05 *" "susp"
[20,] "tos_mon_CanESM2_piControl_r1i1p1_g025.nc"          "11952" "306.03 "  "270.85 *" "286.41 "  "ok"
[21,] "tos_mon_EC-EARTH_piControl_r1i1p1_g025.nc"         "5424"  "304.5 *"  "271.06 *" "286.44 "  "ok"
[22,] "tos_mon_FGOALS-s2_piControl_r1i1p1_g025.nc"        "6000"  "304.63 *" "271.35 "  "286.59 "  "ok"
[23,] "tos_mon_FIO-ESM_piControl_r1i1p1_g025.nc"          "9600"  "305.76 "  "271.21 "  "286.51 "  "ok"
[24,] "tos_mon_GFDL-CM3_piControl_r1i1p1_g025.nc"         "9600"  "307.96 *" "271.25 "  "287.2 "   "ok"
[25,] "tos_mon_GFDL-ESM2G_piControl_r1i1p1_g025.nc"       "6000"  "307.15 *" "271.24 "  "286.92 "  "ok"
[26,] "tos_mon_GFDL-ESM2M_piControl_r1i1p1_g025.nc"       "6000"  "306.42 "  "271.25 "  "287.13 "  "ok"
[27,] "tos_mon_GISS-E2-H-CC_piControl_r1i1p1_g025.nc"     "3012"  "305.29 "  "271.26 "  "287.54 "  "susp"
[28,] "tos_mon_GISS-E2-H_piControl_r1i1p2_g025.nc"        "6372"  "305.39 "  "271.22 "  "288.35 *" "susp"
[29,] "tos_mon_GISS-E2-H_piControl_r1i1p3_g025.nc"        "6372"  "305.22 "  "271.22 "  "288.36 *" "susp"
[30,] "tos_mon_GISS-E2-R-CC_piControl_r1i1p1_g025.nc"     "3012"  "305.02 "  "271.25 "  "287.27 "  "susp"
[31,] "tos_mon_GISS-E2-R_piControl_r1i1p141_g025.nc"      "13956" "305.03 "  "271.24 "  "287.27 "  "susp"
[32,] "tos_mon_GISS-E2-R_piControl_r1i1p142_g025.nc"      "1200"  "304.97 *" "271.25 "  "287.43 "  "susp"
[33,] "tos_mon_GISS-E2-R_piControl_r1i1p2_g025.nc"        "6372"  "305.18 "  "271.25 "  "287.39 "  "susp"
[34,] "tos_mon_GISS-E2-R_piControl_r1i1p3_g025.nc"        "6372"  "304.98 "  "271.25 "  "287.37 "  "susp"
[35,] "tos_mon_HadGEM2-AO_piControl_r1i1p1_g025.nc"       "1200"  "307.44 *" "271.35 "  "286.64 "  "ok"
[36,] "tos_mon_HadGEM2-CC_piControl_r1i1p1_g025.nc"       "2880"  "306.93 "  "271.35 "  "286.72 "  "ok"
[37,] "tos_mon_HadGEM2-ES_piControl_r1i1p1_g025.nc"       "6912"  "307.36 *" "271.35 "  "286.91 "  "ok"
[38,] "tos_mon_IPSL-CM5A-LR_piControl_r1i1p1_g025.nc"     "12000" "305.36 "  "271.19 "  "285.64 *" "susp"
[39,] "tos_mon_IPSL-CM5A-MR_piControl_r1i1p1_g025.nc"     "3600"  "304.96 *" "271.15 "  "286.17 "  "susp"
[40,] "tos_mon_IPSL-CM5B-LR_piControl_r1i1p1_g025.nc"     "3600"  "306.4 "   "271.24 "  "287.16 "  "susp"
[41,] "tos_mon_MIROC-ESM_piControl_r1i1p1_g025.nc"        "8160"  "306.71 "  "271.26 "  "286.61 "  "ok"
[42,] "tos_mon_MIROC5_piControl_r1i1p1_g025.nc"           "8040"  "305.86 "  "271.24 "  "286.97 "  "ok"
[43,] "tos_mon_MPI-ESM-LR_piControl_r1i1p1_g025.nc"       "12000" "306.85 "  "271.25 "  "286.4 "   "ok"
[44,] "tos_mon_MPI-ESM-MR_piControl_r1i1p1_g025.nc"       "12000" "309.83 *" "271.25 "  "286.61 "  "ok"
[45,] "tos_mon_MPI-ESM-P_piControl_r1i1p1_g025.nc"        "13872" "307.29 *" "271.25 "  "286.49 "  "ok"
```

```
[46,] "tos_mon_MRI-CGCM3_piControl_r1i1p1_g025.nc"        "6000"  "304.2 *"  "271.29 "  "287.02 "  "ok"
[47,] "tos_mon_NorESM1-ME_piControl_r1i1p1_g025.nc"       "3024"  "305.67 "  "271.33 "  "286.3 "   "ok"
[48,] "tos_mon_NorESM1-M_piControl_r1i1p1_g025.nc"        "6012"  "305.61 "  "271.33 "  "286.58 "  "ok"
[49,] "tos_mon_bcc-csm1-1-m_piControl_r1i1p1_g025.nc"     "4800"  "305.18 "  "271.26 "  "286.4 "   "ok"
[50,] "tos_mon_bcc-csm1-1_piControl_r1i1p1_g025.nc"       "6000"  "305.12 "  "271.26 "  "286.31 "  "ok"
[51,] "tos_mon_inmcm4_piControl_r1i1p1_g025.nc"           "6000"  "305.33 "  "270.73 *" "287.33 "  "ok"
```

8.2.2 Lattice Krig output

The estimates for λ of the monthly Near Surface Temperature Data under the RCP45 at time $t = 100$ are:

```
#lambda refers to time t=100 in each model projection.
       file                                          lambda
  [1,] "tas_mon_BNU-ESM_rcp45_r1i1p1_g025.nc"        "0.000123653610613044"
  [2,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r10i1p1_g025.nc" "0.000123653610613044"
  [3,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r1i1p1_g025.nc"  "0.000123653610613044"
  [4,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r2i1p1_g025.nc"  "0.000123653610613044"
  [5,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r3i1p1_g025.nc"  "0.000123653610613044"
  [6,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r4i1p1_g025.nc"  "0.000123653610613044"
  [7,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r5i1p1_g025.nc"  "0.000123653610613044"
  [8,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r6i1p1_g025.nc"  "0.000123653610613044"
  [9,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r7i1p1_g025.nc"  "0.000123653610613044"
 [10,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r8i1p1_g025.nc"  "0.000123653610613044"
 [11,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r9i1p1_g025.nc"  "0.000123653610613044"
 [12,] "tas_mon_CSIRO-Mk3L-1-2_rcp45_r1i2p1_g025.nc" "0.000123653610613044"
 [13,] "tas_mon_CSIRO-Mk3L-1-2_rcp45_r2i2p1_g025.nc" "0.000123653610613044"
 [14,] "tas_mon_CSIRO-Mk3L-1-2_rcp45_r3i2p1_g025.nc" "0.000123653610613044"
 [15,] "tas_mon_CanESM2_rcp45_r2i1p1_g025.nc"        "0.000123653610613044"
 [16,] "tas_mon_CanESM2_rcp45_r3i1p1_g025.nc"        "0.000123653610613044"
 [17,] "tas_mon_CanESM2_rcp45_r4i1p1_g025.nc"        "0.000123653610613044"
 [18,] "tas_mon_CanESM2_rcp45_r5i1p1_g025.nc"        "0.000123653610613044"
 [19,] "tas_mon_FGOALS-g2_rcp45_r1i1p1_g025.nc"      "0.000123653610613044"
 [20,] "tas_mon_FIO-ESM_rcp45_r1i1p1_g025.nc"        "0.000123653610613044"
 [21,] "tas_mon_FIO-ESM_rcp45_r2i1p1_g025.nc"        "0.000123653610613044"
 [22,] "tas_mon_FIO-ESM_rcp45_r3i1p1_g025.nc"        "0.000123653610613044"
 [23,] "tas_mon_GFDL-CM3_rcp45_r1i1p1_g025.nc"       "0.000123653610613044"
 [24,] "tas_mon_MIROC5_rcp45_r1i1p1_g025.nc"         "0.000123653610613044"
 [25,] "tas_mon_MIROC5_rcp45_r2i1p1_g025.nc"         "0.000123653610613044"
 [26,] "tas_mon_MIROC5_rcp45_r3i1p1_g025.nc"         "0.000123653610613044"
 [27,] "tas_mon_MPI-ESM-LR_rcp45_r1i1p1_g025.nc"     "0.000123653610613044"
 [28,] "tas_mon_MPI-ESM-LR_rcp45_r2i1p1_g025.nc"     "0.000123653610613044"
 [29,] "tas_mon_MPI-ESM-LR_rcp45_r3i1p1_g025.nc"     "0.000123653610613044"
 [30,] "tas_mon_MPI-ESM-MR_rcp45_r2i1p1_g025.nc"     "0.000123653610613044"
 [31,] "tas_mon_MPI-ESM-MR_rcp45_r3i1p1_g025.nc"     "0.000123653610613044"
 [32,] "tas_mon_NorESM1-ME_rcp45_r1i1p1_g025.nc"     "0.000123653610613044"
 [33,] "tas_mon_NorESM1-M_rcp45_r1i1p1_g025.nc"      "0.000123653610613044"
 [34,] "tas_mon_bcc-csm1-1_rcp45_r1i1p1_g025.nc"     "0.000123653610613044"
 [35,] "tas_mon_CanESM2_rcp45_r1i1p1_g025.nc"        "0.000128441199114534"
 [36,] "tas_mon_MPI-ESM-MR_rcp45_r1i1p1_g025.nc"     "0.000130585695878178"
 [37,] "tas_mon_CMCC-CMS_rcp45_r1i1p1_g025.nc"       "0.000155360492284487"
 [38,] "tas_mon_IPSL-CM5B-LR_rcp45_r1i1p1_g025.nc"   "0.000183376508520238"
 [39,] "tas_mon_IPSL-CM5A-LR_rcp45_r4i1p1_g025.nc"   "0.000223893317401282"
 [40,] "tas_mon_IPSL-CM5A-LR_rcp45_r1i1p1_g025.nc"   "0.000227870814837588"
 [41,] "tas_mon_MIROC-ESM-CHEM_rcp45_r1i1p1_g025.nc" "0.000230632620843341"
 [42,] "tas_mon_IPSL-CM5A-LR_rcp45_r3i1p1_g025.nc"   "0.000240921050097993"
 [43,] "tas_mon_IPSL-CM5A-LR_rcp45_r2i1p1_g025.nc"   "0.000259492885366603"
 [44,] "tas_mon_GFDL-ESM2G_rcp45_r1i1p1_g025.nc"     "0.000273509916846808"
 [45,] "tas_mon_MIROC-ESM_rcp45_r1i1p1_g025.nc"      "0.00027611453641935"
 [46,] "tas_mon_IPSL-CM5A-MR_rcp45_r1i1p1_g025.nc"   "0.000297637125704603"
 [47,] "tas_mon_GFDL-ESM2M_rcp45_r1i1p1_g025.nc"     "0.000346247708013947"
 [48,] "tas_mon_inmcm4_rcp45_r1i1p1_g025.nc"         "0.000348999068918173"
 [49,] "tas_mon_bcc-csm1-1-m_rcp45_r1i1p1_g025.nc"   "0.000381075594932036"
 [50,] "tas_mon_ACCESS1-3_rcp45_r1i1p1_g025.nc"      "0.000438119343432566"
 [51,] "tas_mon_EC-EARTH_rcp45_r2i1p1_g025.nc"       "0.000540317695014142"
 [52,] "tas_mon_EC-EARTH_rcp45_r12i1p1_g025.nc"      "0.000574333688226816"
 [53,] "tas_mon_HadGEM2-ES_rcp45_r1i1p1_g025.nc"     "0.00059999404594843"
 [54,] "tas_mon_HadGEM2-ES_rcp45_r3i1p1_g025.nc"     "0.000627811799681993"
 [55,] "tas_mon_EC-EARTH_rcp45_r1i1p1_g025.nc"       "0.000636201487028358"
 [56,] "tas_mon_HadGEM2-AO_rcp45_r1i1p1_g025.nc"     "0.000658288917001936"
 [57,] "tas_mon_HadGEM2-CC_rcp45_r1i1p1_g025.nc"     "0.000664418050437096"
```

