

The influence of bias corrections on variability, distribution, and correlation of temperatures in comparison to observed and modeled climate data in Europe

Bachelor Thesis

Benjamin Thomas Schwertfeger

36036

Faculty 2

University of Applied Science Bremerhaven, Germany

Reviewer: Prof. Dr. Henrik Lipskoch

Supervisor: Prof. Dr. Gerrit Lohmann

June 20, 2022

Abstract

Bias correction algorithms for modeled climate variables such as temperature, precipitation, and barometric pressure are used to approximate certain aspects of the distribution characteristics to the actual observed values. Thus, modeled climate data predicting future climate scenarios can be bias-adjusted using data from past periods so that climate variables and their distribution, as well as their variability, can be represented more realistically within the bias-adjusted time series. For this reason, it is essential to understand how such bias adjustment algorithms work and what impact they can have on the underlying data.

This bachelor thesis aims to find out and show how bias adjustment procedures work, how they can be implemented and applied in different programming languages, and what influence the application of such techniques can have on modeled temperature data for the region of Europe and its surroundings.

This has been done by demonstrating and implementing five different bias adjustment procedures mathematically as well as in the programming languages Python and C++ and then applying different methods from the field of statistics in a detailed analysis to show the influence as well as the limitations of these techniques.

Contents

Abstract	i
List of Figures	vii
List of Tables	ix
List of Listings	xi
List of Abbreviations	xii
1. Introduction	1
1.1. Climate Models	1
1.2. Bias Adjustments	5
1.3. About this Thesis	8
2. Motivation	11
3. Climate Data	12
3.1. Modeled Data	14
3.2. Observational / Reanalysis Data	17
3.3. Data Preprocessing	19
4. Approach / Methods	21
4.1. The Delta Method	24
4.2. The Linear Scaling Method	26
4.3. The Variance Scaling Method	30
4.4. The Quantile Mapping Method	33
4.5. The Quantile Delta Mapping Method	39
4.6. Extensive Implementation	43
4.7. Hands on C++	48
4.7.1. Scaling-based Bias Adjustment Techniques in C++	48
4.7.2. Distribution-based Bias Adjustment Techniques in C++	53
4.8. Method Discussion	60
5. Results	61
5.1. Method Evaluation	61
5.1.1. Mean Bias Error	64

5.1.2. Root Mean Square Error	73
5.1.3. Variance and Standard Deviation	79
5.2. Change in Quantiles and Distribution	83
5.3. Correlation and minimal Correlation Maps	89
5.3.1. Change in Correlation between different Locations	91
5.3.2. Minimal Correlation Maps	95
6. Conclusion	106
Bibliography	I
Appendix	IV
A. Figures	IV
Affidavit	XII

List of Figures

1.1. Representation of a climate model grid and information exchange between grid boxes (Figure by Kotamarthi et al., 2021)	3
1.2. Different emissions simulations run by CMIP6 (Figure by Böttinger et al., 2020)	4
1.3. Modeled and observed mean 2m air temperatures by day of the year between 1981 and 2010 in Bremerhaven, Germany	6
1.4. Simplified representation of a general bias adjustment procedure	7
3.1. Simplified representation of the dimension structure of a four dimensional NetCDF data set containing temperature values; Dimensions: Time (without subscript), Longitude, Latitude and Altitude (without subscript)	12
3.2. The NetCDF data model (Figure by NetCDF-C Documentation, 2022)	13
3.3. Schematic overview of the components of the MPI-ESM1-2-HR (Figure by Mauritsen et al., 2019)	15
3.4. MPI-ESM1-2-HR model output; mean of daily 2m air temperatures in Europe between 1981 and 2010; -45°W to 65°E by 1 deg. East, 23°N to 72°N by 1 deg. North	20
4.1. Comparison of observed, modeled and linear scaling adjusted monthly mean 2m air temperatures in Bremerhaven at different periods	28
4.2. Comparison of observed, modeled and linear scaling adjusted monthly mean 2m air temperatures in Europe at different periods	28
4.3. Determination of the quantile value by inserting $T_{sim,p}(6190)$ into the CDF of the modeled time series of the control period; black lines indicate the 25% and 75% quantile respectively	33
4.4. Inserting the quantile value of 0.28 into the inverse CDF of historical observational data ($F_{obs,h}^{-1}$) to equate the adjusted temperature value of $T_{sim,p}(6190)$; black lines indicate the 25% and 75% quantile respectively	34
4.5. Cumulative distribution functions of observed, modeled and QM-adjusted daily mean 2m air temperatures in Bremerhaven, Germany for different time periods	34
4.6. Probability density function as line and bar plot of observed daily mean temperatures in Bremerhaven, Germany between 1951 and 1980 as a representation of the need of xbins	36

5.1. Long-term monthly mean <i>MBE</i> in Europe based on daily mean 2m air temperatures (1981-2010)	66
5.2. Mean 2m air temperatures per day of the year in Europe between 1981 and 2010; grey lines represent the daily mean 2m air temperatures of each individual year in the DM-adjusted data set	67
5.3. Mean 2m air temperature difference between modeled and observed and bias-adjusted and observed per day of the year in Europe (1981-2010)	68
5.4. Mean bias error of daily mean 2m air temperatures in Europe (1981-2010)	69
5.5. Mean Bias Error of daily 2m air temperatures in Europe (1981-2010)	70
5.6. Mean bias error of daily mean 2m air temperatures in Europe (1981-2010)	71
5.7. Mean bias error of daily mean 2m air temperatures between modeled and observed data in the control and scenario period	71
5.8. Mean <i>RMSE</i> per month based on daily mean 2m air temperatures in Europe (1981-2010); dotted lines indicate the mean	75
5.9. <i>RMSE</i> between observed, modeled and adjusted daily mean 2m air temperatures in Europe (1981-2010)	76
5.10. Change of <i>RMSE</i> between bias-adjusted and modeled data based on daily mean 2m air temperatures in Europe (1981-2010)	77
5.11. Difference in variance between modeled and observed and bias-adjusted and observed data sets based on daily mean 2m air temperatures in Europe (1981-2010)	80
5.12. Mean variance per month based on daily mean 2m air temperatures in Europe (1981-2010); dotted lines indicate the mean	82
5.13. Q-Q plot of daily mean 2m air temperatures in Bremerhaven (1981-2010)	83
5.14. Q-Q plot of daily mean 2m air temperatures in Oslo (1981-2010)	84
5.15. Q-Q plot of daily mean 2m air temperatures in Milan (1981-2010)	84
5.16. Q-Q plot of daily mean 2m air temperatures in (A) Zurich and (B) Lyon (1981-2010)	85
5.17. Probability density functions of daily mean 2m air temperatures in observed, modeled and bias-adjusted data sets in Europe (1981-2010)	86
5.18. Seasonal probability density functions of daily mean 2m air temperatures in observed, modeled and bias-adjusted data sets in Europe (1981-2010)	87
5.19. Correlation coefficient per year based on daily mean 2m air temperatures in Zurich and Bremerhaven (1981-2010); horizontal lines represent the mean	91
5.20. Correlation coefficient per year; based on daily mean 2m air temperatures in Zurich and Oslo (1981-2010); horizontal lines represent the mean	92
5.21. Correlation coefficient per year; based on daily mean 2m air temperatures in Zurich and Lyon (1981-2010); horizontal lines represent the mean	92
5.22. Correlation coefficient per year; based on daily mean 2m air temperatures in Milan and Lyon (1981-2010); horizontal lines represent the mean	93
5.23. Correlation coefficient per year; based on daily mean 2m air temperatures in Bremerhaven and Iceland (1981-2010); horizontal lines represent the mean	93

5.24. Minimal correlation maps of the observed (NOAA) and modeled (MPI-ESM) data set based on monthly mean 2m air temperatures between 1981 and 2010	97
5.25. Minimal correlation maps of LS- and VS-adjusted data set based on monthly mean 2m air temperatures between 1981 and 2010	98
5.26. Minimal correlation map of DM-adjusted data based on monthly mean 2m air temperatures between 1981 and 2010	98
5.27. Minimal correlation maps of QM- and QDM-adjusted data set based on monthly mean 2m air temperatures between 1981 and 2010	99
5.28. Minimal correlation maps of the observed (NOAA) and modeled (MPI-ESM) data set based on daily mean 2m air temperatures in winter months between 1981 and 2010	100
5.29. Minimal correlation maps of LS- and VS-adjusted data set based on daily mean 2m air temperatures in winter months between 1981 and 2010	101
5.30. Minimal correlation map of DM-adjusted data based on daily mean 2m air temperatures in winter months between 1981 and 2010	101
5.31. Minimal correlation maps of QM- and QDM-adjusted data set based on daily mean 2m air temperatures in winter months between 1981 and 2010	102
5.32. Minimal correlation maps of the observed (NOAA) and modeled (MPI-ESM) data set based on daily mean 2m air temperatures in summer months between 1981 and 2010	103
5.33. Minimal correlation maps of LS- and VS-adjusted data sets based on daily mean 2m air temperatures in summer months between 1981 and 2010	103
5.34. Minimal correlation map of DM-adjusted data based on daily mean 2m air temperatures in summer months between 1981 and 2010	104
5.35. Minimal correlation maps of QM- and QDM-adjusted data set based on daily mean 2m air temperatures in summer months between 1981 and 2010	104
A.1. Mean 2m air temperatures per day of the year in Europe between 1981 and 2010; grey lines represent the daily mean 2m air temperatures of each individual year in LS-adjusted data	IV
A.2. Mean 2m air temperatures per day of the year in Europe between 1981 and 2010; grey lines represent the daily mean 2m air temperatures of each individual year in VS-adjusted data	IV
A.3. Mean 2m air temperatures per day of the year in Europe between 1981 and 2010; grey lines represent the daily mean 2m air temperatures of each individual year in QM-adjusted data	V
A.4. Mean 2m air temperatures per day of the year in Europe between 1981 and 2010; grey lines represent the daily mean 2m air temperatures of each individual year in QDM-adjusted data	V
A.5. Q-Q plots of daily mean 2m air temperatures in winter months (DJF) between 1981 and 2010 in (A) Bremerhaven, (B) Oslo, (C) Milan, (D) Zurich and (E) Lyon	VI

A.6. Q-Q plots of daily mean 2m air temperatures in summer months (JJA) between 1981 and 2010 in (A) Bremerhaven, (B) Oslo, (C) Milan, (D) Zurich and (E) Lyon	VII
A.7. Q-Q plots of daily mean 2m air temperatures between 1981 and 2010 in (A) Bremerhaven, (B) Oslo, (C) Milan, (D) Zurich and (E) Lyon	VIII
A.8. Q-Q plots of daily mean 2m air temperatures between 1981 and 2010 in (A) Bremerhaven, (B) Oslo, (C) Milan, (D) Zurich and (E) Lyon	IX
A.9. Q-Q plots of daily mean 2m air temperatures between 1981 and 2010 in (A) Bremerhaven, (B) Oslo, (C) Milan, (D) Zurich and (E) Lyon	X
A.10. Q-Q plots of daily mean 2m air temperatures between 1981 and 2010 in (A) Bremerhaven, (B) Oslo, (C) Milan, (D) Zurich and (E) Lyon	XI

List of Tables

1.1. Thesis overview and topics	9
1.2. Repositories containing the implementation of the adjustment procedures used in this thesis	9
1.3. Definitions of terms and meanings	10
3.1. Earth system model data set specifications for DKRZ file search	16
4.1. Mathematical symbols, abbreviations and sub-/superscripts	21
4.2. Symbols and abbreviations used in listings	22
4.3. Symbolic example and description of the 1D sequence of temperatures of all Januaries in $T_{sim,p}$; values are 2m air temperatures in deg. Celsius	23
4.4. Available methods for long-term monthly scaling-based bias adjustment implemented in Python	43
5.1. Coordinates of selected locations in Europe	62
5.2. Coordinates of the nearest cell of selected locations	62
5.3. <i>MBE</i> of daily mean 2m air temperatures between observed, modeled and adjusted data respectively in deg. Celsius at different locations in Europe (1981-2010)	64
5.4. Minimum, mean and maximum <i>MBE</i> of daily mean 2m air temperatures between observed, modeled and bias-adjusted data in deg. Celsius in Europe (1981-2010)	65
5.5. Mean <i>MBE</i> per month based on daily mean 2m air temperatures in Europe in deg. Celsius (1981-2010)	65
5.6. <i>RMSE</i> between observed, modeled and bias-adjusted daily mean 2m air temperatures in deg. Celsius in Europe (1981-2010)	73
5.7. Minimum, mean and maximum <i>RMSE</i> of daily mean 2m air temperatures in deg. Celsius in Europe between observed, modeled and bias-adjusted data sets (1981-2010)	74
5.8. Mean <i>RMSE</i> per month based on daily mean 2m air temperatures in Europe in deg. Celsius (1981-2010)	74
5.9. Variance of observed, predicted and bias-adjusted daily mean temperatures in deg. Celsius at selected grid boxes in Europe (1981-2010)	79