```
 [58,] "tas_mon_HadGEM2-ES_rcp45_r2i1p1_g025.nc"     "0.000668908191190323"
 [59,] "tas_mon_EC-EARTH_rcp45_r9i1p1_g025.nc"       "0.00071579595402898"
 [60,] "tas_mon_EC-EARTH_rcp45_r14i1p1_g025.nc"      "0.000728271205625879"
 [61,] "tas_mon_CCSM4_rcp45_r4i1p1_g025.nc"          "0.000734017562592531"
 [62,] "tas_mon_ACCESS1-0_rcp45_r1i1p1_g025.nc"      "0.000734453605063471"
 [63,] "tas_mon_CESM1-BGC_rcp45_r1i1p1_g025.nc"      "0.000778896932121128"
 [64,] "tas_mon_CNRM-CM5_rcp45_r1i1p1_g025.nc"       "0.000779284363668748"
 [65,] "tas_mon_CCSM4_rcp45_r1i1p1_g025.nc"          "0.000791581824222519"
 [66,] "tas_mon_HadGEM2-ES_rcp45_r4i1p1_g025.nc"     "0.000819995997731008"
 [67,] "tas_mon_EC-EARTH_rcp45_r8i1p1_g025.nc"       "0.000824069553137642"
 [68,] "tas_mon_CCSM4_rcp45_r5i1p1_g025.nc"          "0.000829044313245045"
 [69,] "tas_mon_CESM1-CAM5_rcp45_r2i1p1_g025.nc"     "0.000855046415600886"
 [70,] "tas_mon_CCSM4_rcp45_r6i1p1_g025.nc"          "0.000873529789192935"
 [71,] "tas_mon_CESM1-CAM5_rcp45_r3i1p1_g025.nc"     "0.000893730717560739"
 [72,] "tas_mon_CCSM4_rcp45_r2i1p1_g025.nc"          "0.000898718340810694"
 [73,] "tas_mon_CCSM4_rcp45_r3i1p1_g025.nc"          "0.000916731536264678"
 [74,] "tas_mon_MRI-CGCM3_rcp45_r1i1p1_g025.nc"      "0.000936073028757775"
 [75,] "tas_mon_CESM1-CAM5_rcp45_r1i1p1_g025.nc"     "0.000980026851204144"
 [76,] "tas_mon_GISS-E2-R_rcp45_r5i1p1_g025.nc"      "0.00119600719451149"
 [77,] "tas_mon_GISS-E2-R-CC_rcp45_r1i1p1_g025.nc"   "0.00127065681629919"
 [78,] "tas_mon_GISS-E2-R_rcp45_r3i1p3_g025.nc"      "0.0012949535483509"
 [79,] "tas_mon_GISS-E2-R_rcp45_r1i1p1_g025.nc"      "0.00141135543910013"
 [80,] "tas_mon_GISS-E2-H_rcp45_r3i1p3_g025.nc"      "0.00143527176429197"
 [81,] "tas_mon_GISS-E2-H_rcp45_r4i1p3_g025.nc"      "0.00144943543990124"
 [82,] "tas_mon_GISS-E2-R_rcp45_r5i1p2_g025.nc"      "0.00146647260446157"
 [83,] "tas_mon_GISS-E2-R_rcp45_r4i1p3_g025.nc"      "0.00148330279089228"
 [84,] "tas_mon_GISS-E2-R_rcp45_r4i1p1_g025.nc"      "0.00153204113252216"
 [85,] "tas_mon_GISS-E2-R_rcp45_r6i1p3_g025.nc"      "0.0015600708426182"
 [86,] "tas_mon_GISS-E2-H_rcp45_r3i1p1_g025.nc"      "0.00157546282803992"
 [87,] "tas_mon_GISS-E2-H_rcp45_r3i1p2_g025.nc"      "0.00157593464545172"
 [88,] "tas_mon_GISS-E2-R_rcp45_r1i1p2_g025.nc"      "0.00159523610557242"
 [89,] "tas_mon_GISS-E2-H_rcp45_r1i1p1_g025.nc"      "0.00162466496899046"
 [90,] "tas_mon_GISS-E2-R_rcp45_r3i1p2_g025.nc"      "0.00162868299940382"
 [91,] "tas_mon_GISS-E2-H_rcp45_r5i1p3_g025.nc"      "0.00163677341023116"
 [92,] "tas_mon_GISS-E2-R_rcp45_r2i1p2_g025.nc"      "0.00164012658078277"
 [93,] "tas_mon_GISS-E2-H_rcp45_r2i1p1_g025.nc"      "0.00164461144668598"
 [94,] "tas_mon_GISS-E2-H-CC_rcp45_r1i1p1_g025.nc"   "0.00165192162129867"
 [95,] "tas_mon_GISS-E2-R_rcp45_r2i1p1_g025.nc"      "0.00165193074231649"
 [96,] "tas_mon_GISS-E2-H_rcp45_r4i1p2_g025.nc"      "0.00165669599523625"
 [97,] "tas_mon_GISS-E2-H_rcp45_r1i1p2_g025.nc"      "0.00165852916413023"
 [98,] "tas_mon_GISS-E2-H_rcp45_r1i1p3_g025.nc"      "0.00166798140170572"
 [99,] "tas_mon_GISS-E2-R_rcp45_r5i1p3_g025.nc"      "0.00171053855901016"
[100,] "tas_mon_GISS-E2-H_rcp45_r2i1p2_g025.nc"      "0.00171548163471827"
[101,] "tas_mon_GISS-E2-R_rcp45_r1i1p3_g025.nc"      "0.00173972508449444"
[102,] "tas_mon_GISS-E2-H_rcp45_r5i1p2_g025.nc"      "0.00174753604222548"
[103,] "tas_mon_GISS-E2-R_rcp45_r2i1p3_g025.nc"      "0.00175980502931455"
[104,] "tas_mon_GISS-E2-H_rcp45_r4i1p1_g025.nc"      "0.00184457351420222"
[105,] "tas_mon_GISS-E2-H_rcp45_r2i1p3_g025.nc"      "0.00188735119085388"
[106,] "tas_mon_GISS-E2-H_rcp45_r5i1p1_g025.nc"      "0.00188814929359308"
[107,] "tas_mon_GISS-E2-R_rcp45_r3i1p1_g025.nc"      "0.00190870937729969"
[108,] "tas_mon_GISS-E2-R_rcp45_r6i1p1_g025.nc"      "0.00192168723825333"
[109,] "tas_mon_GISS-E2-R_rcp45_r4i1p2_g025.nc"      "0.00202220248937382"
[110,] "tas_mon_CMCC-CM_rcp45_r1i1p1_g025.nc"        "0.00448169866503329"
```

The estimates for σ of the monthly Near Surface Temperature Data under the RCP45 at time $t = 100$ are:

```
> sigmaTas
       file                                          sigma
  [1,] "tas_mon_FIO-ESM_rcp45_r3i1p1_g025.nc"        "0.11006772222915"
  [2,] "tas_mon_FIO-ESM_rcp45_r1i1p1_g025.nc"        "0.11366495539389"
  [3,] "tas_mon_FIO-ESM_rcp45_r2i1p1_g025.nc"        "0.115593458668153"
  [4,] "tas_mon_BNU-ESM_rcp45_r1i1p1_g025.nc"        "0.122867289754893"
  [5,] "tas_mon_FGOALS-g2_rcp45_r1i1p1_g025.nc"      "0.142121953939162"
  [6,] "tas_mon_CSIRO-Mk3L-1-2_rcp45_r1i2p1_g025.nc" "0.152685116458957"
  [7,] "tas_mon_CSIRO-Mk3L-1-2_rcp45_r3i2p1_g025.nc" "0.153301490720799"
  [8,] "tas_mon_CSIRO-Mk3L-1-2_rcp45_r2i2p1_g025.nc" "0.157096157213068"
  [9,] "tas_mon_bcc-csm1-1_rcp45_r1i1p1_g025.nc"     "0.163111081940897"
 [10,] "tas_mon_MIROC-ESM_rcp45_r1i1p1_g025.nc"      "0.167913588714449"
 [11,] "tas_mon_MIROC-ESM-CHEM_rcp45_r1i1p1_g025.nc" "0.172819107874678"
 [12,] "tas_mon_NorESM1-ME_rcp45_r1i1p1_g025.nc"     "0.234056905841689"
 [13,] "tas_mon_NorESM1-M_rcp45_r1i1p1_g025.nc"      "0.23566621221152"
```

```
[14,] "tas_mon_CanESM2_rcp45_r1i1p1_g025.nc"         "0.251752880924171"
[15,] "tas_mon_CanESM2_rcp45_r4i1p1_g025.nc"         "0.252622776640708"
[16,] "tas_mon_CanESM2_rcp45_r5i1p1_g025.nc"         "0.254188764207865"
[17,] "tas_mon_CanESM2_rcp45_r3i1p1_g025.nc"         "0.261787327080742"
[18,] "tas_mon_CanESM2_rcp45_r2i1p1_g025.nc"         "0.264976452335187"
[19,] "tas_mon_GFDL-ESM2G_rcp45_r1i1p1_g025.nc"      "0.291977984931963"
[20,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r8i1p1_g025.nc"   "0.294447453821219"
[21,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r1i1p1_g025.nc"   "0.297226239464511"
[22,] "tas_mon_GFDL-ESM2M_rcp45_r1i1p1_g025.nc"      "0.297708935368126"
[23,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r7i1p1_g025.nc"   "0.298687134004281"
[24,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r5i1p1_g025.nc"   "0.299927626958636"
[25,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r10i1p1_g025.nc"  "0.302081098421895"
[26,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r6i1p1_g025.nc"   "0.304634527443095"
[27,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r3i1p1_g025.nc"   "0.304903070315307"
[28,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r2i1p1_g025.nc"   "0.305151041470755"
[29,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r4i1p1_g025.nc"   "0.306184408792047"
[30,] "tas_mon_GFDL-CM3_rcp45_r1i1p1_g025.nc"        "0.314536211248328"
[31,] "tas_mon_CSIRO-Mk3-6-0_rcp45_r9i1p1_g025.nc"   "0.314720403230571"
[32,] "tas_mon_MPI-ESM-MR_rcp45_r1i1p1_g025.nc"      "0.338105838691436"
[33,] "tas_mon_MPI-ESM-LR_rcp45_r3i1p1_g025.nc"      "0.340028664575745"
[34,] "tas_mon_IPSL-CM5A-LR_rcp45_r1i1p1_g025.nc"    "0.340817048569884"
[35,] "tas_mon_MIROC5_rcp45_r1i1p1_g025.nc"          "0.341888935722267"
[36,] "tas_mon_IPSL-CM5A-LR_rcp45_r3i1p1_g025.nc"    "0.342196192015396"
[37,] "tas_mon_CMCC-CMS_rcp45_r1i1p1_g025.nc"        "0.343687130263206"
[38,] "tas_mon_IPSL-CM5A-LR_rcp45_r4i1p1_g025.nc"    "0.343815529312028"
[39,] "tas_mon_MPI-ESM-MR_rcp45_r3i1p1_g025.nc"      "0.343821225137654"
[40,] "tas_mon_IPSL-CM5A-LR_rcp45_r2i1p1_g025.nc"    "0.343966987075992"
[41,] "tas_mon_MIROC5_rcp45_r3i1p1_g025.nc"          "0.347318040007292"
[42,] "tas_mon_MPI-ESM-MR_rcp45_r2i1p1_g025.nc"      "0.348028264016044"
[43,] "tas_mon_MPI-ESM-LR_rcp45_r1i1p1_g025.nc"      "0.348770783401351"
[44,] "tas_mon_MIROC5_rcp45_r2i1p1_g025.nc"          "0.34931079805394"
[45,] "tas_mon_MPI-ESM-LR_rcp45_r2i1p1_g025.nc"      "0.351193056857404"
[46,] "tas_mon_bcc-csm1-1-m_rcp45_r1i1p1_g025.nc"    "0.365927677210095"
[47,] "tas_mon_IPSL-CM5B-LR_rcp45_r1i1p1_g025.nc"    "0.386942398438445"
[48,] "tas_mon_IPSL-CM5A-MR_rcp45_r1i1p1_g025.nc"    "0.393727243698931"
[49,] "tas_mon_inmcm4_rcp45_r1i1p1_g025.nc"          "0.406750316901783"
[50,] "tas_mon_ACCESS1-3_rcp45_r1i1p1_g025.nc"       "0.452497224520464"
[51,] "tas_mon_HadGEM2-ES_rcp45_r3i1p1_g025.nc"      "0.453228532394665"
[52,] "tas_mon_HadGEM2-ES_rcp45_r1i1p1_g025.nc"      "0.453439735786529"
[53,] "tas_mon_HadGEM2-ES_rcp45_r4i1p1_g025.nc"      "0.458558720518478"
[54,] "tas_mon_HadGEM2-AO_rcp45_r1i1p1_g025.nc"      "0.461717332050332"
[55,] "tas_mon_EC-EARTH_rcp45_r12i1p1_g025.nc"       "0.465556936460681"
[56,] "tas_mon_CCSM4_rcp45_r1i1p1_g025.nc"           "0.468621573726459"
[57,] "tas_mon_CCSM4_rcp45_r6i1p1_g025.nc"           "0.468753491380023"
[58,] "tas_mon_CCSM4_rcp45_r4i1p1_g025.nc"           "0.471318802085698"
[59,] "tas_mon_HadGEM2-ES_rcp45_r2i1p1_g025.nc"      "0.471678405706982"
[60,] "tas_mon_CCSM4_rcp45_r3i1p1_g025.nc"           "0.473422104045817"
[61,] "tas_mon_HadGEM2-CC_rcp45_r1i1p1_g025.nc"      "0.474969251670787"
[62,] "tas_mon_EC-EARTH_rcp45_r1i1p1_g025.nc"        "0.477035700958721"
[63,] "tas_mon_CCSM4_rcp45_r2i1p1_g025.nc"           "0.478438028699955"
[64,] "tas_mon_CESM1-BGC_rcp45_r1i1p1_g025.nc"       "0.480546247502464"
[65,] "tas_mon_MRI-CGCM3_rcp45_r1i1p1_g025.nc"       "0.483280195484371"
[66,] "tas_mon_EC-EARTH_rcp45_r2i1p1_g025.nc"        "0.483417250596197"
[67,] "tas_mon_EC-EARTH_rcp45_r8i1p1_g025.nc"        "0.483853312905823"
[68,] "tas_mon_CCSM4_rcp45_r5i1p1_g025.nc"           "0.486160991087662"
[69,] "tas_mon_EC-EARTH_rcp45_r14i1p1_g025.nc"       "0.490400475646624"
[70,] "tas_mon_EC-EARTH_rcp45_r9i1p1_g025.nc"        "0.491934406261398"
[71,] "tas_mon_ACCESS1-0_rcp45_r1i1p1_g025.nc"       "0.493605976628916"
[72,] "tas_mon_CESM1-CAM5_rcp45_r1i1p1_g025.nc"      "0.509102704695612"
[73,] "tas_mon_CESM1-CAM5_rcp45_r3i1p1_g025.nc"      "0.520239388917884"
[74,] "tas_mon_CESM1-CAM5_rcp45_r2i1p1_g025.nc"      "0.522204370025365"
[75,] "tas_mon_GISS-E2-R_rcp45_r3i1p2_g025.nc"       "0.526191911252957"
[76,] "tas_mon_GISS-E2-R_rcp45_r2i1p1_g025.nc"       "0.528768288727814"
[77,] "tas_mon_GISS-E2-R_rcp45_r2i1p3_g025.nc"       "0.528993742931682"
[78,] "tas_mon_GISS-E2-R_rcp45_r6i1p3_g025.nc"       "0.533866435910881"
[79,] "tas_mon_GISS-E2-R_rcp45_r4i1p1_g025.nc"       "0.535774646417207"
[80,] "tas_mon_GISS-E2-R_rcp45_r5i1p3_g025.nc"       "0.536739449040725"
[81,] "tas_mon_GISS-E2-R_rcp45_r5i1p1_g025.nc"       "0.537064869779151"
[82,] "tas_mon_GISS-E2-R_rcp45_r3i1p3_g025.nc"       "0.537424161184229"
[83,] "tas_mon_GISS-E2-R_rcp45_r1i1p2_g025.nc"       "0.537837714474258"
[84,] "tas_mon_GISS-E2-R_rcp45_r1i1p1_g025.nc"       "0.538853936738275"
[85,] "tas_mon_GISS-E2-R_rcp45_r4i1p2_g025.nc"       "0.539707249081328"
[86,] "tas_mon_GISS-E2-R_rcp45_r4i1p3_g025.nc"       "0.541019809740245"
```

```
 [87,] "tas_mon_GISS-E2-R_rcp45_r1i1p3_g025.nc"      "0.542739000540321"
 [88,] "tas_mon_GISS-E2-R-CC_rcp45_r1i1p1_g025.nc"   "0.54804664865616"
 [89,] "tas_mon_GISS-E2-R_rcp45_r3i1p1_g025.nc"      "0.548228996701106"
 [90,] "tas_mon_GISS-E2-R_rcp45_r2i1p2_g025.nc"      "0.548603283093651"
 [91,] "tas_mon_GISS-E2-R_rcp45_r5i1p2_g025.nc"      "0.551277895418193"
 [92,] "tas_mon_GISS-E2-R_rcp45_r6i1p1_g025.nc"      "0.553503449863243"
 [93,] "tas_mon_GISS-E2-H_rcp45_r4i1p2_g025.nc"      "0.554553133320771"
 [94,] "tas_mon_GISS-E2-H_rcp45_r1i1p3_g025.nc"      "0.557070985981428"
 [95,] "tas_mon_GISS-E2-H_rcp45_r4i1p3_g025.nc"      "0.557396708093455"
 [96,] "tas_mon_GISS-E2-H_rcp45_r5i1p1_g025.nc"      "0.559554115949138"
 [97,] "tas_mon_GISS-E2-H_rcp45_r4i1p1_g025.nc"      "0.561627600284377"
 [98,] "tas_mon_GISS-E2-H_rcp45_r2i1p1_g025.nc"      "0.562381313824867"
 [99,] "tas_mon_GISS-E2-H_rcp45_r5i1p2_g025.nc"      "0.56242937369888"
[100,] "tas_mon_GISS-E2-H_rcp45_r1i1p2_g025.nc"      "0.563763052606992"
[101,] "tas_mon_GISS-E2-H_rcp45_r3i1p3_g025.nc"      "0.565812800350639"
[102,] "tas_mon_CNRM-CM5_rcp45_r1i1p1_g025.nc"       "0.567894576297148"
[103,] "tas_mon_GISS-E2-H_rcp45_r3i1p2_g025.nc"      "0.568096898758881"
[104,] "tas_mon_GISS-E2-H_rcp45_r2i1p3_g025.nc"      "0.572225686262292"
[105,] "tas_mon_GISS-E2-H_rcp45_r5i1p3_g025.nc"      "0.572325813631521"
[106,] "tas_mon_GISS-E2-H-CC_rcp45_r1i1p1_g025.nc"   "0.57613901948118"
[107,] "tas_mon_GISS-E2-H_rcp45_r2i1p2_g025.nc"      "0.577021593947984"
[108,] "tas_mon_GISS-E2-H_rcp45_r3i1p1_g025.nc"      "0.578371145557919"
[109,] "tas_mon_GISS-E2-H_rcp45_r1i1p1_g025.nc"      "0.579337834948866"
[110,] "tas_mon_CMCC-CM_rcp45_r1i1p1_g025.nc"        "0.681343428572589"
```

The estimates for σ of the monthly Surface Upwelling Longwave Radiation under the RCP45 at time $t = 100$ are:

```
> sigmaRlus
       file                                          sigma
  [1,] "rlus_mon_bcc-csm1-1_rcp45_r1i1p1_g025.nc"     "1.12257069943875"
  [2,] "rlus_mon_FGOALS-g2_rcp45_r1i1p1_g025.nc"      "1.17189204415031"
  [3,] "rlus_mon_BNU-ESM_rcp45_r1i1p1_g025.nc"        "1.22784408801046"
  [4,] "rlus_mon_CSIRO-Mk3L-1-2_rcp45_r3i2p1_g025.nc" "1.34032649858499"
  [5,] "rlus_mon_MIROC-ESM_rcp45_r1i1p1_g025.nc"      "1.34964408479685"
  [6,] "rlus_mon_CSIRO-Mk3L-1-2_rcp45_r1i2p1_g025.nc" "1.35192757013706"
  [7,] "rlus_mon_MIROC-ESM-CHEM_rcp45_r1i1p1_g025.nc" "1.35490311136217"
  [8,] "rlus_mon_CSIRO-Mk3L-1-2_rcp45_r2i2p1_g025.nc" "1.35559110276213"
  [9,] "rlus_mon_NorESM1-ME_rcp45_r1i1p1_g025.nc"     "1.80707457264746"
 [10,] "rlus_mon_NorESM1-M_rcp45_r1i1p1_g025.nc"      "1.86287700292733"
 [11,] "rlus_mon_CanESM2_rcp45_r1i1p1_g025.nc"        "2.0739086568381"
 [12,] "rlus_mon_CanESM2_rcp45_r4i1p1_g025.nc"        "2.08397987437844"
 [13,] "rlus_mon_CanESM2_rcp45_r5i1p1_g025.nc"        "2.0854431278767"
 [14,] "rlus_mon_CanESM2_rcp45_r3i1p1_g025.nc"        "2.12078052993802"
 [15,] "rlus_mon_CanESM2_rcp45_r2i1p1_g025.nc"        "2.15569826809795"
 [16,] "rlus_mon_IPSL-CM5A-LR_rcp45_r1i1p1_g025.nc"   "2.22175951228836"
 [17,] "rlus_mon_IPSL-CM5A-LR_rcp45_r3i1p1_g025.nc"   "2.2319039415424"
 [18,] "rlus_mon_IPSL-CM5A-LR_rcp45_r2i1p1_g025.nc"   "2.23865455374341"
 [19,] "rlus_mon_IPSL-CM5A-LR_rcp45_r4i1p1_g025.nc"   "2.24032342702726"
 [20,] "rlus_mon_CSIRO-Mk3-6-0_rcp45_r8i1p1_g025.nc"  "2.24994081628824"
 [21,] "rlus_mon_CSIRO-Mk3-6-0_rcp45_r1i1p1_g025.nc"  "2.26589971000066"
 [22,] "rlus_mon_CSIRO-Mk3-6-0_rcp45_r2i1p1_g025.nc"  "2.31641031306482"
 [23,] "rlus_mon_CSIRO-Mk3-6-0_rcp45_r10i1p1_g025.nc" "2.32748654815469"
 [24,] "rlus_mon_CSIRO-Mk3-6-0_rcp45_r7i1p1_g025.nc"  "2.32847049667178"
 [25,] "rlus_mon_CSIRO-Mk3-6-0_rcp45_r4i1p1_g025.nc"  "2.33207596789444"
 [26,] "rlus_mon_CSIRO-Mk3-6-0_rcp45_r5i1p1_g025.nc"  "2.33502320886262"
 [27,] "rlus_mon_CSIRO-Mk3-6-0_rcp45_r3i1p1_g025.nc"  "2.34151062542213"
 [28,] "rlus_mon_CSIRO-Mk3-6-0_rcp45_r6i1p1_g025.nc"  "2.35467105785737"
 [29,] "rlus_mon_IPSL-CM5B-LR_rcp45_r1i1p1_g025.nc"   "2.39813010152707"
 [30,] "rlus_mon_CSIRO-Mk3-6-0_rcp45_r9i1p1_g025.nc"  "2.45412672328327"
 [31,] "rlus_mon_GFDL-CM3_rcp45_r1i1p1_g025.nc"       "2.45630826834349"
 [32,] "rlus_mon_bcc-csm1-1-m_rcp45_r1i1p1_g025.nc"   "2.52423901280235"
 [33,] "rlus_mon_IPSL-CM5A-MR_rcp45_r1i1p1_g025.nc"   "2.57747636486176"
 [34,] "rlus_mon_MIROC5_rcp45_r3i1p1_g025.nc"         "2.59042457827"
 [35,] "rlus_mon_MIROC5_rcp45_r1i1p1_g025.nc"         "2.60811914714179"
 [36,] "rlus_mon_MIROC5_rcp45_r2i1p1_g025.nc"         "2.6086139612406"
 [37,] "rlus_mon_MPI-ESM-MR_rcp45_r1i1p1_g025.nc"     "2.6232077751079"
 [38,] "rlus_mon_GFDL-ESM2G_rcp45_r1i1p1_g025.nc"     "2.62507285076436"
 [39,] "rlus_mon_MPI-ESM-MR_rcp45_r3i1p1_g025.nc"     "2.67820731486106"
 [40,] "rlus_mon_MPI-ESM-LR_rcp45_r3i1p1_g025.nc"     "2.68954353629021"
 [41,] "rlus_mon_MPI-ESM-LR_rcp45_r1i1p1_g025.nc"     "2.73386354344992"
 [42,] "rlus_mon_GFDL-ESM2M_rcp45_r1i1p1_g025.nc"     "2.73761230712462"
```

```
 [43,] "rlus_mon_MPI-ESM-LR_rcp45_r2i1p1_g025.nc"     "2.74815747984698"
 [44,] "rlus_mon_MPI-ESM-MR_rcp45_r2i1p1_g025.nc"     "2.77273400204008"
 [45,] "rlus_mon_inmcm4_rcp45_r1i1p1_g025.nc"         "2.81690207721943"
 [46,] "rlus_mon_CMCC-CMS_rcp45_r1i1p1_g025.nc"       "2.8456462206077"
 [47,] "rlus_mon_HadGEM2-ES_rcp45_r4i1p1_g025.nc"     "3.09779916956948"
 [48,] "rlus_mon_ACCESS1-0_rcp45_r1i1p1_g025.nc"      "3.10307763824093"
 [49,] "rlus_mon_HadGEM2-ES_rcp45_r1i1p1_g025.nc"     "3.11817679473343"
 [50,] "rlus_mon_HadGEM2-ES_rcp45_r3i1p1_g025.nc"     "3.14044410779061"
 [51,] "rlus_mon_ACCESS1-3_rcp45_r1i1p1_g025.nc"      "3.19398896144411"
 [52,] "rlus_mon_CCSM4_rcp45_r4i1p1_g025.nc"          "3.23759283083161"
 [53,] "rlus_mon_HadGEM2-CC_rcp45_r1i1p1_g025.nc"     "3.24610393368884"
 [54,] "rlus_mon_CCSM4_rcp45_r6i1p1_g025.nc"          "3.24637765524711"
 [55,] "rlus_mon_HadGEM2-ES_rcp45_r2i1p1_g025.nc"     "3.25023597135025"
 [56,] "rlus_mon_GISS-E2-R_rcp45_r2i1p3_g025.nc"      "3.28033177633024"
 [57,] "rlus_mon_GISS-E2-R_rcp45_r3i1p2_g025.nc"      "3.28880204289186"
 [58,] "rlus_mon_GISS-E2-R_rcp45_r2i1p1_g025.nc"      "3.31183316691269"
 [59,] "rlus_mon_CCSM4_rcp45_r1i1p1_g025.nc"          "3.31210080452402"
 [60,] "rlus_mon_GISS-E2-R_rcp45_r4i1p2_g025.nc"      "3.31916797030403"
 [61,] "rlus_mon_GISS-E2-H-CC_rcp45_r1i1p1_g025.nc"   "3.32032008838395"
 [62,] "rlus_mon_GISS-E2-R_rcp45_r1i1p2_g025.nc"      "3.32476971671713"
 [63,] "rlus_mon_GISS-E2-R-CC_rcp45_r1i1p1_g025.nc"   "3.33199814349713"
 [64,] "rlus_mon_GISS-E2-R_rcp45_r3i1p3_g025.nc"      "3.33513038619489"
 [65,] "rlus_mon_GISS-E2-H_rcp45_r5i1p1_g025.nc"      "3.33659185633837"
 [66,] "rlus_mon_MRI-CGCM3_rcp45_r1i1p1_g025.nc"      "3.33846229841964"
 [67,] "rlus_mon_GISS-E2-H_rcp45_r4i1p2_g025.nc"      "3.34065807527213"
 [68,] "rlus_mon_CCSM4_rcp45_r3i1p1_g025.nc"          "3.34210424107114"
 [69,] "rlus_mon_GISS-E2-R_rcp45_r4i1p1_g025.nc"      "3.34241482990196"
 [70,] "rlus_mon_GISS-E2-R_rcp45_r1i1p1_g025.nc"      "3.34468799759035"
 [71,] "rlus_mon_GISS-E2-H_rcp45_r5i1p2_g025.nc"      "3.34520451140043"
 [72,] "rlus_mon_GISS-E2-R_rcp45_r1i1p3_g025.nc"      "3.34832373286322"
 [73,] "rlus_mon_GISS-E2-H_rcp45_r2i1p1_g025.nc"      "3.35101907772418"
 [74,] "rlus_mon_GISS-E2-H_rcp45_r4i1p3_g025.nc"      "3.35368073389759"
 [75,] "rlus_mon_GISS-E2-R_rcp45_r5i1p1_g025.nc"      "3.36043182823501"
 [76,] "rlus_mon_CESM1-BGC_rcp45_r1i1p1_g025.nc"      "3.36168684968168"
 [77,] "rlus_mon_GISS-E2-H_rcp45_r4i1p1_g025.nc"      "3.36170315761098"
 [78,] "rlus_mon_CCSM4_rcp45_r5i1p1_g025.nc"          "3.36573640690487"
 [79,] "rlus_mon_CCSM4_rcp45_r2i1p1_g025.nc"          "3.36619351581275"
 [80,] "rlus_mon_GISS-E2-R_rcp45_r5i1p2_g025.nc"      "3.36683216953893"
 [81,] "rlus_mon_GISS-E2-H_rcp45_r5i1p3_g025.nc"      "3.36701823641309"
 [82,] "rlus_mon_GISS-E2-H_rcp45_r2i1p3_g025.nc"      "3.37157233040796"
 [83,] "rlus_mon_GISS-E2-H_rcp45_r1i1p3_g025.nc"      "3.37318975838176"
 [84,] "rlus_mon_GISS-E2-H_rcp45_r3i1p1_g025.nc"      "3.37777535074578"
 [85,] "rlus_mon_GISS-E2-R_rcp45_r5i1p3_g025.nc"      "3.37993144649907"
 [86,] "rlus_mon_GISS-E2-H_rcp45_r1i1p2_g025.nc"      "3.38006339550918"
 [87,] "rlus_mon_GISS-E2-R_rcp45_r4i1p3_g025.nc"      "3.3841544934816"
 [88,] "rlus_mon_GISS-E2-H_rcp45_r3i1p3_g025.nc"      "3.3847171010753"
 [89,] "rlus_mon_GISS-E2-R_rcp45_r3i1p1_g025.nc"      "3.38722037434004"
 [90,] "rlus_mon_GISS-E2-R_rcp45_r2i1p2_g025.nc"      "3.39810755017122"
 [91,] "rlus_mon_GISS-E2-H_rcp45_r1i1p1_g025.nc"      "3.40121724617596"
 [92,] "rlus_mon_GISS-E2-R_rcp45_r6i1p3_g025.nc"      "3.4046325011337"
 [93,] "rlus_mon_GISS-E2-R_rcp45_r6i1p1_g025.nc"      "3.4063820054556"
 [94,] "rlus_mon_GISS-E2-H_rcp45_r3i1p2_g025.nc"      "3.41856638908484"
 [95,] "rlus_mon_GISS-E2-H_rcp45_r2i1p2_g025.nc"      "3.44570652269342"
 [96,] "rlus_mon_CESM1-CAM5_rcp45_r1i1p1_g025.nc"     "3.53904229858017"
 [97,] "rlus_mon_CESM1-CAM5_rcp45_r3i1p1_g025.nc"     "3.63363356344099"
 [98,] "rlus_mon_CESM1-CAM5_rcp45_r2i1p1_g025.nc"     "3.65789195162577"
 [99,] "rlus_mon_CNRM-CM5_rcp45_r1i1p1_g025.nc"       "3.82696568740668"
[100,] "rlus_mon_CMCC-CM_rcp45_r1i1p1_g025.nc"        "4.68354930141825
```

8.3 Source code

The R code in this thesis has been written with Rstudio V0.99.489 for Linux. It would be beyond the scope of this written report to include all the R scripts that have been written. Nevertheless, one can find some extracts of the most important R scripts that are used throughout this thesis below.

8.3.1 Preanalysis of data source code