5.10. Minimum, mean and maximum deviation in variance of daily mean 2m air temperatures in deg. Celsius in Europe; observed minus modeled and bias-adjusted data sets (1981-2010)	81
5.11. Monthly mean variance of daily mean 2m air temperatures in deg. Celsius in Europe (1981-2010)	81
5.12. Correlation coefficients based on daily mean 2m air temperatures at different locations between observed, modeled and bias-adjusted data sets (1981-2010)	90
5.13. Minimum, mean and maximum correlation coefficient of daily mean 2m air temperatures between observed, modeled and bias-adjusted data in Europe (1981-2010)	90
5.14. Minimum-, mean- and maximum-minimum correlation based on monthly mean 2m air temperatures in Europe (1981-2010)	97
5.15. Minimum-, mean- and maximal-minimum correlation based on daily mean 2m air temperatures in winter months in Europe (DJF, 1981-2010)	99
5.16. Minimum-, mean- and maximum-minimum correlation based on daily mean 2m air temperatures in summer months in Europe (JJA, 1981-2010)	102

Listings

1.1. Installation of required packages using the Python package manager Pip . . .	10
1.2. Import required modules into a Python script	10
3.1. Output of <code>ncdump</code> command applied an a NetCDF data set; three dots are symbolic for more rows of information	14
3.2. Apply observational grid configuration to the modeled data set, change the unit of temperatures from Kelvin to deg. Celsius and select desired timespan using <code>cdo</code>	19
3.3. Select European region using <code>cdo</code>	19
3.4. Select specific timespan of observational data using <code>cdo</code>	20
4.1. Implementation of the additive and multiplicative delta method	25
4.2. Implementation of the additive and multiplicative linear scaling method . . .	27
4.3. Implementation of the variance scaling method for temperatures	32
4.4. Define monotonously increasing vector <code>xbins</code> , containing equally sized bins between the maximum and minimum historical modeled and observed temperature values	35
4.5. Define function <code>get_cdf</code> that computes the <i>PDF</i> of a vector (i.e. 1D sequence) by given probability boundaries (i.e. <code>xbins</code>) to equate and return the <i>CDF</i>	36
4.6. Implementation of the quantile mapping method (Part 1)	37
4.7. Implementation of the quantile mapping method (Part 2)	37
4.8. Execute the quantile mapping method	37
4.9. Implementation of function <code>get_inverse_cdf</code> that computes the inverse <i>CDF</i> and enables inserting values in one step	41
4.10. Implementation of quantile delta mapping method (Part 1)	41
4.11. Implementation of quantile delta mapping method (Part 2)	42
4.12. Application of the quantile delta method	42
4.13. Loading $T_{obs,h}$, $T_{sim,h}$ and $T_{sim,p}$ into a Python script	43
4.14. Selection of the desired adjustment method and preparation of the results data set by deep copying all attributes and values of the $T_{sim,p}$ time series .	44
4.15. Create dictionary containing indices grouped by month	44
4.16. Pseudocode representing the grouped key-value pairs containing monthly separated indices	44

4.17. Declaration and initialization of the 1D sequences containing all temperature values of all same named months: $m(T_{obs,h}(i))$, $m(T_{sim,h}(i))$ and $m(T_{sim,p}(i))$	45
4.18. Application of the adjustment method and mapping of the computed results to the right position in the result sequence	45
4.19. Output first 10 values of $T_{obs,h}$, $T_{sim,h}$, $T_{sim,p}$ and $T_{sim,p}^{*LS}$	46
4.20. Save adjusted data set to file	46
4.21. Data set preparation for scaling-based bias adjustment	49
4.22. Implementation of the additive and multiplicative delta method in C++	50
4.23. Implementation of the additive and multiplicative linear scaling method in C++	50
4.24. Implementation of the variance scaling method in C++	51
4.25. Execution of monthly separated adjustment and merge to final adjusted data set using Bash and cdo	52
4.26. Function to create a monotonically increasing double vector	53
4.27. Function to create the probability density function in C++	54
4.28. Function to create the cumulative distribution function in C++	55
4.29. Implementation of the quantile mapping bias adjustment procedure in C++ (Part 1)	55
4.30. Implementation of the quantile mapping bias adjustment procedure in C++ (Part 2)	56
4.31. Implementation of the quantile delta mapping bias adjustment procedure in C++ (Part 1)	57
4.32. Implementation of the quantile delta mapping bias adjustment procedure in C++ (Part 2)	57
4.33. Pseudocode to demonstrate the quantile delta mapping function call	58
5.1. Compute the European unweighted mean	63

List of Abbreviations

AWI	Alfred Wegener Institute, Helmholtz Centre for Polar and Marine Research
MPI	Max-Plank Insitute for Meteorologie
NOAA	National Oceanic and Atmospheric Administration
PSL	Physical Sciences Laboratory
CIRES	Cooperative Institute for Research in Environmental Sciences
CMIP	Coupled Model Intercomparison Project
GCM	General Circulation Model
ESM	Earth System Model
RCM	Regional Climate Model
NWP	Numerical Weather Prediction Model
PDF	Probability Density Function
CDF	Cumulative Distribution Function
DM	Delta Method
LS	Linear Scaling
VS	Variance Scaling
QM	Quantile Mapping
QDM	Quantile Delta Mapping
MBE	Mean Bias Error
RMSE	Root Mean Square Error

1. Introduction

Scientists from all over the world are conducting research at various institutions and educational facilities in order to better understand the earth's climate and the physical as well as biochemical processes and their interrelationships. This is no longer a matter of simple simulations and weather forecasts, but of large-scale climate studies that use computer models to map the earth's climate for past and future time periods. Based on these climate model outputs, conclusions can be drawn about various processes and a wide variety of climatic scenarios can be simulated. With climate models, the influence of humans on the earth's climate and, above all, climate change, which is increasingly influencing the actions of human life can be investigated.

Even though these highly complex computer models are state of the art in technology and science, they are still not able to fully represent the multiplicity and complexity of small-scale physical processes and climate variability. Due to memory limitations, as well as the uncertainty because of parameterized small-scale processes that have been little or unexplored so far, it turns out that modeled climate data show deviations in their distribution properties of different climate variables, such as temperature, precipitation, and air pressure, compared to observed data.

Due to these systematic errors, also called bias, algorithms are used to adjust modeled climate data sets to approximate certain distributional properties, such as the variance and short- or long-term means of different periods, to the time series of observed data.

Bias adjustments of climate data take place wherever studies are conducted with data from climate models. In paleoclimate, climate impact, and climate change studies, in which different scenarios of the earth's climate are simulated, investigated, and evaluated, the data must undergo bias adjustment before analysis and further use because trends and distribution properties of the modeled data can thereby be optimized and adjusted based on the values that actually occurred in the past. This not only enables more accurate predictions to be made about time series of future periods but also offers the possibility of optimizing existing models with the help of bias-adjusted data sets.

1.1. Climate Models

Modeling in general is characterized by a simplified representation of complex and partly unexplored processes. Climate models, which simulate the earth's climate, are no longer only simplified representations of complex climatic processes, as it was at the beginning of climate modeling in the second half of the 20th century. Today, climate models are highly complex computer algorithms, which are based on current scientific knowledge from the fields of physics, biology, chemistry, mathematics, and many others and use a large number of already known and partly also experimental equations and parameters to describe the earth's climate and of course help scientists to understand and study the climate of past and future time periods.

In climate models, the earth and its atmosphere as well as the ocean and land surface and vegetation are mapped into 3-dimensional grid boxes, which calculate the physical processes, such as the carbon cycle, biochemical relationships of organisms, and also heat, dust, water and vapor exchange by simulating the interactions between different climate variables like precipitation, solar radiation, air, sea-surface and land temperatures in different altitudes. These variables and also processes like drying or the influence of seasonality are calculated and estimated for the respective grid box for one time step and are used then as the initial state for the following time step. These grid boxes not only pass on information to the next time step but also influence the surrounding grid boxes (i.e. spatial regions; cf. Fig. 1.1). This means that regional and global processes can also be mapped and simulated, in that processes do not only occur in individual grid boxes but can spread over large areas.

Some of these models also use information and data obtained through empirical studies as part of the initial arguments from which the calculations begin or continue. Unlike ordinary weather forecasting, which only covers very short time periods, scientists use climate models to explore long-term processes and trends in climate under specific framework conditions.

These climate models can be very different from each other because as in many fields of science, it is important to be specialized in certain areas. For this reason, there are climate models that are specialized in individual components of the Earth's climate (atmosphere, hydrosphere, cryosphere, land surface, and biosphere) by describing and simulating the processes occurring therein as realistically as possible (Stocker, 2011).

- Atmospheric models use equations that describe the dynamics and thermodynamics (air and energy circulation) at different longitudes, latitudes, and altitudes, but also processes close to the earth's surface, such as weather, clouds, air temperatures, precipitation, air pressure, water vapor, wind and storms, heat exchange, radiation and many more of these well-known variables and processes are considered within these models.
- Ocean circulation models can simulate most of the known variables and processes in and on the water, both in the oceans and in lakes and seas. Temperatures and heat and vapor transport, as well as flow behavior and the carbon budget, are also central topics here.
- The melting of polar regions and glaciers, as well as the associated rise in sea level and the release of carbon once trapped under ice, can be simulated using cryosphere and sea ice models. These models are also used primarily for studying the paleoclimate in order to learn from past processes and compare them with current processes.
- There are also models for the simulation of land surface. These include plate tectonics, orography, distribution of land mass and sea level, as well as the classification of climate zones and their characteristics with respect to climate. An important part here is also the type of soils, because different substrates reflect long- and short-wave radiation differently and also have different energy storage and transport properties.

- Finally there are also biosphere and geochemical models. These are specialized in the transfer, storage and exchange of CO₂ and other gases and substances. Thus, they allow to locate the sources and reservoirs of different substances, as well as to simulate them under certain conditions. This not only has an impact on the organic land mass, which is also modeled, but also serves to represent the earth's carbon budget and its reflectivity as realistically as possible, in order to better understand processes such as climate change.

There are several other models that simulate the climate or individual processes of the earth such as energy balance models, models that only simulate the carbon cycle, marine ecosystems, only specific atmospheric chemistry, and many more.

Since climatic conditions and influences can vary greatly depending on the location, each region has its climatic characteristics and is influenced by local events. Thus, the aforementioned components are not always found in the same combination or expression but are specifically selected and weighted depending on the study.

General circulation models (GCM) consist of a combination of an atmospheric and an oceanographic component but they can also occur individually depending on the focus of the study. GCMs are designed to simulate the fundamental physical processes in the sea and atmosphere together or individually to map the earth's climate globally. In this way, physical dependencies, interrelationships, and fundamental climatic issues can be investigated. The latest trend in climate modeling is the use of earth system models (ESM). ESMs not only serve to understand the fundamental physical processes of dynamics and thermodynamics like GCMs, but they can also realistically represent the biochemical processes of the atmosphere and the oceans, the exchange of substances, and the interactions between land and sea mass, vegetation, and atmosphere, as well as a realistic carbon cycle (Kotamarthi et al., 2021).

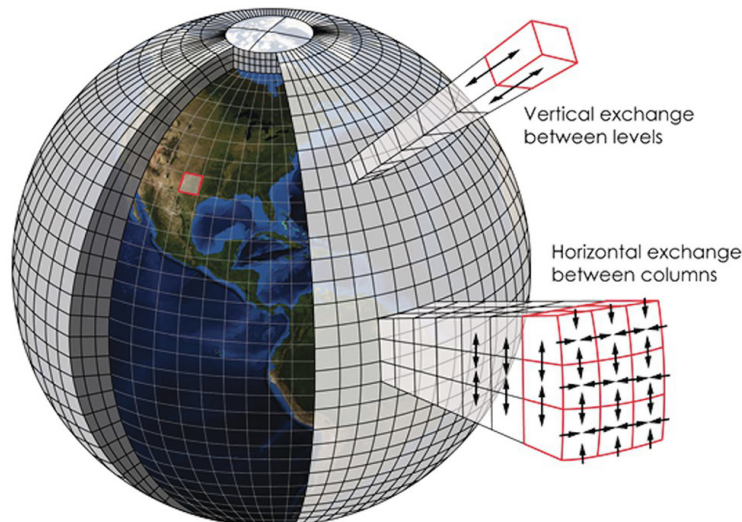


Figure 1.1.: Representation of a climate model grid and information exchange between grid boxes (Figure by Kotamarthi et al., 2021)

This is accomplished by coupling many different specialized climate and process models,

which allows the strengths of a wide variety of models to be combined to run even more comprehensive and accurate simulations, enabling scientists to make more substantiated statements about the earth’s climate of the past, present, and future.