```
#standardTest input: path to NetCDF file
#standardTest output:
#name: NetCDF file name
#varname:     climate variable (see section on climate variables)
#type:        'land', 'sea' or 'global'
#             (depending on where the variable can be measured)
#             E.g., tos, i.e., sea surface temperature can only be
#             measured at regions of sea.
#missing:     TRUE/FALSE, if TRUE -->there are missing values
#                     if FALSE--> no missing values
#numbOfNA:    number of missing values
#ratioNA:     ratio of missing values, i.e.,
#             ratioNA=(number of missing values)/(144*72*timeDim)
#             144*72*timeDim corresponds to the total number of values
#             that can be assigned for a 2.5x2.5 degree pixel.
#missComment: "ok"/"suspicious", depending on the ratioNA.
#             "suspicious" if it is higher than a set threshold
#             thresholds: 40% for sea type, 80% for land type variables
#sgn:         sign of the climate variable values
#totmax:      maximum of the climate variable values
#totmin:      minimum of the climate variable values
#average:     arithmetic mean of the climate variable values
#std:         standard deviation of the climate variable values
#timeDim:     number of time units (months, years, seasons) that are modeled
#range:       "ok"/"suspicious"
#             "suspicious": variable values are higher or lower then
#             predefined bounds

#Author: Carina Schneider (2016)

standardTest <- function(path){
  name <- basename(path)
  varname <- read.table(text=name,sep="_", as.is=T)$V1
  nc <- open.ncdf(path)
  data <- get.var.ncdf(nc)
  close.ncdf(nc)
  timeDim <- dim(data)[3]
  totmax <- max(data,na.rm=TRUE)
  totmin <- min(data,na.rm=TRUE)
  average <- mean(data,na.rm=TRUE)
  std <- sd(data,na.rm=TRUE)

    range <- "ok"
    bounds <- numeric(2)
    datframe <- rangeCheck(varname)
    bounds[1] <- datframe$lbound
    bounds[2] <- datframe$ubound
    type <- datframe$vartype

    if((totmax>bounds[2])|(totmin<bounds[1])){
      range <- "suspicious"
    }
  missing <- missingValue(data,type)
  sgn <- sign(bounds)
  df <- data.frame(name,varname,type,missing,sgn,
                totmax,totmin,average,std,timeDim,range)
  return(df)
}
```

```
#Input:
#pathDir: path to a directory with NetCDF files that
#         need to be checked for reasonableness
#alpha:   The level chosen, s.t. 1-alpha is the confidence level
#         of the tolerance intervals
#P:       The proportion of the population to be covered by the
#         tolerance intervals
#All NetCDF files in this directory are then applied to
#the method ''standardTest''
#
#Output:
```

```
#matrix containing information on the NetCDF files of the
#directory

#Author: Carina Schneider (2016)

multipleStanTest <- function(pathDir,alpha,P){
  files <- list.files(path=pathDir,pattern="*.nc",
                      full.names=T,recursive=FALSE)
  baseFilesList <- list.files(path=pathDir,pattern="*.nc",
                      full.names=F,recursive=FALSE)
  var <- scen <- tempRes <- spRes <- character(length=length(baseFilesList))
  for(i in 1:length(baseFilesList)){
    var[i] <- read.table(text=baseFilesList[i],sep="_", as.is=T)$V1
    tempRes[i] <- read.table(text=baseFilesList[i],sep="_", as.is=T)$V2
    scen[i] <- read.table(text=baseFilesList[i],sep="_", as.is=T)$V4
    spRes[i] <- read.table(text=baseFilesList[i],sep="_", as.is=T)$V6
  }

 for(i in 2:length(baseFilesList)){
  if((var[i]!=var[1])|(scen[i]!=scen[1])|
     (tempRes[i]!=tempRes[1])|(spRes[i]!=spRes[i])){
    stop("multipleStanTest only works on a directory of
    CMIP5-ng files with equal scenario,variable and
    temporal/spatial resolution.")
  }
 }

  frames <- lapply(files,standardTest)
  #return(frames)
  n <- length(frames)
  signTable <- getGlobCharNumb(frames,alpha,P)

  numberOfBad <- length(which(signTable[,6]=="susp"))

  cat("There are", numberOfBad, " suspicious files in your directory.")
    return(signTable)
}
```

```
#Input:  Output of the multipleStanTest()
#Output: matrix containing the files which
#        have the most frequently attained time dimension
#it extracts only the files in a directory that have the same time dimension

#Author: Carina Schneider (2016)

extractTimeCoh <- function(out){
  freqSummary <- as.matrix(summary(as.factor(out[,2])))
  maxcount <- max(freqSummary)
  row <- which(as.matrix(freqSummary==maxcount))
  timeDim <- rownames(freqSummary)[row]
  ind <- which(out[,2]==timeDim)
  cat("The most frequent number of time steps is:", timeDim, '\n')
  return(out[ind,c(1,2,6)])
}
```

8.3.2 SNHT source code

Code description is available on CRAN Browning and Schneider [2015].
ATTENTION: The version 1.03 of the SNHT package has a bug. Version 1.04 will be uploaded soon. The code for the version 1.04 is available on github:
https://github.com/rockclimber112358/Stan-Norm-Hom-Test/tree/master/snht

8.3.3 GMRF source code

```
#generates the precision matrix
#mu: mean vector of length lon*lat
#b:  conditional spatial correlation
```

```
#f:  conditional temporal correlation
#c:  scaling factor for kappa
#lon: number of pixels in direction of the longitude
#lat: number of pixels in direction of the latitude
#T:   number of time units
library(spam)

#Author: Carina Schneider (2016)

mgmrf.prec <- function(b,c,f,lon,lat,T){
  n <- lon*lat
  Qsp <- precmat.GMRFreglat(lon,lat,par=b,model="m1p1")
  Qte <- diag.spam(-f,n)
  C <- diag.spam(0,T)
  C[cbind(1:(T-1),2:T)] <- 1
  C[cbind(2:T,1:(T-1))] <- 1
  Q <- kronecker.spam(diag.spam(1,T),Qsp)+kronecker.spam(C,Qte)
  return(Q*c)
}
```

```
#generates GMRF data...
#mu: mean vector
#b:  conditional spatial correlation
#f:  conditional temporal correlation
#c:  scaling factor
#lon: number of pixels in direction of the longitude
#lat: number of pixels in direction of the latitude

#Author: Carina Schneider (2016)

dataGenerator <- function(mu,b,c,f,lon,lat,T){
  Sigmainv <- mgmrf.prec(b,c,f,lon,lat,T)
  Q <- as.spam( Sigmainv, eps=1e-4)
  set.seed(2)
  x <- rmvnorm.prec(1, mu = mu, Q)
  xx <- x
  dim(xx) <- c(lon,lat,T)
  return(xx)
}
```

```
#Input:

#y:     matrix of dimension (time x number of locations)
#mu1:   starting values for the mean: should be a vector of length rown*coln
#rown:  number of longitudes
#coln:  number of latitudes
#f:     starting value for the parameter f
#b:     starting value for the parameter b
#c:     starting value for the parameter c
#m:     number time units (months, years, seasons)

#Output:
#optim object:  containing the neg2loglikelihood value
#               as well as the parameters that minimize it under H0

#Comment: parts of this code have been taken over from
#         Rebekka Schibli's master's thesis code
#Author: Carina Schneider (2016)

test.H0 <- function(y,mu1,b,c,f,rown,coln,m=nrow(y),Rstruct=NULL,...) {
  n <- coln*rown
  spam.options(cholupdatesingular="warning")
  if (!is(Rstruct, "spam.chol.NgPeyton")) {
    Q <- mgmrf.prec(b,c,f,rown,coln,m)
    if (!is.spam(Q))
    stop("'Covariance' should return a spam object.")
    Rstruct <- chol.spam(Q,...)
  }

  neg2loglikelihood <- function(fulltheta,...) {
      resid <- c(t(y)-fulltheta[1:n])
```

```
      Q <- mgmrf.prec(fulltheta[n+1],fulltheta[n+2],fulltheta[n+3], rown,coln,m)

      p=FALSE
      p=tryCatch({
          cholS <- update.spam.chol.NgPeyton(Rstruct,Q,...)
          p=FALSE},

        warning = function(w) {
          p=TRUE
          return(p)}
      )

      if(p==TRUE){
        return(10^6)
      }
      else{
        return(n*m*log(2*pi)-2*(c(determinant.spam.chol.NgPeyton(cholS)$modulus))
        +sum(resid*(Q%*%(resid))))
      }
  }

  return(optim(c(mu1,b,c,f), neg2loglikelihood, method = "L-BFGS-B"
               ,lower=c(rep(-1,n),1e-5,0.1,1e-5),upper=c(rep(1,n),0.29,3.5,0.49),
               control=list(maxit=200)))
}
```

```
#Input:

#y:     matrix of dimension (time x number of locations)
#mu1:   vector of starting values for the mean before the shift
#       (should be of length rown*coln)
#rown:  number of longitudes
#coln:  number of latitudes
#tb:    time at which the neg2loglikelihood (-2H_1) is evaluated
#a:     starting value for the shift height
#f:     starting value for the parameter f
#b:     starting value for the parameter b
#m:     time units

#Output:
#optim object:  containing the neg2loglikelihood value
#               as well as the parameters that minimize
#               it under H1 (having a global shift at tb)

#Comment: parts of this code have been taken over from
#         Rebekka Schibli's master thesis code
#Author: Carina Schneider (2016)

test.H1Glob <- function(y,mu1,a,tb,b,c,f,rown,coln,m=nrow(y), Rstruct=NULL,...) {
  n <- coln*rown
  if (!is(Rstruct, "spam.chol.NgPeyton")) {
    Q <- mgmrf.prec(b,c,f,rown,coln,m)
    if (!is.spam(Q))
    stop("'Covariance' should return a spam object.")
    Rstruct <- chol.spam(Q,...)
  }
  neg2loglikelihood <- function(fulltheta,...) {
      resid1 <- t(y[1:tb,])-fulltheta[1:n]
      mu2 <- fulltheta[1:n]+fulltheta[(n+4)]
      resid2 <- t(y[(tb+1):m,])-mu2
      resid <- c(cbind(resid1,resid2))