The Coupled Model Intercomparison Project Phase 6 (CMIP6) is one of the best-known and widely used sources of modeled climate data. This project links scientists and institutions around the world, promotes and organizes research and the development of various climate models, and has contributed to the simulation of about 120 global climate models in 45 institutions and organizations (“WCRP Coupled Model Intercomparison Project (CMIP)”, 2022). One of these climate models is the MPI-ESM1-2-HR, whose data will be researched later in this thesis.

General circulation models and also earth system models, which aim to model climatic processes of the entire globe, can represent the climate well on average over medium and long periods of time, but show significantly higher deviations of climate parameters such as temperature and precipitation compared to observed data, than models that have been developed specifically for a certain region or a special event. For this reason, regional circulation models (RCM) are often used. These have a much higher resolution and are only used for a specific region, e.g. Europe or the Mediterranean. Due to a resolution of less than 50 km at the equator, they have much smaller cells for which the climate can be simulated. Depending on the topography, these RCMs can even have resolutions of a few kilometers and thus offer the possibility to map processes also on a regional level much better than GCMs, which typically only have a resolution of 100 - 250 km.

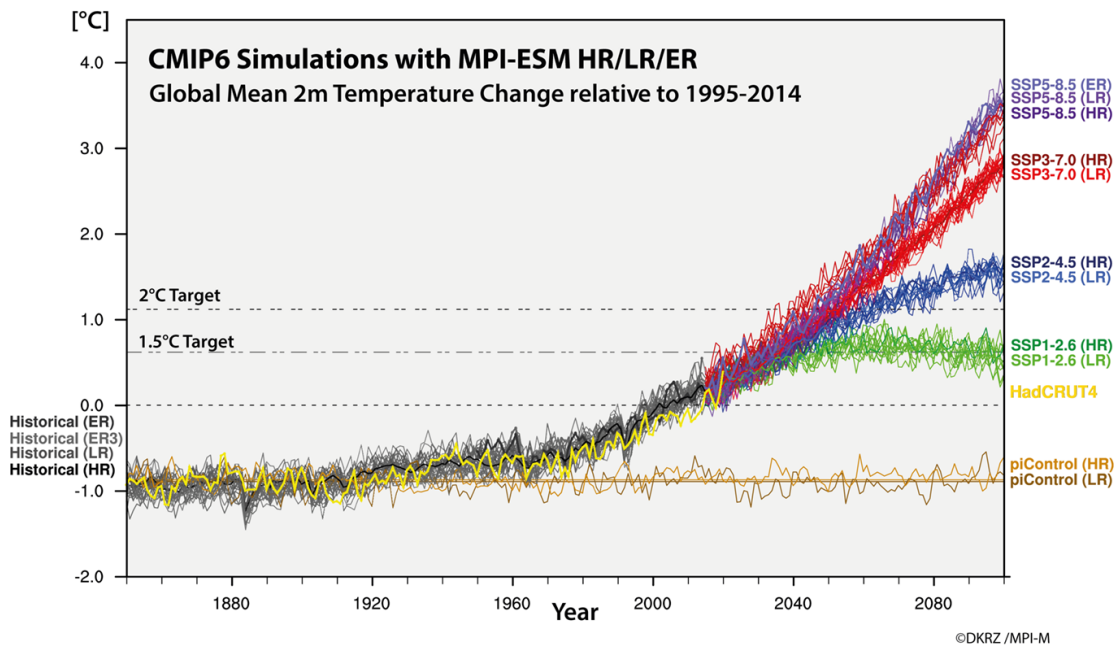


Figure 1.2.: Different emissions simulations run by CMIP6 (Figure by Böttinger et al., 2020)

A central topic of climate modeling is also climate change because it already has a strong impact on many areas of human life and the environment. Transportation, urban planning, water supply, as well as forestry and agriculture will have to adapt much more to climatic

changes in the future. Climate modeling increasingly takes into account the impact of humans on the environment, too. Therefore, research on different numerical climate models is an important process to simplify our understanding of our environment and at the same time to show how and by which means we can provide for a diverse and livable life on planet Earth. Modeling can also produce information about past climate events and thus simplify projections for the future by running simulations of different emission scenarios (also called RCP; Representative Concentration Pathway, Vuuren et al., 2011) that simulate the earth’s climate under the influence of certain concentrations of greenhouse gases (cf. Fig. 1.2), or even the abrupt increase of these.

These simulations are not only of interest to scientists but are also disseminated in various forms (e.g. IPCC Assessment Reports) to governments and organizations in order to take appropriate measures to curb CO₂ emissions, as well as other climate targets defined in the Paris Climate Agreement, including limiting global warming to a maximum of 1.5°C compared to preindustrial levels (“The Paris Agreement”, 2015).

Thus, modeling the earth’s climate can help recreate a variety of complex processes and interactions of the climate by writing down known and suspected physical, biological, and chemical processes as realistically as possible in the form of equations and conditions in millions of lines of source code in climate models to produce usable results for further research and development in the most time and energy-efficient manner.

1.2. Bias Adjustments

So there are several different models that can simulate the earth’s climate globally, but also regionally under certain framework conditions. Here, of course, the question arises as to how accurately the outputs of such models agree with the temperatures, precipitation, and other variables that actually occurred. For the validation of GCMs and ESMs, past climate forcings such as volcanic eruptions, changing CO₂ concentrations and sea levels, as well as temperatures and precipitation, can be passed to the initial arguments of the models to check whether the results generated in this way are similar to the assumed or even observed historical data obtained from ice core drillings, sediments, corals and other climate proxies. Of course, this assumes that observed data are already available for the time period under investigation. Thus, although it is possible to verify that the modeled processes are realistically represented, there are still discrepancies between the modeled and observed values. This becomes apparent when the data are not considered as a whole, but at regional and local scales as well as at short time scales.

Due to the limited resolution of these climate models and their large-scale grid cells, the topography and generalization of the processes cannot be mapped equally everywhere, which can lead to high deviations between observed and modeled data on a regional scale, since small-scale processes cannot be fully represented. These statistical deviations and errors are called “bias” in climate sciences. Depending on the focus of the research, bias can describe different aspects of the distributional properties of a climate variable, such as the mean, the variance and the standard deviation. Also, the distribution of certain

quantiles and the extremes can only be determined for one cell individually at a time in a climate model, which means that these cells only represent averaged values for a relatively large area, depending on the model resolution.

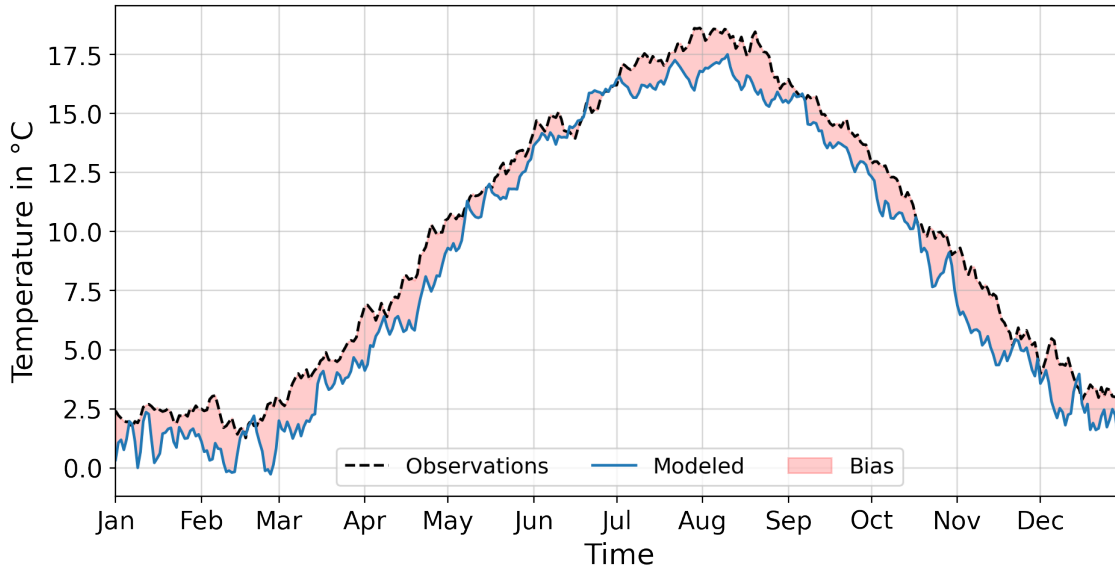


Figure 1.3.: Modeled and observed mean 2m air temperatures by day of the year between 1981 and 2010 in Bremerhaven, Germany

These deviations are particularly troublesome in research when it comes to conducting studies on threshold and extreme value analyses, because even if modeled data can realistically represent current behavior, tides, and seasonal changes in temperature and precipitation, these data in raw form are not good enough, for example, for research on the development and behavior of marine organisms that reproduce or grow particularly well only at certain temperatures. Even when it comes to flood prevention, data from climate models cannot provide accurate information about the risk of individual localities, but only provide averaged values for a larger region (depending on the size of the grid boxes within the model), which means that clouds, floods, droughts, or other events within a grid box can only be averaged or even not represented at all. It also happens that climate model output generally show temperatures that are too high or too low compared to observed time series (cf. Fig 1.3) or storms occur in a different region, but these originate from the dynamics within the model and thus cannot be adjusted by hand. Unknown or simplified processes, and the limited computing and storage capacities are also sources of error that must be optimized in the future.

If these deviations between observed and simulated data exist, procedures can be applied to minimize these errors. Due to the fact that these so-called bias corrections do not eliminate all errors, but can only reduce them to a minimum, the term “bias adjustment” is also valid and will be used for it in the further course of this thesis.

To perform a statistical transformation in the form of a bias adjustment on modeled climate data (A), observed (B) and modeled (C) data of a climate variable of the same time period are needed. Usually, these time series (B and C) cover a historical time period

(i.e. control period) on which the deviation between modeled (C) and observed (B) data can be determined. Since most adjustment techniques assume that the error between modeled and observed data remains persistent, i.e. the bias also exists in future periods (Maraun, 2012), this can be used to adjust modeled time series of the future (A), by taking into account the change between modeled historical (C) and future (B) periods, and thus minimizing the deviations discussed earlier. Of course, there are many other ways to determine and minimize the bias.

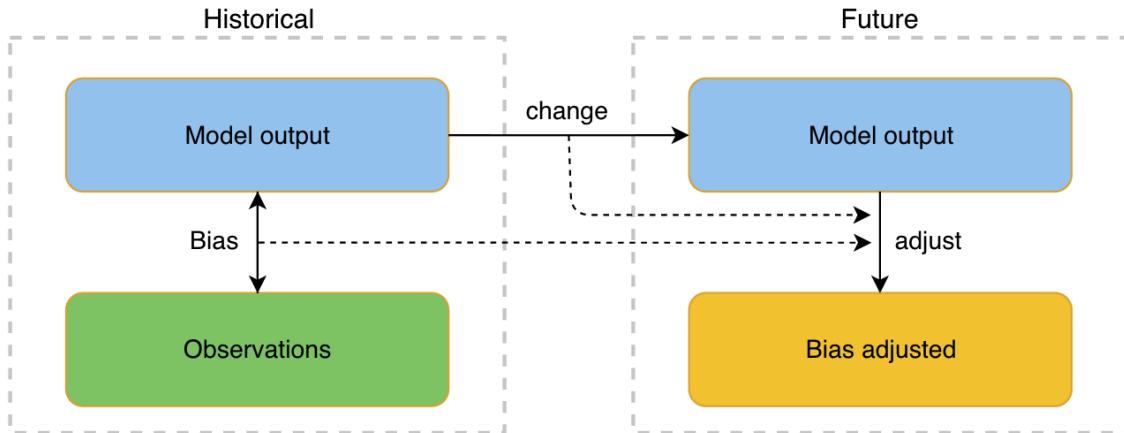


Figure 1.4.: Simplified representation of a general bias adjustment procedure

This presupposes the basic assumption that there is an existing relationship between the data of modeled climate scenarios and the real, actual climatic processes and conditions, and that topographic influences in observed as well as modeled time series are also related to each other.

In research, the term “statistical downscaling” is also used in connection with bias adjustments. In downscaling, large-scale grid cells of simulated data are interpolated to obtain a higher resolution. This can be used in combination with bias adjustments to interpolate modeled data from a low resolution to a higher resolution in order to subsequently minimize the bias. This enables to adjust modeled time series of future periods and at the same time increases the resolution of the data which can lead to making regionally more accurate statements about the climate of future time series, based on historical modeled and observed data.

To perform bias adjustments effectively, the quality of the observed data is particularly important, as these serve as the basis for the adjustments, and thus it is assumed that there is no error between them and the actual climate and weather that occurred. For meaningful results, long-term data sets are needed that represent variability and extremes well. Too long periods, on the other hand, can also lead to erroneous results, since the climate is also subject to change. The selection of the optimal adjustment procedure for the respective application can be varied, since different methods can lead to different results, depending on the period under consideration, the variable, the quality, and the resolution of the data.

It is also advisable to use simulated data from the same climate model for the historical

time series and for the time series to be adjusted. In this way the model's own trends and dynamics are better preserved instead of being adjusted. Furthermore, only then the change between the modeled data from the control and scenario period is comparable.

Insurance companies, farms, aid organizations, and governments depend on reliable data to prepare early for changing climatic conditions and extreme events, as well as the new challenges they bring, which include water shortages, rising temperatures, droughts, and storms. Recently, several storms and floods in Rhineland-Palatinate and North Rhine-Westphalia cost many lives and caused billions in damage (“Jahrhunderthochwasser 2021 in Deutschland”, 2021). Hamburg and other cities located near large rivers, dams, or in valleys also experience recurrent water and mud floods. If we leave Germany and turn our gaze towards Africa, we will see that extreme heat waves regularly occur there, resulting in drought and famine (“Why Africa's heatwaves are a forgotten impact of climate change”, 2020).

There are countless such examples, not only in Europe and Africa but all over the world. Researching the earth's climate, as well as going back to historical data, can help to better assess the climate of the future, classify different scenarios, to understand the influence of humans, the natural processes and dynamics of the earth's climate, and accordingly to point out ways in which we can enable a sustainable and livable existence on this planet.

1.3. About this Thesis

This thesis aims to demonstrate different bias adjustment techniques for climate data both mathematically and in the programming languages Python and C++. All procedures will be implemented and applied to evaluate the results of these adjustments. In doing so, different statistical methods, such as root mean square error, mean bias error and correlation, are applied to compare the bias-adjusted data sets with the underlying observational and modeled data to find out what influence the adjustment methods have had on the modeled data.

Air temperatures at 2m height are adjusted exclusively, but occasionally reference is made to the adjustment of precipitation. Temperatures are easier to adjust compared to precipitation since their natural variability and the complexity of the trends are much lower. Furthermore, precipitation occurs much more locally than temperatures, which is why temperatures offer medium validity for a larger range whereas precipitation, on the other hand, is not subject to seasonal trends in the vast majority of regions than temperatures.

Unless otherwise indicated, all illustrations are made by the author. The following table provides an overview of the topics and issues addressed in the individual chapters.

Section	Content
Abstract	What is this Thesis about?
Introduction	What are climate models and what means bias adjustment?
Motivation	What is interesting about bias adjustments?
Data	What data is used in this thesis?
Approach / Methods	Which methods are applied and how do they work?
Results	Do the methods improve the data and how do they differ?
Conclusion	Are there any special results?

Table 1.1.: Thesis overview and topics

At the beginning of this thesis, all procedures and calculations were implemented in the programming language Python. Due to the enormous performance differences, these were also implemented in C++. The C++ implementations run much more efficiently and can be executed on an ordinary PC. However, since the Python programming language is much easier to represent and understand and can be easily reproduced without much effort, the code excerpts, which are listed directly for each method, are presented in Python. In an extra section, the C++ implementations of the bias adjustment procedures are presented (Sec. 4.7). The compilation of the C++ implementations has been done with g++ (Apple clang version 13.1.6). It is also recommended to use version 17 of the standard library `std`. This can be defined during compilation with `-std=c++17`.

In addition, all calculations for the examination of the results were carried out with the help of a large number of C++ and Shell scripts. These are also not presented in this work, since the focus is the demonstration of the procedures, as well as the evaluation of these. Diagrams and visualizations, have been made using Python and various modules. All source code (bias adjustment implementations in C++, Python, and Shell; evaluation scripts in C++, Python, and Shell; diagrams, notes, and more in Python with the help of Jupyter Notebooks; LaTeX source code of this thesis) is submitted to the University of Bremerhaven with this document and is partially publicly versioned on the author's Github account.

Name	URL
Bias-Adjustment-Cpp	https://github.com/btschwertfeger/Bias-Adjustment-Cpp
Bias-Adjustment-Python	https://github.com/btschwertfeger/Bias-Adjustment-Python

Table 1.2.: Repositories containing the implementation of the adjustment procedures used in this thesis

All calculations and visualizations have been performed on a MacBook Pro 13" 2017 with the operating system macOS Monterey 12.4. This has an Intel Dual Core i5 processor with 2.3 GHz and 8 GB of RAM. Visual Studio Code was chosen as the development

environment, as this offers support for all programming languages and tools used, as well as enabling the integration of various Anaconda environments.

All adjustment techniques presented here are applied to the data sets described in Chapter 3. Explanations of the procedures, as well as the demonstration of the functionalities, are provided at the appropriate places with simple text and source code examples.

In all listings showing source code snippets of the Python programming language, it is assumed that the PyPi modules Numpy and Xarray are installed and imported with the abbreviation `np` resp. `xr`. These modules can be installed into the respective environment via the package manager Pip. It is recommended to use at least Python3.6+.

```
1 python3 -m pip install numpy xarray
```

Listing 1.1: Installation of required packages using the Python package manager Pip

Installed modules can be imported using the command `import <package name>`. If additionally `as <name>` is specified, all classes, functions, and attributes in the module will be stored in the specified variable, so that the desired functionalities can be accessed via it.

```
1 import numpy as np
2 import xarray as xr
```

Listing 1.2: Import required modules into a Python script

In the listings shown, care has been taken to display data types that do not have to be explicitly specified in the Python programming language. This can be helpful at appropriate points to understand the data structures used.

Furthermore, it is necessary to define some terms in order to avoid ambiguities in advance:

Phrase	Meaning
data set	In this thesis, a data set consists of a three-dimensional matrix, which represents a variable, such as air temperature, in the dimensions of time, longitude and latitude.
time series	A time series describes a one-dimensional vector, which represents a variable over a time dimension. For example, the time series of Bremerhaven corresponds to the temperatures for a certain longitude, and latitude index in a data set. When talking about time series in plural, multiple time series of a data set can be meant.
grid box / grid cell / cell	A grid box describes the spatial position and size in which a time series can be within a data set (cf. Fig. 1.1).

Table 1.3.: Definitions of terms and meanings

2. Motivation

Due to the research carried out in advance at the Alfred-Wegener Institute, which included the investigation and comparison of teleconnections in global and regional, as well as bias-adjusted data sets, it was noticed that the bias-adjusted time series not only differ from the raw model outputs in means, variances and further distributional properties, but that the applied adjustment also has an influence on the dependencies in the form of correlations of different regions with each other.

Since these analyses have been performed on data where it was not clear which adjustment method and parameters were applied, it will be a part of this thesis to show and implement different techniques. The goal is to perform these procedures for adjusting and minimizing deviations between modeled and observed data both mathematically and in the Python and C++ programming languages to answer the question of what influences the application of these methods have on the data.

There is the assumption that bias adjustment techniques, while adjusting the distributional properties of climate model-generated data sets to those of the observed time series on average, can artificially negatively affect regional and global dependencies. Since there is no public scientific work yet on the influence of bias adjustments on correlations between locations within the data, this topic is particularly important when it comes to mapping feedback events and remote effects, this shall be one of the central elements of the evaluation of the methods presented here.

In addition, this document is intended to serve as a kind of guide and reference to present various climate data adjusting techniques in a comprehensible way, as well as ways to implement them. The implementations of the methods developed during this thesis will also be made public so that they can be freely used for further research and development.

3. Climate Data

Climate data generated and made available by various institutions serve as the basis for the bias adjustment techniques and analyses presented here.

In the scientific context, climate data are mostly available in the file format “NetCDF” (Network Common Data Form). This self-describing data format has the file extension `nc` and, due to the binary data format, offers fast and efficient storage of multidimensional data sets and can also be read in many programming languages such as C, C++, Python, R, and Fortran.

This file format is mainly used in climate research because data such as atmospheric and oceanographic variables do not consist only of one-dimensional vectors of values. Most data sets consist of three to five dimensions. Time is usually the first dimension, followed by one or more dimensions for space like longitude, latitude, altitude, and depth. If these dimensions are dereferenced, a certain value at a certain time at a certain location can be obtained.

This can be imagined as a net of square grids (i.e. grid cells) spanning around the globe (cf. Fig. 1.1). Cells on the same latitude usually have the same spatial size. Since the circumference of the latitudes decreases above and below the equator, cells there cover a smaller area than those closer to the equator. This behavior can be different in models that uses triangular grid cells, but so far these are quite rare. Within a data set, the storage of climate variable values can be imagined as in multidimensional arrays:

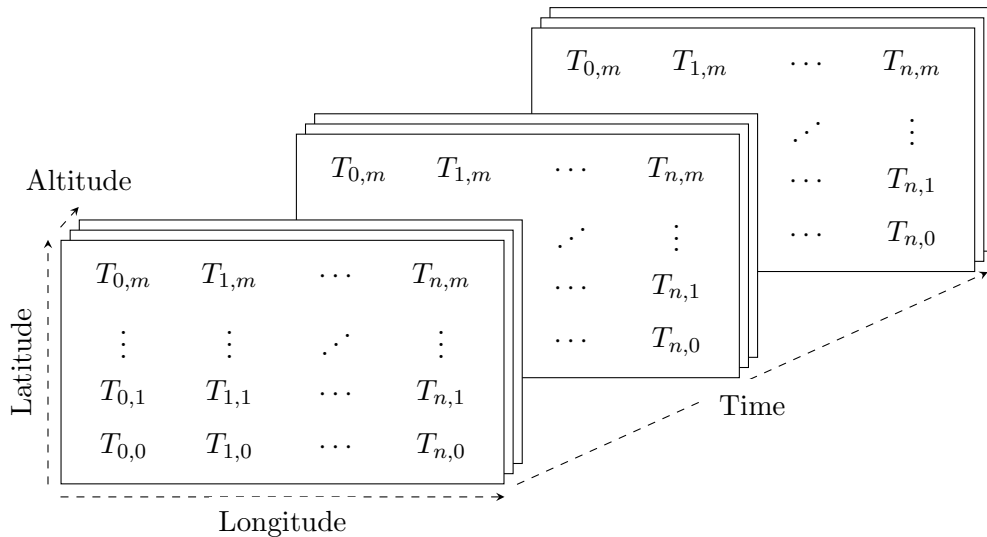


Figure 3.1.: Simplified representation of the dimension structure of a four dimensional NetCDF data set containing temperature values; Dimensions: Time (without subscript), Longitude, Latitude and Altitude (without subscript)

(Fig. 3.2) A NetCDF file consists of various components that are related to each other. These include the previously mentioned dimensions that represent the variables contained in the data set (i.e. climate variables such as temperature and precipitation). These

variables have attributes that contain a description on the one hand, but also the values stored in them. Since the NetCDF4 standard, groups and user-defined data types have been added (marked red). These allow an even more flexible manipulation of the data so that they can be adapted to the corresponding information depending on the application and research field. For example, grouping makes it possible to combine information within a data set into groups without splitting them over several files. Also, the coordinate system can be customized by not taking the curvature of the earth’s surface into account or by taking it into account in a different way when manipulating the data.