      Q <- mgmrf.prec(fulltheta[n+1],fulltheta[n+2],fulltheta[n+3],
                      rown,coln,m)

      p=FALSE
      p=tryCatch({
        cholS <- update.spam.chol.NgPeyton(Rstruct,Q,...)
        p=FALSE},

        warning = function(w) {
```

```
          p=TRUE
          return(p)}
      )

      if(p==TRUE){
        return(10^6)
      }
      else{
        return(n*m*log(2*pi)-2*(c(determinant.spam.chol.NgPeyton(cholS)$modulus))
        +sum(resid*(Q%*%(resid))))
      }
  }
  return(optim(c(mu1,b,c,f,a), neg2loglikelihood, method = "L-BFGS-B",
               lower=c(rep(-1,n),1e-5,0.1,1e-5,-10),upper=c(rep(1,n),0.29,3.5,0.49,10),
               control=list(maxit=200)))
}
```

```
#Input:

#y:     matrix of dimension (time x number of locations)
#mu1:   vector of starting values for the mean before the shift
#       (should be of length rown*coln)
#rown:  number of longitudes
#coln:  number of latitudes
#a:     starting value for the shift height
#f:     starting value for the parameter f
#b:     starting value for the parameter b
#m:     time units
#tb:    time at which the neg2loglikelihood (-2H_1) is evaluated
#l0:    location at which the neg2loglikelihood (-2H_1) is evaluated

#Output:
#optim object:  containing the neg2loglikelihood value
#               as well as the parameters that minimize it under
#               H1 (having a local shift at tb and l0)

#Author: Carina Schneider (2016)

test.H1Loc <- function(y,mu1,a,tb,l0,b,c,f,rown,coln,m=nrow(y), Rstruct=NULL,...) {
  n <- coln*rown
  if (!is(Rstruct, "spam.chol.NgPeyton")) {
    Q <- mgmrf.prec(b,c,f,rown,coln,m)
    if (!is.spam(Q))
    stop("'Covariance' should return a spam object.")
    Rstruct <- chol.spam(Q,...)
  }

  neg2loglikelihood <- function(fulltheta,...) {
      resid1 <- t(y[1:tb,])-fulltheta[1:n]

      mu2 <- fulltheta[1:n]
      mu2[l0] <- mu2[l0]+fulltheta[n+4]
      resid2 <- t(y[(tb+1):m,])-mu2

      resid <- c(cbind(resid1,resid2))

      Q <- mgmrf.prec(fulltheta[n+1],fulltheta[n+2],fulltheta[n+3],
                      rown,coln,m)

      p=FALSE
      p=tryCatch({
        cholS <- update.spam.chol.NgPeyton(Rstruct,Q,...)
        p=FALSE},

        warning = function(w) {
          p=TRUE
          return(p)}
      )

      if(p==TRUE){
        return(10^6)
      }
```

```
      else{
        return(n*m*log(2*pi)-2*(c(determinant.spam.chol.NgPeyton(cholS)$modulus))
               +sum(resid*(Q%*%(resid))))
      }
  }
  return(optim(c(mu1,b,c,f,a), neg2loglikelihood, method = "L-BFGS-B",
               lower= c(rep(-1,n),1e-5,0.1,1e-5,-10),upper=c(rep(1,n),0.29,3.5,0.49,10),
               control=list(maxit=200)))

}
```

```
#This function performs local or global homogeneity detection.
#If an inhomogeneity is found, it prints out the location (if "local"), time and height of the
#most extreme inhomogeneity (i.e. the one with the highest likelihood ratio statistics)

#Input:

#desData:     3 dimensional array (lon x lat x T),
#             i.e., the number of 2.5 pixels in longitude/latitude and number of time units
#type:        "global" or "local"
#             ("global": global shift detection is performed,
#              "local": local shift detection is performed)
#muStart:     optim starting vector for the mean before the potential global/local shift
#             should be of length numbOfLon
#bStart:      optim starting value for parameter b of the spatio-temporal model
#cStart:      optim starting value for parameter c of the spatio-temporal model
#fStart:      optim starting value for parameter f of the spatio-temporal model
#sigLevel:    significance level. Suggested is 0.05. The program takes care of the
#             Bonferroni correction
#L:           indicates how many times the likelihood ratio statistics is evaluated.
#             Default is set to 5, i.e. at every 5-th time.

#Output:

#If "global":
#list containing the vector of likelihood ratio statistics and the vector of
#estimates for the shift heights 'a'
#evaluated at every time

#If "local":
#list containing the matrix of likelihood ratio statistics (dim: N x T) and the estimates for the
#shift heights evaluated at every time at the found location of inhomogeneity
library(spam)

#Author: Carina Schneider (2016)

gmrfHomogeneityTestComp <- function(desData,type,muStart,bStart,cStart,fStart, sigLevel,L=5){

  lon <- dim(desData)[1]
  lat <- dim(desData)[2]
  N <- lon*lat
  T <- dim(desData)[3]
  if(length(muStart)!=N){
    stop("mu does not have the right format. The length
         should equal the number of locations.")
  }

  desDataNew <- desData
  dim(desDataNew) <- c(N,T)

  if(type=="global"){

    MLEH0 <- test.H0(y=t(desDataNew),muStart,bStart,cStart,fStart,lon,lat)
    print(MLEH0)
    if(MLEH0$convergence!=0){
      stop("quasi Newton method did not converge under H0.")
    }
    lRatioStat <- numeric(T)
    a <- numeric(T)
    for(i in 2:(T-3)){
      if(i %% L==0){
        MLEH1 <- test.H1Glob(y=t(desDataNew),muStart,a=0,tb=i,bStart,cStart,fStart,lon,lat)
```

```
      #cat("this is time ", i)
      print(MLEH1)
      if(MLEH1$convergence!=0){
        stop("quasi Newton method did not converge under H1
             (i.e. having a global shift).")
        #this usually happens if not the same bounds, parscale
        # or maxit are chosen in the test.H0() and test.H1() functions
      }
      else{
        lRatioStat[i] <- MLEH0$value-MLEH1$value
        if(lRatioStat[i]<0){
          print(MLEH0)
          print(MLEH1)
          stop("Likelihood ratio statistic <0.")
        }
        a[i] <- MLEH1$par[N+4]
      }
    }
  }
  evTime <- Matrix::nnzero(a)
  th <- qchisq(1-sigLevel/evTime, df=1)
  inhomoFound <- FALSE
  timeOfInhomo <- 0
  heightInhomo <- 0
  if(max(lRatioStat)>th){
    inhomoFound <- TRUE
    timeOfInhomo <- which.max(lRatioStat)
    heightInhomo <- a[timeOfInhomo]
  }
  print(data.frame(inhomoFound,timeOfInhomo,heightInhomo))
  return(data.frame(list(a=a,lRatioStat=lRatioStat)))
  #timeOfInhomo <- which.max(lRatioStat)


}

#---------
else if(type=="local"){

  MLEH0 <- test.H0(y=t(desDataNew),muStart,bStart,cStart,fStart,lon,lat)
  print(MLEH0)
  if(MLEH0$convergence!=0){
    stop("Does not converge under H0")
  }

  a <- matrix(0,T,N)
  lRatioStat <- matrix(0,T,N)
  for(i in 2:(T-2)){
    if(i %% L==0){
    for(k in 1:N){
      MLEH1 <- test.H1Loc(y=t(desDataNew),muStart,a=0,tb=i,
                      l0=k,bStart,cStart,fStart,lon,lat)
      if(MLEH1$convergence!=0){
        stop("Does not converge under H1.")
      }
      a[i,k] <- MLEH1$par[N+4]
      lRatioStat[i,k] <- MLEH0$value-MLEH1$value
    }
    }
  }
  evTime <- Matrix::nnzero(a[,1])
  th <- qchisq(1-sigLevel/evTime, df=1)
  inhomoFound <- FALSE
  timeOfInhomo <- 0
  heightInhomo <- 0
  locInhomo <- 0
  if(max(lRatioStat)>th){
    inhomoFound <- TRUE
    ind <- which.max(lRatioStat)
    timeOfInhomo <- ind %% T
    locInhomo <- ind %/% T +1
    heightInhomo <- a[timeOfInhomo, locInhomo]
  }
```

```
    print(data.frame(inhomoFound,timeOfInhomo,heightInhomo,locInhomo))
    return(list(lRatioStat=lRatioStat,a=a[,locInhomo]))
  }
  else{
    stop("This is not an admissible type.")
  }

}
```

```
#This function can be used if optim does not converge under H0.
#It is a modified version of the original test.H0().
#input, output: see original test.H0()

#Author: Carina Schneider (2016)

test.H0_VR <- function(y,mu1,b,c,f,rown,coln,m=nrow(y),Rstruct=NULL,...) {
  n <- coln*rown

  spam.options(cholupdatesingular="warning")
  if (!is(Rstruct, "spam.chol.NgPeyton")) {
    Q <- mgmrf.prec(b,c,f,rown,coln,m)
    if (!is.spam(Q))
    stop("'Covariance' should return a spam object.")
    Rstruct <- chol.spam(Q,...)
  }


  neg2loglikelihood <- function(fulltheta,...) {
      resid <- c(t(y)-fulltheta[1:n])
      Q <- mgmrf.prec(b,fulltheta[n+1],fulltheta[n+2], rown,coln,m)

      p=FALSE
      p=tryCatch({
          cholS <- update.spam.chol.NgPeyton(Rstruct,Q,...)
          p=FALSE},

        warning = function(w) {
          p=TRUE
          return(p)}
      )

      if(p==TRUE){
        return(10^6)
      }
      else{
        return(n*m*log(2*pi)-2*(c(determinant.spam.chol.NgPeyton(cholS)$modulus))+sum(resid*(Q%*%(resid))))
      }
  }
  upF <- -0.49/0.29*b+0.49-0.02
  return(optim(c(mu1,c,f), neg2loglikelihood, method = "L-BFGS-B"
               ,lower=c(rep(-1,n),1e-5,1e-5),upper=c(rep(1,n),3.5,upF),
               control=list(maxit=200)))

}
```

```
#This function is a modified version of the test.H1Glob() R function.
#It can be used if optim fails to converge
#ATTENTION: Applying this function can lead to a biased likelihood ratio statistics
#           that can increase the probability of committing a type I error.

#Input, output: See the original test.H1Glob()

#Author: Carina Schneider (2016)

test.H1Glob_VR <- function(y,mu1,a,tb,b,c,f,rown,coln,m=nrow(y), Rstruct=NULL,...) {
  n <- coln*rown
  if (!is(Rstruct, "spam.chol.NgPeyton")) {
    Q <- mgmrf.prec(b,c,f,rown,coln,m)
    if (!is.spam(Q))
    stop("'Covariance' should return a spam object.")
    Rstruct <- chol.spam(Q,...)
```

```
  }
    Q <- mgmrf.prec(b,c,f,rown,coln,m)
    cholS <- update.spam.chol.NgPeyton(Rstruct,Q)
    neg2loglikelihood <- function(fulltheta,...) {
    resid1 <- t(y[1:tb,])-fulltheta[1:n]
    mu2 <- fulltheta[1:n]+fulltheta[(n+1)]
    resid2 <- t(y[(tb+1):m,])-mu2
    resid <- c(cbind(resid1,resid2))

    return(n*m*log(2*pi)-2*(c(determinant.spam.chol.NgPeyton(cholS)$modulus))+
    sum(resid*(Q%*%(resid))))
  }
  return(optim(c(mu1,a), neg2loglikelihood, method = "L-BFGS-B",
               lower=c(rep(-1,n),-10),upper=c(rep(1,n),10),
               control=list(maxit=200)))
}
```

```
#This function is a modified version of the test.H1Loc() R function.
#It can be used if optim fails to converge under H1.
#ATTENTION: Applying this function can lead to a biased likelihood ratio statistics
#           that can increase the probability of committing a type I error.