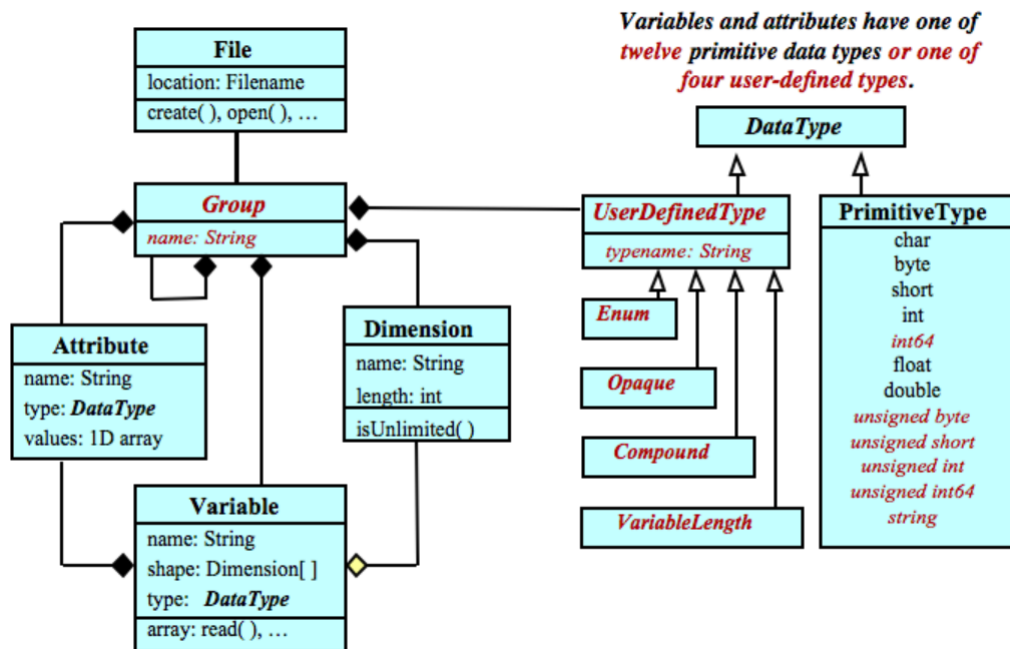


Figure 3.2.: The NetCDF data model (Figure by NetCDF-C Documentation, 2022)

Furthermore, this file format makes it possible to apply multiple write and read processes to one file at the same time, thus ensuring dynamic retrieval and storage of information by distributed systems. Above all, the system independence, which enables scientists and other users to exchange findings and data among each other without having to pay attention to the respective system environment, makes this scalable data type particularly attractive for research.

To display a NetCDF file in ASCII format in a readable way, the command line tool `ncdump` can be used. This requires the installation of the NetCDF library for the command line (cf. “GeoServer 2.20.x User Manual / Installing required NetCDF-4 Native libraries”, 2022). The information thus output shows in the first part the dimensions and variables with their attributes, followed by metadata dealing with the origin, the survey, the history, and referencing information. Additional information can be added to the metadata as desired. In the final data part, all data values of all climate variables in the dataset are listed.

An example of the output of the `ncdump` command applied to the NetCDF file `tas_day_noaa_reanalysis_v3_eur_19810101-20101231_noleap.nc` provides the following listing:


```

1 netcdf tas_day_NOAA_Reanalysis_v3_EUR_19810101-20101231_noLeap {
2     dimensions:
3         time = 10950; lat = 50; lon = 111;
4     variables:
5         float tas(time, lat, lon); tas:units = "degC";
6             tas:standard_name = "air_temperature";
7             tas:long_name = "daily mean 3-hourly Air Temperature at 2 m";
8             ...
9             tas:GridType = "Cylindrical Equidistant Projection Grid";
10        double time(time);
11            time:standard_name = "time"; time:long_name = "Time"
12            time:calendar = "standard";
13        float lon(lon) ;
14            lon:standard_name = "longitude"; lon:long_name = "Longitude"
15            lon:units = "degrees_east";
16        float lat(lat) ;
17            lat:standard_name = "latitude"; lat:long_name = "Latitude";
18            lat:units = "degrees_north";
19        // global attributes:
20        :CDI = "Climate Data Interface version 1.9.6 (http://mpimet.mpg.de/cdi)" ;
21        :history = "Thu Feb 24 10:25:38 2022: cdo -f nc -sellonlatbox
22        ,-45.5,65.5,23,72.5 -selvar,tas tas_day_NOAA_Reanalysis_v3_19810101
23        -20101231.nc tas_day_EUR_NOAA_Reanalysis_v3_19810101-20101231.nc\nThu Feb
24        24 10:11:27 2022: ... ;
25        :title = "mean Daily NOAA/CIRES/DOE 20th Century Reanalysis V3" ;
26        :spatial_resolution = "1.0 degree";
27        :product = "reanalysis"; ...;
28        data:
29            tas = 23.43435, 23.33434, 23.23434, 23.14684, 23.05935, 22.94683,
30            22.80935,22.69686, 22.59683, 22.48434, 22.38433, ...
31    }

```

Listing 3.1: Output of `ncdump` command applied on a NetCDF data set; three dots are symbolic for more rows of information

The data used in this thesis uses three dimensions. These include the time, longitude, and latitude to represent the variable “tas” (i.e. temperature) for the region of Europe.

As defined in section 1.2, observed and modeled data sets of different time spans are needed for the application of bias adjustment procedures. Subsequent sections describe the data sets used here, as well as their preparation.

3.1. Modeled Data

The modeled data used in this thesis was generated within the CMIP6 research project mentioned in the introduction. The climate model that produced this data is called MPI-ESM1-2-HR, was published in 2017 and executed by the Max-Planck Institute for Meteorology in Hamburg, Germany (cf. Mauritsen et al., 2019). This model is not only designed to answer fundamental and advanced scientific questions by providing a global and realistic representation of the earth’s major and partly minor physical and biochemical

processes and trends, but it also improved results in atmospheric and oceanic dynamics due to reduced biases due to its higher resolution and more realistic implementations of processes compared to previous versions.

This earth system model couples an atmospheric general circulation model with an ocean and sea ice circulation model and also simulates the biochemical interactions between the ocean, atmosphere, land, and also vegetation. This ESM is one of the best-known models whose outputs are used for a variety of climate studies and publications. The name of the model is composed of the name of the Max-Planck Institute (MPI), the type of model (ESM, i.e. Earth System Model) and the version number. HR means that it is the high-resolution variant.

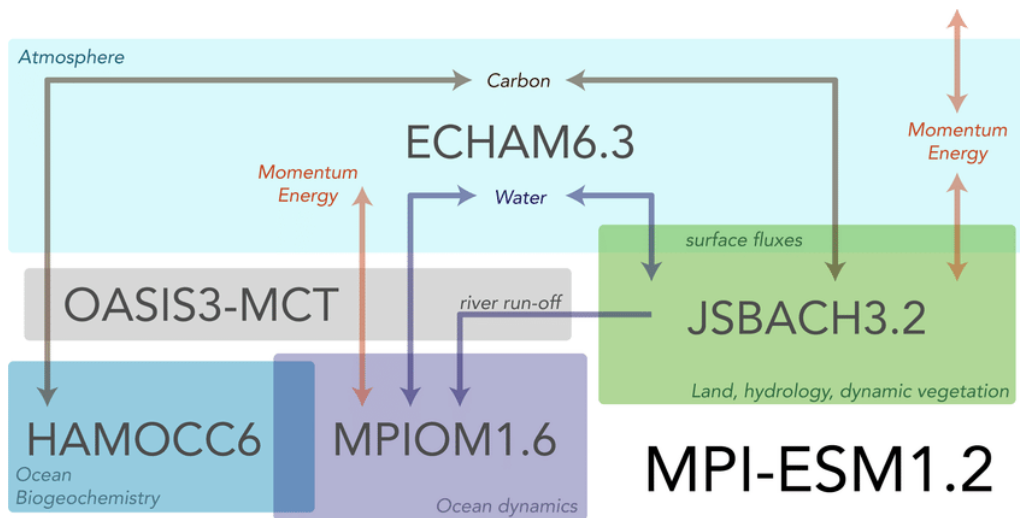


Figure 3.3.: Schematic overview of the components of the MPI-ESM1-2-HR (Figure by Mauritsen et al., 2019)

(Fig. 3.3) the MPI-ESM1-2-HR consists of four components. ECHAM6.3 is an atmospheric general circulation model, which is used for the simulation of the atmosphere and its thermodynamics. This GCM is coupled with the land surface model JSBACH3.2. JSBACH is responsible for the modeling and calculation of the processes on the land, including the vegetation living there. Current behavior, dynamics of the seas and oceans as well as the processing of individual hydrological signals of the JSBACH3.2 are processed by the MPIOM1.6, which is also directly coupled with the HAMOCC6, which is specialized in the biochemical processes of the seas and creatures living there.

OASIS3-MCT is the software used to realize the coupling of these models and offers many interfaces for parameter transfer between the components. For example, water from the MPIOM can be transferred to the atmospheric model ECHAM by condensation. Also, the release of CO_2 , for example by volcanic activities in JSBACH, can be passed on to the atmosphere. In the other direction, plants in JSBACH can bind CO_2 from the atmospheric component. All these components were developed at the Max-Planck Institute. Further information on this model, as well as on the description and configuration of the components can be found in the article by Mauritsen et al., 2019.

The 2m air temperature data set generated by this model has a resolution of 384 x 192 longitude/latitude in the global grid. Thus, it has $384 \cdot 192 = 73728$ cells distributed around the entire globe as shown in figure 1.1. Since this thesis focuses on air temperatures at 2m height, the data set used here has no vertical dimensions. In addition, these temperatures were calculated by the ECHAM6.3 component, which is an atmospheric general circulation model as a component of the MPI-ESM1-2-HR.

There are also variants of the MPI-ESM that can generate data in higher or lower resolution. The MPI-ESM1-2-LR, for example, has a resolution of 192 x 96 longitude/latitude and thus has larger grid boxes. In addition, unlike the high-resolution variant, this one has dynamic vegetation, as well as carbon and nitrogen cycles (Mauritsen et al., 2019, “CMIP6: The Coupled Model Intercomparison Project”, 2020).

In this thesis, the high-resolution variant (MPI-ESM1-2-HR) was chosen, because neither the vegetation nor the carbon cycle is investigated here. Furthermore, a high resolution is better suited to minimize errors that may occur in advance due to insufficient grid size, for example in impassable areas. The high-resolution variant also has a similar resolution as the observation data presented subsequently.

Since regions and thus also grid cells are to be compared, it is necessary that all data sets used here have the same resolution. This would not be the case if statistical downscaling were to be performed, but this is usually done with data from regional climate models since these usually have a lower resolution than GCMs and ESMs by themselves.

Since the grid of the MPI-ESM1-2-HR has a resolution of 384 x 192 longitude/latitude, but the observation/reanalysis data presented later have a resolution of 360 x 181 longitude/latitude, it is necessary to make an adjustment by scaling the grid of the ESM to that of the reanalysis data (cf. Lst. 3.2). This has the disadvantage that by the scaling artificial deviations are produced, which do not originate from the model. Thus, values of the original data may not exactly match the values of the same geographic coordinates in the scaled data, because of interpolation at certain points (i.e. statistical upscaling). However, since there are currently no publicly available observational data with the same resolution as those of the 2m air temperature output of the MPI-ESM1-2-HR, scaling is the only way to produce comparable data here.

The following table lists the parameters that can be used to find and download the ESM data used in this thesis at <https://esgf-data.dkrz.de/search/cmip6-dkrz/>.

Search parameter	Value
MIP ERA	CMIP6
Source ID	MPI-ESM1-2-HR
Experiment ID	historical
Variant Label	r1i1p1f1
Frequency	day
Realm	atmos
Variable	tas

Table 3.1.: Earth system model data set specifications for DKRZ file search

In order to perform and subsequently evaluate bias adjustments, two periods of simulated data are used. The historical control period covers the years 1951 to 1980 inclusive, and the period to be bias-adjusted, referred to as the scenario, future, or adjusted time series, covers the years 1981 to 2010 inclusive.

Climate models do not produce usable and realistic data immediately after startup, because various physical and biochemical processes must first be in so-called equilibrium, which in climate research usually corresponds to the preindustrial state of before 1850. This is achieved by running a model with its initial parameters until it reaches this state. This so-called spin-up process ensures that the same starting conditions always exist for different simulations. In addition, it can be evaluated whether a model can correctly represent the different processes, and it also ensures the generated data are no longer dependent on the initial parameters (Mauritsen et al., 2019).

Since the MPI-ESM1-2-HR model output of the historical simulations already start with the equilibrium state at year 1850, errors due to the run-off period of the climate model can therefore be excluded.

3.2. Observational / Reanalysis Data

Observations are always subjective. So it can happen that gauges and measuring instruments, which are only a few meters apart from each other, record different values for air pressure, temperature and other climate parameters. Also due to the replacement of devices, which have different sensitivities, the same quality of the measured data cannot be guaranteed throughout. These inaccuracies are supported by the patchy distribution of the measuring stations, whose density within urban areas is higher than towards the sea.

The development of satellite-based information systems, which investigate the processes and properties in different atmospheric layers, is constantly being improved, so that more exact data and higher accuracies in the measurement can be achieved here. However, this also leads to the fact that the previously generated and observed data are qualitatively subject and even the improved systems are, however, not able to completely map the planet and its climate variables, not only spatially, but also over a longer period of time.

With observed data from measurements alone, it is not yet possible to create an exact grid for observed climate data that maps the actual values for temperatures, CO₂ concentration, air pressure, and more for each time at each location.

Atmospheric climate reanalysis is a very popular method to process these observed climate data for science and research. Here, historical time series of different atmospheric parameters from observations are combined with simulations to reconstruct missing data points of the observations, to estimate existing ones and to re-evaluate them if necessary.

This is done by using the observation data for each time step as input to a dynamic numerical weather prediction model (NWP), which is then designed to reproduce the next time step of the observed data as accurately as possible. NWPs are not such comprehensive as

climate models like ESMs, even though they also take into account temperatures, precipitation, water vapor and other climate parameters. Particularly small cloud formations, interactions of solar radiation and other processes on small spatial scales cannot be accurately represented and are therefore approximated (Deutscher Wetterdienst, 2022). Data assimilation in this way is also called the application of the “Ensemble Kalman Filter” (Katzfuss et al., 2016).

In contrast to the interpolation of missing values, this procedure allows to preserve the dynamics of the climate by not transforming them linearly or cubically, but by adjusting them on the basis of physical laws (Thejll and Gleisner, 2015). By this procedure not only the missing data points are assigned values, but also the already known observed values get the new predicted values, which have only a minimal deviation to the observed time step. This ensures that all data points of a time step are generated with the same forcings and are therefore in relation to each other.

Reanalysis is used to create complete and consistent data sets for various climate variables to be able to create a picture of the past weather and climate as close to reality as possible. The quality of the observed data is important not only for bias adjustments but also for reanalysis since these are also used as the starting point for further procedures.

With the development of new and more efficient systems, the intervals of time steps, which are re-evaluated by reanalysis, are getting smaller and smaller. Not only are reanalysis data sets now available for monthly or daily averages values, but projects such as ERA-Land (European Reanalysis for Land Surface) even provide hourly values for a period from 1950 to the present for a variety of land variables with a resolution of $0.1^\circ \times 0.1^\circ$ longitude/latitude (i.e. a native resolution of 9km at the equator, “ERA5-Land hourly data from 1950 to present”, 2022).

The enormous resolution of the ERA-Land data set makes it particularly interesting for regional studies. However, since a very high resolution also requires a lot of computing capacity for transformations in the form of bias adjustments and the evaluation thereafter, but the goal of this thesis is to perform and study the effects of the procedures on ESM data, it was decided to use climate data from another source, which has a lower resolution. The ERA-Land data would be particularly useful if the bias in data from a regional climate model needs to be adjusted to perform local studies.

The reanalysis data for this thesis come from the NOAA-CIRES-DOE “The Twentieth Century Reanalysis Project”, 2022, which is currently in its third phase (V3). It provides various parameters for 3-hourly, daily average, and monthly average values for a period from 1836 to 2015 inclusive (“NOAA-CIRES-DOE Twentieth Century Reanalysis (V3): Summary”, 2022). For this thesis, the daily mean values of the temperatures at 2m height are used. Like the data of the ESM, these data are also divided into two parts, so that the years 1950 up to and including 1980 represent the historical control period and the years 1981 up to and including 2010 are used to evaluate the results of the adjustment procedures. These data have 381×180 grid boxes and thus provide a resolution of $1.0^\circ \times 1.0^\circ$ longitude/latitude.

Since the data of the reanalysis, or in the following called observations, are the basis of the procedures and analyses carried out in this thesis, the scaling of the grid cells mentioned in Section 3.1 is only applied to the 2m air temperature data sets of the MPI-ESM1-2-HR.

3.3. Data Preprocessing

For scaling the ESM data, the command line tool `cdo` (“Climate Data Operators”, 2022) is used. This is developed and provided by the Max-Planck Institute for Meteorology. It offers an extensive collection of methods for manipulation and selection of climate data in different formats and is based on simple to complex in- and output commands, can accept files and streams, as well as output and write them. Climate Data Operators commands are executed from right to left, unlike many other command-line tools. This means that the command closest to the `cdo` command is executed last. The syntax is as follows:

```
cdo <key2>,<value> <key1>,<value1>,<value2> input output
```

If more than one argument is passed to `cdo`, the keys of the key-value pairs are to be marked with a dash (“-”). The vast majority of values assigned to the keys of the arguments are separated from the keys by a comma. The assignment of several values to one key can be realized by several commas. Also, ranges can be defined with the “/” sign.

```

1   cdo -f nc \ # force nc output
2   -remapcon,tas_NOAA_1951-2015.nc \ # scale grid of input file
3   -subc,273.15 \ # change deg. Kelvin to deg. Celsius
4   -selyear,1951/2010 \ # select timespan 1951 - 2010
5   tas_MPI-ESM1-2-HR_1850-2014.nc \ # input file
6   tas_MPI-ESM1-2-HR_1951-2010.nc # output file
```

Listing 3.2: Apply observational grid configuration to the modeled data set, change the unit of temperatures from Kelvin to deg. Celsius and select desired timespan using `cdo`

(Lst. 3.2) `remapcon` is the command that can be used to transfer the grid configurations, which include distances between longitudes and latitudes, surface curvature, and the number of cells, from the observed to the modeled data set by cubic interpolation (cf. Schulzweida, 2021 and “Hybrid cubic spherical Bernstein-Bézier patch interpolation”, 2022). In addition, the time span is narrowed down to the years 1951 to 2010 and the unit of measurement of temperatures is changed from Kelvin to deg. Celsius by subtracting all values with 273.15.

Then the region is narrowed down to Europe so that the data sets only have values for the longitudes from -45°W to 65°E by 1 deg. East and latitudes from 23°N to 72°N by 1 deg. North (cf. Lst. 3.3).

```

1   cdo -f nc \ # force nc output
2   -sellonlatbox,-45,65,23,72 \ # slice space to -45°W - 65°E, 23°N - 72°N
```

```

3   tas_NOAA_1951-2010.nc      \ # input file
4   tas_NOAA_EUR_1951-2010.nc \ # output file

```

Listing 3.3: Select European region using cdo

Finally, the modeled time series, as well as those of the observations, are divided into two data sets, each covering the periods 1951 to 1980 and 1981 to 2010. Thus, all data sets represent a period of 30 years and each has 10950 daily mean 2m air temperature values for 111 x 50 grid cells (i.e. 111 longitudes and 50 latitudes). There are no missing values.

```

1   cdo -f nc                \ # force nc output
2   -selyear,1951/1981       \ # select timespan 1951 - 1981
3   -selvar,tas              \ # select only temperature values
4   tas_NOAA_1951-2010.nc    \ # input file
5   tas_NOAA_EUR_1951-1981.nc \ # output file

```

Listing 3.4: Select specific timespan of observational data using cdo

It should also be noted that days that only occur in leap years have been removed from the observed and modeled data sets. This is a common method to avoid sources of error in advance.

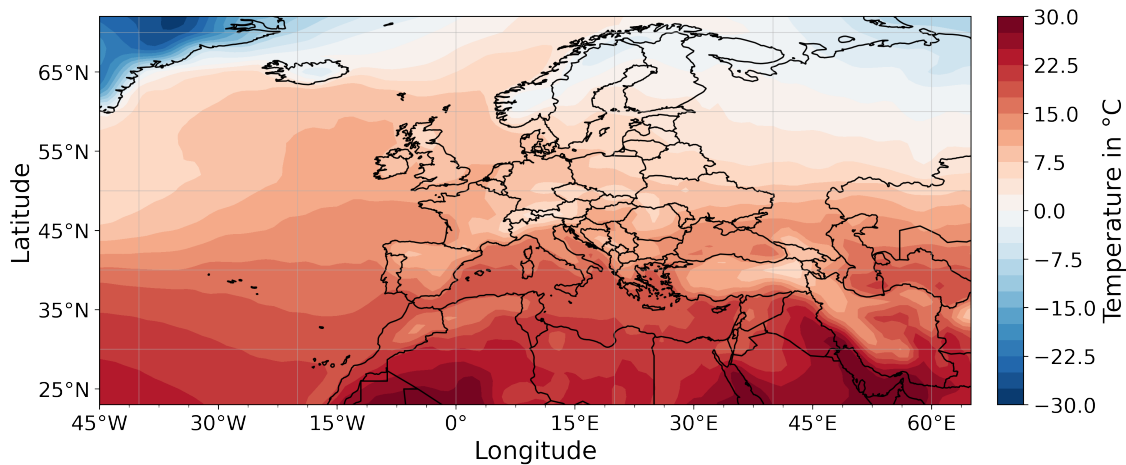
**Figure 3.4.:** MPI-ESM1-2-HR model output; mean of daily 2m air temperatures in Europe between 1981 and 2010; -45°W to 65°E by 1 deg. East, 23°N to 72°N by 1 deg. North

Figure 3.4 represents the mean 2m air temperature per grid cell of the modeled data for the years 1981 up to and including 2010. This is intended to provide a first overview of the selected region. This is also the data set, which will be adjusted in the further course of the thesis using various methods. This data has been interpolated to produce better visual contours instead of $111 \cdot 50 = 5550$ grid boxes.

A somewhat larger section was chosen, which includes not only the countries of Europe but also the north of Africa, as well as regions in eastern Europe and a part of Greenland. This is due to the fact that a larger spectrum of regions with different climatic conditions can be considered. Therefore, when Europe is referred to in this thesis, the section shown in figure 3.4 is meant.

4. Approach / Methods

In the following, five different bias adjustment techniques are presented mathematically, as well as in the Python and C++ programming languages, and illustrated with simple examples. Bias adjustments can be classified into scaling- and distribution-based methods. Starting with the scaling-based methods such as the delta method (DM), as well as the linear (LS) and variance scaling (VS) method, two more complex, distribution-based techniques, named quantile mapping (QM) and quantile delta mapping (QDM), are then shown. At the end of this chapter, there is a detailed implementation with explanations for the execution of selected methods in Python, so that all presented methods can be taken over and applied directly. In addition, implementation examples and pseudocode for executing the presented methods in C++ are included in Section 4.7.

The mathematical basis of the scaling-based bias adjustment procedures is derived from the articles by Teutschbein and Seibert, 2012 and Beyer et al., 2020. For the mathematical representation of the QM and QDM procedures, the description of Tong et al., 2021, as well as Cannon et al., 2015 was followed.

Table 4.1 records symbols and abbreviations used in the equations presented in this thesis.

Symbol	Meaning	Symbol	Meaning
T	Temperature	*	Bias adjusted
Pr	Precipitation	$\{\{1..n\}\}$	Intermediate result (e.g. $T_{sim,p}^{VS(1)}$)
f	Transformation function	obs	Observational data (NOAA)
F	Cumulative distribution function	sim	Modeled data (MPI-ESM1-2-HR)
P	Probability density function	h	Historical, control period (1951-1980)
DM	Delta Method	p	Future, predicted time period (1981-2010)
LS	Linear Scaling	m	Monthly interval
VS	Variance Scaling	i	Index in 1D sequence
QM	Quantile Mapping	μ	Mean
QDM	Quantile Delta Mapping	σ	Standard deviation
		σ^2	Variance

Table 4.1.: Mathematical symbols, abbreviations and sub-/superscripts

Bias adjustments are statistical mathematical transformations used to adjust climate data. For example, the adjustment of a modeled temperature time series representing a future climate scenario can be represented as $T_{sim,p}^* = f(T_{sim,p})$, where f is an arbitrary adjustment procedure that aims to minimize deviations between $T_{sim,p}$ and $T_{obs,p}$. This transformation function is also passed the values of $T_{sim,h}$ and $T_{obs,h}$ in all methods shown here. Thus, depending on the method, deviations in the mean, as well as in the variance and in the quantiles can be minimized.

The period from 1951 to 1980 is used for the historical control period and is identified with $T_{obs,h}$ for observed and $T_{sim,h}$ for modeled data, respectively. The years from 1981 to 2010 inclusive comprise the so-called future or scenario period, which is identified with $T_{obs,p}$

for observed and $T_{sim,p}$ for modeled data, respectively. These time series are respectively a one-dimensional sequence of temperatures for a location or a grid cell as shown in Figure 1.1, in the respective data set.

To determine a certain temperature value within this sequence, the index i ($i \in \{0, 10949\} \in \mathbb{N}$) is used in the equations presented here. The number 10949 corresponds to the last index, which at the same time means that there are 10950 entries in each sequence where a daily temperature value is stored. This number results from the number of days between 1951 to 1980 or 1981 to 2010, whereby days, which occur only in leap years, are not considered.