#Input: See the original test.H1Loc()
#Output: optim object with n+1 MLE parameters

#Author: Carina Schneider (2016)

test.H1Loc_VR <- function(y,mu1,a,tb,l0,b,c,f,rown,coln,m=nrow(y), Rstruct=NULL,...) {
  n <- coln*rown
  if (!is(Rstruct, "spam.chol.NgPeyton")) {
    Q <- mgmrf.prec(b,c,f,rown,coln,m)
    if (!is.spam(Q))
      stop("'Covariance' should return a spam object.")
    Rstruct <- chol.spam(Q,...)
  }
  Q <- mgmrf.prec(b,c,f,rown,coln,m)
  cholS <- update.spam.chol.NgPeyton(Rstruct,Q)

  neg2loglikelihood <- function(fulltheta,...) {
    resid1 <- t(y[1:tb,])-fulltheta[1:n]

    mu2 <- fulltheta[1:n]
    mu2[l0] <- mu2[l0]+fulltheta[n+1]
    resid2 <- t(y[(tb+1):m,])-mu2
    resid <- c(cbind(resid1,resid2))
    return(n*m*log(2*pi)-2*(c(determinant.spam.chol.NgPeyton(cholS)$modulus))+sum(resid*(Q%*%(resid))))

  }
  return(optim(c(mu1,a), neg2loglikelihood, method = "L-BFGS-B",
               lower= c(rep(-1,n),-10),upper=c(rep(1,n),10),control=list(maxit=200)))
}
```

```
#This function can be used if optim does not converge under H0/H1.
#It is very restrictive and fixes b under H0 and H1 and fixes all parameters under H1
#to the parameters obtained by test.H0()
#ATTENTION: Applying the function can lead to a biased likelihood ratio statistics with the
#           tendency to increase the probability of committing a type I error.

#input, output: see original gmrfHomogeneitytTestComp()

#Author: Carina Schneider (2016)

library(spam)
gmrfHomogeneityTestComp_VR <- function(desData,type,muStart,bStart,cStart,fStart, sigLevel,L=5){
  rown <- dim(desData)[1]
  coln <- dim(desData)[2]
  N <- rown*coln
  T <- dim(desData)[3]
  if(length(muStart)!=N){
    stop("mu does not have the right format. The length should equal the number of locations.")
  }
```

```
desDataNew <- desData
dim(desDataNew) <- c(N,T)

if(type=="global"){

  MLEH0 <- test.H0_VR(y=t(desDataNew),muStart,bStart,cStart,fStart,rown,coln)
  print(MLEH0)
  if(MLEH0$convergence!=0){
    stop("quasi Newton method did not converge under H0.")
  }
  par <- MLEH0$par[(N+1):(N+2)]
  lRatioStat <- numeric(T)
  a <- numeric(T)
  for(i in 2:(T-2)){
    if(i %% L==0){
      MLEH1 <- test.H1Glob_VR(y=t(desDataNew),muStart,a=0,i,bStart,par[1],par[2],rown,coln)
      cat("this is time ", i)
      print(MLEH1)
      if(MLEH1$convergence!=0){
        stop("quasi Newton method did not converge under H1 (i.e. having a global shift).")
      }
      else{
        lRatioStat[i] <- MLEH0$value-MLEH1$value
        if(lRatioStat[i]<0){
          lRatioStat[i] <- 0
          #stop("Likelihood ratio statistic <0.")
        }
        a[i] <- MLEH1$par[N+1]
      }
    }

  }
  evTime <- Matrix::nnzero(a)
  th <- qchisq(1-sigLevel/evTime, df=1)
  inhomoFound <- FALSE
  timeOfInhomo <- 0
  heightInhomo <- 0
  if(max(lRatioStat)>th){
    inhomoFound <- TRUE
    timeOfInhomo <- which.max(lRatioStat)
    heightInhomo <- a[timeOfInhomo]
  }
  print(data.frame(inhomoFound,timeOfInhomo,heightInhomo))
  return(data.frame(list(a=a,lRatioStat=lRatioStat)))

}


#---------
else if(type=="local"){

  MLEH0 <- test.H0_VR(y=t(desDataNew),muStart,bStart,cStart,fStart,rown,coln)
  print(MLEH0)
  if(MLEH0$convergence!=0){
    stop("Does not converge under H0")
  }
  par <- MLEH0$par[(N+1):(N+2)]
  a <- matrix(0,T,N)
  lRatioStat <- matrix(0,T,N)
  for(i in 2:(T-2)){
    if(i %% L==0){
    for(k in 1:N){
      MLEH1 <- test.H1Loc_VR(y=t(desDataNew),muStart,a=0,tb=i,
                             l0=k,bStart,par[1],par[2],rown,coln)
      if(MLEH1$convergence!=0){
        stop("Does not converge under H1.")
      }
      a[i,k] <- MLEH1$par[N+1]
      lRatioStat[i,k] <- MLEH0$value-MLEH1$value
    }
    }
  }
```

```
    evTime <- Matrix::nnzero(a[,1])
    th <- qchisq(1-sigLevel/evTime, df=1)
    inhomoFound <- FALSE
    timeOfInhomo <- 0
    heightInhomo <- 0
    locInhomo <- 0
    if(max(lRatioStat)>th){
      inhomoFound <- TRUE
      ind <- which.max(lRatioStat)
      timeOfInhomo <- ind %% T
      locInhomo <- ind %/% T +1
      heightInhomo <- a[timeOfInhomo, locInhomo]
    }
    print(data.frame(inhomoFound,timeOfInhomo,heightInhomo,locInhomo))
    return(list(lRatioStat=lRatioStat,a=a[,locInhomo]))
  }


  else{
    stop("This is not an admissible type.")
  }

}
```

```
# This script provides code to calculate the weighted mean of
# 3-dimensional NetCDF data file contents
# in a specific directory. The weights have been defined based on the number
# of ensembles a specific climate model
# has produced.

# Input:
# path: path to a directory containing all the NetCDF files of the same type:
#       I.e., same climate scenario, variable and spatial/temporal resolution
# weighted: If TRUE weighted means are built.
#           If FALSE all file contents are weighted the same. No weights are introduced.


# Output:
# ‘‘weighted’’ Mean: as a 3-dimensional array (lon x lat x time)

#Author: Carina Schneider (2016)

source(‘/.../meanBuilder.R’)

meanOfFiles<-function(path,weighted=TRUE){
  files<-list.files(path=path,pattern="*.nc",
                    full.names=T,recursive=FALSE)


  n<-length(files)
  if(n==1){
    stop("no mean needs to be computed. There is only 1 file in this directory.")
  }
  if(weighted==FALSE){
    return(meanBuilder(files))

  }
  else{
   basisfiles<-list.files(path=path,pattern="*.nc",
                                 full.names=F,recursive=FALSE)

   mod<-character(n)
   mod[1]<-as.character(read.table(text=basisfiles[1],sep="_")$V3)
   modList<-numeric(n)
   modList[1]<-1
    for(i in 2:n){
      mod[i]<-as.character(read.table(text=basisfiles[i],sep="_")$V3)
      if(mod[i]!=mod[i-1]){
        modList[i]<-1
      }
    }
   numbMod<-sum(modList)
```

```
   indList<-list()
   k<-1
   v<-which(modList==1)
   #print(v)
   frames<-list()
   if(length(v)==1){
     #only one model and only 1 frame that is the mean of all ensembles this model
     #produces.
     return(meanBuilder(files))
   }
   for(i in 2:length(v)){
     if(v[i]-v[i-1]==1){
       indList[[k]]<-v[i-1]
       k<-k+1
     }
     else{
       indList[[k]]<-v[i-1]:(v[i]-1)
       k<-k+1
     }
   }

   for(i in 1:length(indList)){
     frames[[i]]<-meanBuilder(files[indList[[i]]])
   }
   temp<-0
   for(i in 1:length(indList)){
     temp<-frames[[i]]+temp
   }
   weightedMean<-1/length(indList)*temp
   return(weightedMean)
  }
  }
```

```
#removes seasonality and trends with the GAMM
#data: data frame containing the months, years, data and time (see Section 5.5.2)

#Author: Carina Schneider (2016)

gamPeriodTrendRem <- function(data){
  new <- within(data, Date <- as.Date(paste(year, month, day.of.month="15",
                                        sep = "-")))
  plot(values ~ Date, data = new, type = "l")
  ctrl <- list(niterEM=0,msVerbose=TRUE,optimMethod="L-BFGS-B")
  mod <- gamm(values ~ s(month, bs = "cc",k=12) + s(time, bs = "cr"),
              data = new, control=ctrl)
 return(mod$gam$residuals)
}
```

```
# this function takes the mean and a certain file and gives back the
# standardized difference of the two
# path: to NetCDF file
library(ncdf)

difference <- function(Mean,path, standardize=T){
  file <- path
  nc <- open.ncdf(file)
  data <- get.var.ncdf(nc)
  close.ncdf(nc)
  dataNew <- Mean-data
  if(standardize==F){
    return(dataNew)
  }
  else{

    for(i in 1:(dim(dataNew))[1]){
      for(j in 1:(dim(dataNew))[2]){
        dataNew[i,j,] <- 1/sd(dataNew[i,j,])*dataNew[i,j,]
      }
    }
    return(dataNew/sd(dataNew))
  }
}
```

```
#this function returns the means of all models
#weighted according to the number of ensembles
#pathToDir: path to the directory of NetCDF files of the same type

#Author: Carina Schneider (2016)

getMeansOfModel <- function(pathToDir){
    files <- list.files(path=pathToDir,pattern="*.nc",
                        full.names=T,recursive=FALSE)
    basisfiles <- list.files(path=pathToDir,pattern="*.nc",
                          full.names=F,recursive=FALSE)
    n <- length(basisfiles)
    mod <- character(n)
    if(n==1){
      stop("Only one file in the directory found. No need to apply this function.")
    }
    mod[1] <- as.character(read.table(text=basisfiles[1],sep="_")$V3)
    modList <- numeric(n)
    #we know that at this point n>1 because of the exception above.
    modList[1] <- 1
    for(i in 2:n){
      mod[i] <- as.character(read.table(text=basisfiles[i],sep="_")$V3)
      if(mod[i]!=mod[i-1]){
        modList[i] <- 1
      }
    }
    numbMod <- sum(modList)
    indList <- list()
    k <- 1
    v <- which(modList==1)
    frames <- list()
    if(length(v)==1){
      #only one model and only 1 frame that is the mean of all ensembles this model
      #produces.
      return(meanBuilder(files))
    }
    frames <- list()
    if(length(v)==1){
      #only one model and only 1 frame that is the mean of all ensembles this model
      #produces.
      return(meanBuilder(files))
    }
    for(i in 2:length(v)){
      if(v[i]-v[i-1]==1){
        indList[[k]] <- v[i-1]
        k <- k+1
      }
      else{
        #not single group
        indList[[k]] <- v[i-1]:(v[i]-1)
        k <- k+1
      }
    }
    modelNames <- character(length(indList))
    for(i in 1:length(indList)){
      frames[[i]] <- meanBuilder(files[indList[[i]]])
      modelNames[i] <- mod[indList[[i]]]
    }
    names(frames) <- modelNames
    return(frames)
}
```

8.3.4 Lattice Krig source code

```
#This function gives back the difference of the basis function
#coefficients of the reference applied to Lattice Krig
#compared to the specific file-data.
#candidate: 2 dimensional array of dimension 144x72
#reference: 2 dimensional array of dimension 144x72
```

```
#           (does not have to be from the weighted mean, can be
#            any other reference model as well)
#lambda:    if NULL, then the estimate of the weighted mean is used,
#           otherwise it is fixed to the passed real value
#xNew:      matrix of coordinates of the grid points (longitudes, latitudes)

#Author: Carina Schneider (2016)

refLatTest <- function(candidate,reference,xNew, lambda=NULL){
  LKinfo <- LKrigSetup(x=xNew,nlevel=3, alpha=c(1/3,1/3,1/3),
                       a.wght=4.05,NC=36,NC.buffer=0, overlap=2.5)

  if(length(lambda)==0){
    meanObj <- LatticeKrig(x=xNew,c(reference),LKinfo=LKinfo)
    lambda <- meanObj$lambda.fixed
  }
  else{
    meanObj <- LKrig(xNew,c(reference),LKinfo=LKinfo,lambda=lambda)
  }
  obj <- LKrig(xNew,c(candidate),LKinfo=LKinfo,lambda=lambda)
  numbOfLevels <- length(LKinfo$latticeInfo$mLevel)
  ver <- numeric(numbOfLevels)
  for(i in 1:numbOfLevels){
    ver[i] <- sum(LKinfo$latticeInfo$mLevel[1:i])
  }
  plot(abs(meanObj$c.coef-obj$c.coef),type="l",
      xlab="Index of coefficients",ylab="Differences of coefficients",
      main="Differences of coefficients between reference and candidate")
  for(i in 1:(numbOfLevels-1)){
    abline(v=ver[i],col="red")
  }
  #print out the max, median and mean of the absolute differences of the
  #coefficients of the reference and candidate
  #in the first 3 levels
  absDiff <- abs(meanObj$c.coef-obj$c.coef)
  if(numbOfLevels==3){
    level1 <- absDiff[1:ver[1]]
    level2 <- absDiff[(ver[1]+1):ver[2]]
    level3 <- absDiff[(ver[2]+1):length(obj$c.coef)]
    max <- c(max(level1),max(level2),max(level3))
    med <- c(median(level1),median(level2),median(level3))
    mean <- c(mean(level1),mean(level2),mean(level3))
    infoMat <- cbind(max,med,mean)
    colnames(infoMat) <- c("max","median","mean")
    rownames(infoMat) <- c("level 1","level 2", "level 3")
    print(infoMat)
  }

  return(list(absDiff=abs(meanObj$c.coef-obj$c.coef),
              lambdaMean=lambda,modCoef=obj$c.coef,meanCoef=meanObj$c.coef))
}
```

```
#pathToDir: path to the directory with NetCDF files of the same type
#           (i.e. same variable,scenario & resolution)
#T:         time of evaluation
#lambda:    smoothing factor that is fixed

#Output:
#matrix with the files ordered according to the values of sigma_MLE

#Author: Carina Schneider (2016)

sigmaLatTest <- function(pathToDir,lambda,T,xNew){
  files <- list.files(pathToDir,full.names = TRUE)
  filesShort <- list.files(pathToDir,full.names = FALSE)
  LKinfo <- LKrigSetup(x=xNew,nlevel=3, alpha=c(1/3,1/3,1/3),
                       a.wght=4.05,NC=36,NC.buffer=0, overlap=2.5)
  sigma <- numeric(length(files))
  for(i in 1:length(files)){
    nc <- open.ncdf(files[i])
    data <- get.var.ncdf(nc)
    close.ncdf(nc)
```

```
    obj <- LKrig(xNew,c(data[,,T]),LKinfo=LKinfo,lambda=lambda)
    sigma[i] <- obj$sigma.MLE
  }
  ind <- order(sigma)
  sigma<-cbind(filesShort[ind],sigma[ind])
  colnames(sigma)<-c("file", "sigma")
  return(sigma)
}
```

```
#essentially the same as sigmaLatTest but for lambda...
#the lambda estimates can also be obtained by getLambda(...) when "type" is set to "median"

#Author: Carina Schneider (2016)

lambdaLatTest <- function(pathToDir,T,xNew){
  files <- list.files(pathToDir,full.names = TRUE)
  filesShort <- list.files(pathToDir,full.names = FALSE)
  LKinfo <- LKrigSetup(x=xNew,nlevel=3, alpha=c(1/3,1/3,1/3),
                       a.wght=4.05,NC=36,NC.buffer=0, overlap=2.5)
  lambda <- numeric(length(files))
  for(i in 1:length(files)){
    nc <- open.ncdf(files[i])
    data <- get.var.ncdf(nc)
    close.ncdf(nc)
    obj <- LatticeKrig(x=xNew,c(data[,,T]),LKinfo=LKinfo)
    #print(obj)
    lambda[i] <- obj$lambda.fixed
  }
  ind <- order(lambda)
  mat <- (cbind(filesShort[ind],lambda[ind]))
  colnames(mat)<- c("file","lambda")
  return(mat)
}
```

```
#get coordinates
#indLat: Latitude indices of the spatial locations
#indLon: Longitude indices of the spatial locations
#E.g. indLat=c(1:10), indLon=c(55:60) are 60 locations over central Europe

#Author: Carina Schneider (2016)

getCoordinates <- function(indLon,indLat){
  LAT <- seq(-88.75,88.75,by=2.5)
  LON <- c(seq(1.25,178.75,by=2.5),(-1)*rev(seq(1.25,178.75,by=2.5)))
  LAT <- LAT[indLat]
  LON <- LON[indLon]
  col <- length(LON)
  row <- length(LAT)
  coordLat <- rep(LAT[1],col)
  for(i in 2:row){
    coordLat <- cbind(coordLat,rep(LAT[i],col))
  }
  coordLat <- c(coordLat)
  coordLon <- rep((LON),row)
  return(data.frame(coordLon=coordLon,coordLat=coordLat))
}
```

Bibliography

Alexandersson, H. and Moberg, A. (1997). Homogenization of Swedish temperature data. Part I: Homogeneity test for linear trends. *International Journal of Climatology*, 17(1):25–34.

Browning, J. and Schneider, C. (2015). snht: Standard Normal Homogeneity Test. R package version 1.0.3. `https://cran.r-project.org/web/packages/snht/index.html`. [Online; accessed 14-12-2015].

Browning, J. and Schneider, C. (2016). snht: Standard Normal Homogeneity Test. R package version 1.0.4.

Canty, A. and Ripley, B. (2015). boot: Bootstrap Functions (Originally by Angelo Canty for S). R package version 1.3-17.

Covey, C., AchutaRao, K. M., Cubasch, U., Jones, P., Lambert, S. J., Mann, M. E., Phillips, T. J., and Taylor, K. E. (2003). An overview of results from the Coupled Model Intercomparison Project. *Global and Planetary Change*, 37(1):103–133.

Furrer, R. (2015). spam: SPArse Matrix. R package version 1.3-0.

Haimberger, L. (2005). *Homogenization of radiosonde temperature time series using ERA-40 analysis feedback information*. European Centre for Medium-Range Weather Forecasts.

Held, L. and Bové, D. S. (2013). *Applied Statistical Inference: Likelihood and Bayes*. Springer Science and Business Media.

IPCC (2016). IPCC Organization. `http://www.ipcc.ch/organization/organization.shtml`. [Online; accessed 18-February-2016].

Masson, D. and Knutti, R. (2011). Climate model genealogy. *Geophysical Research Letters*, 38(8).

Menne, M. J. and Williams Jr, C. N. (2009). Homogenization of temperature series via pairwise comparisons. *Journal of Climate*, 22(7):1700–1717.

Nychka, D., Bandyopadhyay, S., Hammerling, D., Lindgren, F., and Sain, S. (2013). A Multi-Resolution Gaussian process model for the analysis of large spatial data sets. *Journal of Computational and Graphical Statistics*, 24(2).

Nychka, D., Hammerling, D., Sain, S., and Lenssen, N. (2015). LatticeKrig: Multiresolution Kriging Based on Markov Random Fields. R package version 5.4-1.

Pierce, D. (2015). ncdf: Interface to Unidata netCDF Data Files. R package version 1.6.8.

Rue, H. and Held, L. (2005). *Gaussian Markov Random Fields: Theory and Applications*. CRC Press.

Schibli, R. (2011). Spatio-temporal homogeneity of a satellite-derived global radiation climatology. Master's thesis, University of Zurich.

Stocker, T., Qin, D., Plattner, G., Tignor, M., Allen, S., Boschung, J., Nauels, A., Xia, Y., Bex, B., and Midgley, B. (2013). IPCC, 2013: Climate Change 2013: the Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change.

Taylor, K. (2009). CMIP-Coupled Model Intercomparison Project-Overview. `http://cmip-pcmdi.llnl.gov/`. [Online; accessed 12-October-2015].

Taylor, K. (2013a). CMIP5-Data Access-Availability. `http://cmip-pcmdi.llnl.gov/cmip5/availability.html`. [Online; accessed 12-October-2015].

Taylor, K. (2013b). CMOR Table. `http://cmip-pcmdi.llnl.gov/cmip5/docs/standard_output.pdf`. [Online; accessed 12-October-2015].

Taylor, K. (2014). CMIP5 Experiments. `http://www.ipcc-data.org/sim/gcm_monthly/AR5/CMIP5-Experiments.html`. [Online; accessed 12-October-2015].

Toreti, A., Kuglitsch, F., Xoplaki, E., Della-Marta, P., Aguilar, E., Prohom, M., and Luterbacher, J. (2011). A note on the use of the standard normal homogeneity test to detect inhomogeneities in climatic time series. *International Journal of Climatology*, 31(4):630–632.

Van Loan, C. F. (2000). The ubiquitous kronecker product. *Journal of computational and applied mathematics*, 123(1):85–100.

Wendland, H. (1995). Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in computational Mathematics*, 4(1):389–396.

Wood, S. (2015). gamair: Data for "GAMs: An Introduction with R". R package version 0.0-9.

Young, D. S. (2015). tolerance: Functions for Calculating Tolerance Intervals. R package version 1.1.0.