For the mathematical demonstration of the procedures, the temperature value of December 17, 1997 is always adapted for the location Bremerhaven. This day corresponds to index 6190, which can be accessed via the subscript i .

Methods to adjust and minimize deviations between observed and modeled climate data are applied in various ways. For example, the scaling-based methods presented here, are applied based on long-term monthly periods. This means that equal months are adjusted together by dividing the modeled time series, as well as those of the observations, into 12 groups, one for each month. Thus, all Januaries of a data set can be adjusted separately from all other months. Of course, this also applies to February, March through December. By considering grouped months over the entire period, one has a more extensive contingent of month-dependent climate events, which can better represent the climate, its variability, and extremes over a longer period of time than if each month of each year were adjusted separately or all together.

Mathematically, this aggregation of equal months, for 30 years, can be expressed with the following equation, provided that the day at index 0 corresponds to the first of January and each year has exactly 365 days (i.e. days in leap years are ignored).

$$m(T_{sim,p}(i)) = \{T_{sim,p}(x) \mid \forall x \in \{0, 10949\} \subseteq \mathbb{N}, x \text{ has same month name as } i\} \quad (4.1)$$

$$\implies m(T_{sim,p}(0)) \equiv m(T_{sim,p}(30)) \equiv m(T_{sim,p}(10615)) \quad (4.2)$$

Different variables are used in the source code excerpts shown. These have been designed uniformly so that the approach of the methods can be easily reflected in the source code excerpts based on the mathematical equations. The following table serves to define these uniform variable names.

Symbol	Meaning	Symbol	Meaning
obs	$T_{obs,h}$ as 1D Sequence	m_obs	$m(T_{obs,h}(i))$ as 1D Sequence
simh	$T_{sim,h}$ as 1D Sequence	m_simh	$m(T_{sim,h}(i))$ as 1D Sequence
simp	$T_{sim,p}$ as 1D Sequence	m_simp	$m(T_{sim,p}(i))$ as 1D Sequence
>	Output		

Table 4.2.: Symbols and abbreviations used in listings

The source code excerpts shown to illustrate the Python implementations for the scaling-based procedures assume that the respective commands are applied to 1D sequences (e.g. 1D list, 1D `numpy.array` or 1D `xarray.core.dataarray.DataArray`), which have been separated by month as described before. This is needed, for example, when calculating monthly means or standard deviations. That means, `m_obs`, `m_simh` and `m_simp` in these listings each represent one month over 30 years (i.e. `m_simp`, if it is the month of January, includes $30 \cdot 31 = 930$ entries; daily values of each January in the period from 1981 to 2010 inclusive; cf. Tab. 4.2f.).

i	Index in $_m(T_{sim,p}(i))$	Value	Date
0	0	-0.677857	January, 1. 1981
1	1	-1.611131	January, 2. 1981
2	2	-1.641309	January, 3. 1981
\vdots	\vdots	\vdots	\vdots
10614	928	3.495331	January, 30. 2010
10615	929	2.792645	January, 31. 2010

Table 4.3.: Symbolic example and description of the 1D sequence of temperatures of all Januaries in $T_{sim,p}$; values are 2m air temperatures in deg. Celsius

Since this step is used in several procedures, the separation of months in Python is implemented and explained in Section 4.6 with a detailed example. The monthly separation and adjustment using C++ and Shell is shown in Section 4.7.

Various tests in the course of this work have shown that this separation of the months leads to unrealistic results using the quantile and quantile delta mapping procedures. It has been shown that the first and the last day of the adjusted months give on average a much too high deviation from the last day of the previous month and from the first day of the following month, respectively because the distributions of the individual months, as well as their quantiles are no longer in relation to each other. For this reason, the QM and QDM procedures are applied directly to the full periods.

4.1. The Delta Method

The delta method (DM) basically consists of adding an observed daily historical value at a certain index with the difference of the long-term monthly means between the modeled predicted and historical values of the corresponding month. Thus, the change between the historical and predicted model output can be determined.

$$T_{sim,p}^{*DM}(i) = T_{obs,h}(i) + (\mu_m(T_{sim,p}(i)) - \mu_m(T_{sim,h}(i))) \quad (4.3)$$

The delta method assumes that the bias between modeled and observed data remains consistent over time, so the calculated change in modeled data can be summed with the value of the historical observation to estimate the future value (Beyer et al., 2020).

In terms of the data presented in Chapter 3, this means that the bias of the entire time series of one grid cell in the modeled data from 1981 to 2010, inclusive, can be adjusted by starting with the first day, January 1, 1981, and determining the monthly mean over all Januaries in this data set, as well as in the modeled data of the control period from 1951 to 1980. The difference of these two values is the change between the historical and predicted modeled data and is then added to the historically observed value for January 1, 1951. Thus, the first day is adjusted at index $i = 0$. This is repeated for all further days in the time series to be adjusted, taking care to calculate the mean corresponding to the month in which the day to be adjusted lies over all years (i.e. long-term monthly mean).

Example

The procedure of the delta method is demonstrated below with the adjustment of the author's birthday:

$$T_{sim,p}^{*DM}(i) = T_{obs,h}(i) + (\mu_m(T_{sim,p}(i)) - \mu_m(T_{sim,h}(i))) \quad (4.4)$$

$$= 2.93^\circ\text{C} + (2.44^\circ\text{C} - 3.0^\circ\text{C}) \quad (4.5)$$

$$= 2.93^\circ\text{C} + (-0.56^\circ\text{C}) \quad (4.6)$$

$$= 2.37^\circ\text{C} \quad (4.7)$$

where:

$$i = 6190 \text{ (index of December, 17. 1997 in predicted period)}$$

$$\mu_m(T_{sim,h}(i)) = \text{Mean of all December temperatures in } T_{sim,h}$$

$$\mu_m(T_{sim,p}(i)) = \text{Mean of all December temperatures in } T_{sim,p}$$

The temperature adjusted by the delta change method for December 17, 1997 is 0.56°C colder than the model simulated it for that day. Here, the mean of all temperatures in the December months between 1951 and 1980 was subtracted from the mean of all temperatures in the December months between 1981 and 2010 and added with the historical observed temperature value according to the index $i = 6190$. This means that the modeled

data for the Bremerhaven region is on average 0.56°C warmer in the December months between 1981 and 2010 than in the time series from 1951 to 1980.

Implementation

The delta method is shown in the following example using the Python programming language. Here, line 5 shows the procedure of Equation 4.3, but note that `m_obs`, `m_simh` and `m_simp` are 1D sequences of a month over all years (cf. Tab. 4.2). The time series can be adjusted by subtracting the mean of the historical modeled data from the mean of the predicted time series, as previously shown mathematically, and then adding it to the values of the observed 1D sequence:

```

1  def delta_method(
2      m_obs: [float], m_simh: [float], m_simp: [float], kind: str='+'
3  ) -> [float]:
4      if kind == '+':
5          return m_obs + np.nanmean(m_simp) - np.nanmean(m_simh) # Eq. 4.3
6      elif kind == '*':
7          return m_obs * (np.nanmean(m_simp) / np.nanmean(m_simh)) # Eq. 4.8
8
9  dm_corrected = delta_method(m_obs, m_simh, m_simp) # cf. Tab. 4.2

```

Listing 4.1: Implementation of the additive and multiplicative delta method

The additive delta method has the disadvantage that it can give unrealistic results when applied to variables such as precipitation or wind speed, since negative values for $T_{sim,p}^{*DM}(i)$ can be calculated. For this reason, the multiplicative procedure of the delta method can be applied, which consists of multiplying the daily observed temperature values by the ratio of predicted and historical modeled long-term monthly means.

$$Pr_{sim,p}^{*DM}(i) = Pr_{obs,h}(i) \cdot \left[\frac{\mu_m(Pr_{sim,p}(i))}{\mu_m(Pr_{sim,h}(i))} \right] \quad (4.8)$$

Equation 4.8 is implemented in Listing 4.1l. 7 by multiplying the relative change between the historical and predicted modeled long-term monthly means by the historical observed value, as described previously. The grouping of months is not shown because this is addressed in Section 4.6 as mentioned previously.

4.2. The Linear Scaling Method

The linear scaling method (LS) is one of the basic statistical methods for adjusting time series climate data. Like the delta method, the linear scaling method is used to approximate model-generated data to that of observed data on a monthly average basis. It also has additive and multiplicative applications. The additive is much better suited for climate parameters such as temperatures since small differences in multiplication are too significant for temperatures. The application of the multiplicative method is more suitable for variables such as precipitation, wind speed, and vapor pressure.

(Eq. 4.9) To adjust daily temperature data of a predicted time series using the linear scaling procedure, the long-term monthly mean of the historical modeled data is subtracted from the historical long-term monthly mean of the observed time series. Then this value is added to the value of the day to be adjusted. Again, the long-term monthly means are to be determined according to the month of the day to be adjusted.

$$T_{sim,p}^{*LS}(i) = T_{sim,p}(i) + \mu_m(T_{obs,h}(i)) - \mu_m(T_{sim,h}(i)) \quad (4.9)$$

(Eq. 4.10) The adjustment of a modeled time series can also take place by applying the linear scaling procedure with observed and modeled data of the same period. This makes sense if the data adjusted in this way serve as input for further model simulations or equations.

$$T_{sim,h}^{*LS}(i) = T_{sim,h}(i) + \mu_m(T_{obs,h}(i)) - \mu_m(T_{sim,h}(i)) \quad (4.10)$$

The linear scaling procedure is particularly well suited to adjust bias between historical long-term monthly means by adding the difference to the value to be adjusted. This is shown by the fact that the long-term monthly mean of the adjusted time series $T_{sim,h}^{*LS}$, matches the long-term monthly mean of the observed time series in the control period: $\mu_m(T_{sim,h}^{*LS}(i)) = \mu_m(T_{obs,h}(i))$ (Teutschbein and Seibert, 2012).

In the multiplicative variant, a factor based on the ratio between the long-term monthly mean of historical observations and the modeled time series is multiplied by the corresponding value of the time series to be adjusted. This is not used for temperature data but is presented below for completeness.

$$Pr_{sim,p}^{*LS}(i) = Pr_{sim,p}(i) \cdot \left[\frac{\mu_m(Pr_{obs,h}(i))}{\mu_m(Pr_{sim,h}(i))} \right] \quad (4.11)$$

This also makes sense, as with the delta method, for variables whose value range is unilaterally limited and this is also the resting state (e.g. default precipitation equals zero and cannot be less than zero).

Example

The following equations serve as an example of the application of the additive linear scaling procedure, where the temperature of 4.34°C predicted by the model is adjusted

using Equation 4.9:

$$T_{sim,p}^{*LS}(i) = T_{sim,p}(i) + (\mu_m(T_{obs,h}(i)) - \mu_m(T_{sim,h}(i))) \quad (4.12)$$

$$= 4.34^\circ\text{C} + (3.48^\circ\text{C} - 3.0^\circ\text{C}) \quad (4.13)$$

$$= 4.34^\circ\text{C} + 0.48^\circ\text{C} \quad (4.14)$$

$$= 4.82^\circ\text{C} \quad (4.15)$$

where:

$$i = 6190 \text{ (index of December, 17. 1997 in predicted period)}$$

$$\mu_m(T_{obs,h}(i)) = \text{Mean of all December temperatures in } T_{obs,h}$$

$$\mu_m(T_{sim,h}(i)) = \text{Mean of all December temperatures in } T_{sim,h}$$

After applying the linear scaling procedure, the temperature on December 17, 1997 is 0.48°C warmer, because the observed temperatures, averaged over all December months in the historical control period, were warmer by exactly this value than the model simulated it for the same period. This also means that every day in all December months in Bremerhaven between 1981 and 2010 is 0.48°C warmer after applying the linear scaling method with these time series.

Implementation

In Python, the additive, as well as multiplicative linear scaling method can be implemented as follows:

```

1  def linear_scaling(
2      m_obs: [float], m_simh: [float], m_simp: [float], kind: str='+'
3  ) -> [float]:
4      if kind == '+':
5          return m_simp + np.nanmean(m_obs) - np.nanmean(m_simh) # Eq. 4.9
6      elif kind == '*':
7          return m_simp * np.nanmean(m_obs) / np.nanmean(m_simh) # Eq. 4.11
8
9  ls_corrected = linear_scaling(m_obs, m_simh, m_simp) # cf. Tab. 4.2

```

Listing 4.2: Implementation of the additive and multiplicative linear scaling method

Again, it is a requirement that the values of `m_obs`, `m_simh`, and `m_simp` have been separated into monthly groups (cf. Tab 4.2). The implementation of this grouping is done in Section 4.6, Listing 4.15ff..

Figure 4.1 represents the mean temperatures per month for the location Bremerhaven in the control and scenario period. By applying the linear scaling procedure, changes in the monthly means can now be seen. It is noticeable that the deviation between the adjusted time series and $T_{obs,p}$ in February, compared to the modeled data $T_{sim,p}$, has increased, i.e. worsened.

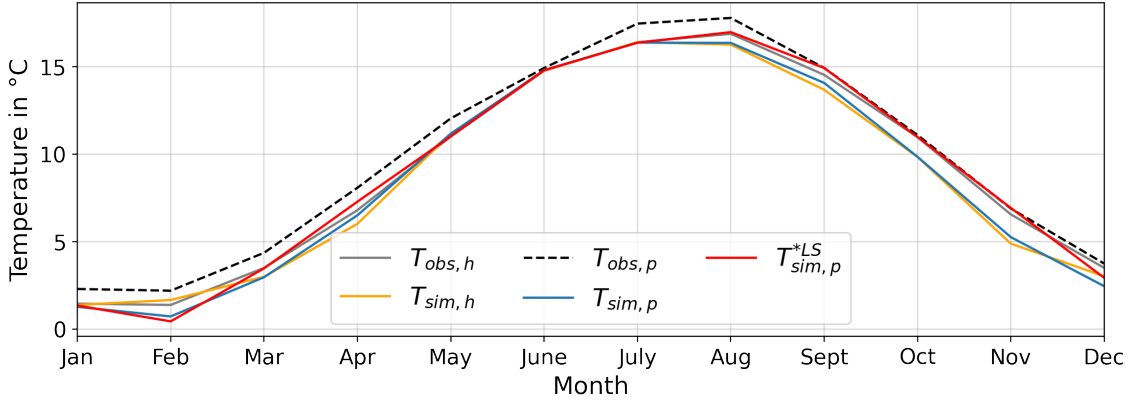


Figure 4.1.: Comparison of observed, modeled and linear scaling adjusted monthly mean 2m air temperatures in Bremerhaven at different periods

For this reason, the historical observed and modeled long-term monthly means are also shown. Since this bias adjustment procedure assumes that errors remain persistent, this difference was also taken into account during the adjustment. However, in the control period, the modeled time series is warmer than the observed data, but this is not the case in the period from 1981 to 2010, so the bias actually increased here.

In March and April, the difference has been somewhat reduced. In June, it is noticeable that the modeled and observed data of the control period show rather similar values, which means that here, too, the adjustment has brought only minimal changes. In autumn, on the other hand, a significant improvement is seen. This is because the difference between $T_{obs,h}$ and $T_{sim,h}$ is about the same as between $T_{obs,p}$ and $T_{sim,p}$.

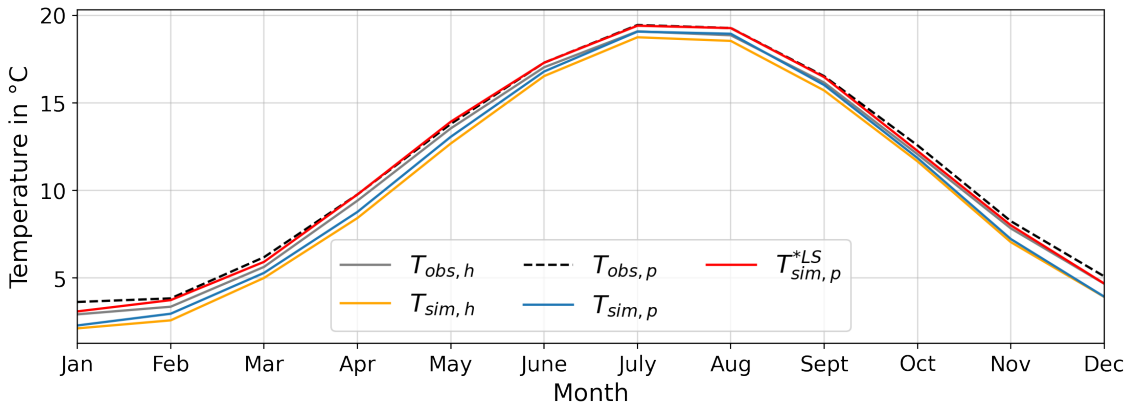


Figure 4.2.: Comparison of observed, modeled and linear scaling adjusted monthly mean 2m air temperatures in Europe at different periods

(Fig. 4.2) By applying the linear scaling procedure not only to the time series of one region but to all grid boxes of the modeled data presented in Chapter 3, it can be seen that the difference between modeled and observed data has the highest deviations in the winter months. In January, it can be seen that the difference between $T_{obs,h}$ and $T_{obs,p}$ has increased, but in the modeled data, this temperature increase between $T_{sim,h}$ and $T_{sim,p}$ is not as high. For this reason, a too small difference is added to the time series to be

adjusted, so that the LS-adjusted data set still show values that are too cold.

The difference of this technique compared to the delta method is that the linear scaling procedure attempts to project the historical difference between the observed and modeled data onto the time series to be adjusted, rather than adding the change between the modeled data onto the historical observations.

4.3. The Variance Scaling Method

As an extension of the linear scaling method, the variance scaling (VS) not only allows minimizing the deviations in the long-term monthly mean but can also adjust the long-term monthly variance of the temperatures of modeled climate data to approximate those of the historically observed time series. This assumes, of course, as with the linear scaling, that the climate conditions for the historical and future modeled and observed time series are the same. Because of the same conditions in different periods, it is also assumed that the bias between modeled and observed values remains constant over time.

When applying the variance scaling procedure on temperature data, the first step is to perform the additive linear scaling method as shown before (cf. Eq. 4.9f.). Then, the long-term monthly mean is subtracted from the linear scaled value at index i :

$$T_{sim,h}^{VS(1)}(i) = T_{sim,h}^{*LS}(i) - \mu_m(T_{sim,h}^{*LS}(i)) \quad (4.16)$$

$$T_{sim,p}^{VS(1)}(i) = T_{sim,p}^{*LS}(i) - \mu_m(T_{sim,p}^{*LS}(i)) \quad (4.17)$$

If the equations 4.16f. are applied to all entries of the time series, then:

$$\mu_m(T_{sim,h}^{VS(1)}(i)) = \mu_m(T_{sim,p}^{VS(1)}(i)) = 0 \quad (4.18)$$

(Eq. 4.19) With the long-term monthly mean shifted to zero, it is now possible to adjust the standard deviation by multiplying the intermediate result $T_{sim,p}^{VS(1)}$ by the ratio of its long-term monthly standard deviation and the result from Equation 4.16. This, when applied to all elements of $T_{sim,p}^{VS(1)}$, ensures that the values are scaled up or down without affecting the mean (i.e. adjustment of the standard deviation and so variance, too).

$$T_{sim,p}^{VS(2)}(i) = T_{sim,p}^{VS(1)}(i) \cdot \left[\frac{\sigma_m(T_{obs,h}(i))}{\sigma_m(T_{sim,h}^{VS(1)}(i))} \right] \quad (4.19)$$

(Eq. 4.20) Finally, the long-term mean of the respective month, of the time series linearly scaled at the beginning (cf. Eq. 4.9f.), is added with the standard deviation-adjusted value of $T_{sim,p}^{VS(2)}$ to account for the long-term monthly standard deviation as well as the long-term monthly mean.

$$T_{sim,p}^{*VS}(i) = T_{sim,p}^{VS(2)}(i) + \mu_m(T_{sim,p}^{*LS}(i)) \quad (4.20)$$

Thus, the linear scaling procedure is applied to adjust the data in the long-term monthly mean. To account for the standard deviation as well, the long-term monthly mean is subtracted from the previously linearly scaled time series (i.e. shift to zero mean; cf. Eq. 4.16f.) and multiplied by the ratio of the observed and modeled long-term monthly standard deviations (cf. Eq. 4.19). This completes the scaling of the long-term monthly standard deviation. To add the linear scaled long-term monthly mean again, the previous intermediate result is added to the linear scaled value from the beginning (cf. Eq. 4.20).

Example

Also, the variance scaling method is applied as an example to adjust the temperature value of December 17, 1997. In the first step, the linear scaling procedure is applied for the historical time series as well as for the time series to be adjusted. Thus two linear adjusted time series result, over which in the following the average of all December months is formed. This mean is subtracted from the temperature value at index $i = 6190$:

$$T_{sim,h}^{VS(1)}(i) = T_{sim,h}^{*LS}(i) - \mu_m(T_{sim,h}^{*LS}(i)) \quad T_{sim,p}^{VS(1)}(i) = T_{sim,p}^{*LS}(i) - \mu_m(T_{sim,p}^{*LS}(i)) \quad (4.21) \quad (4.24)$$

$$= 3.23^\circ\text{C} - 3.49^\circ\text{C} \quad (4.22) \quad = 4.82^\circ\text{C} - 2.92^\circ\text{C} \quad (4.25)$$

$$= -0.26^\circ\text{C} \quad (4.23) \quad = 1.9^\circ\text{C} \quad (4.26)$$

where:

$$i = 6190 \text{ (index of December, 17. 1997 in predicted period)}$$

$$\mu_m(T_{sim,h}^{*LS}(i)) = \text{Mean of all December temperatures in } T_{sim,h}^{*LS}$$

$$\mu_m(T_{sim,p}^{*LS}(i)) = \text{Mean of all December temperatures in } T_{sim,p}^{*LS}$$

To determine $T_{sim,p}^{VS(2)}(i)$, the previous step must be applied not only for $i = 6190$, but for all i that exhibit days in December months (cf. Eq. 4.1). Through this, the long-term mean of temperatures in December is shifted to zero. Once this is done, the standard deviation of the historical observed December values can be related to the standard deviation of the zero-mean shifted values of the historical modeled data calculated in the previous step. Multiplying this ratio by $T_{sim,p}^{VS(1)}(i)$ realizes the scaling of the standard deviation and so variance, too.

$$T_{sim,p}^{VS(2)}(i) = T_{sim,p}^{VS(1)}(i) \cdot \left[\frac{\sigma_m(T_{obs,h}(i))}{\sigma_m(T_{sim,h}^{VS(1)}(i))} \right] \quad (4.27)$$

$$= 1.9^\circ\text{C} \cdot \frac{3.12^\circ\text{C}}{3.85^\circ\text{C}} \quad (4.28)$$

$$= 1.5397^\circ\text{C} \quad (4.29)$$

where:

$$i = 6190 \text{ (index of December, 17. 1997 in predicted period)}$$

$$\sigma_m(T_{obs,h}(i)) = \text{Standard deviation of all December temperatures in } T_{obs,h}$$

$$\sigma_m(T_{sim,h}^{VS(1)}(i)) = \text{Standard deviation of all December temperatures in } T_{sim,h}^{VS(1)}$$

The result of Equation 4.29 corresponds to the standard deviation adjusted temperature value for the predicted time series. To add the long-term monthly mean, which was removed at the beginning, it can be added to the previously standard deviation scaled

intermediate result:

$$T_{sim,p}^{*VS}(i) = T_{sim,p}^{VS(2)}(i) + \mu_m(T_{sim,p}^{*LS}(i)) \quad (4.30)$$

$$= 1.5397^\circ\text{C} + 2.92^\circ\text{C} \quad (4.31)$$

$$= 4.4597^\circ\text{C} \quad (4.32)$$

where:

$i = 6190$ (index of December, 17. 1997 in predicted period)

$\mu_m(T_{sim,p}^{*LS}(i)) = \text{Mean of all December temperatures in } T_{sim,p}^{*LS}$

The variance scaling procedure also changed the temperature value and produced a warmer value for December 17, 1997 ($4.4597^\circ\text{C} > 4.34^\circ\text{C}$). However, this is not quite as warm as the linear scaling procedure reported. This is due to the fact that the standard deviation of December temperatures in $T_{sim,h}^{VS(1)}$ is higher than in historically observed December months and is therefore scaled-down (cf. Eq. 4.28).

Implementation

This can also be implemented in the Python programming language. Here, as in the numerical example, the linear scaling method is applied in the first step to then determine VS1_simh (i.e. $T_{sim,h}^{VS(1)}$) and VS1_simp (i.e. $T_{sim,p}^{VS(1)}$) by subtracting the linearly scaled mean. In line 11, the scaling of the standard deviation is done to return this scaled result added with the monthly mean in line 13.

```

1  def variance_scaling(
2      m_obs: [float], m_simh: [float], m_simp: [float]
3  ) -> np.array:
4
5      LS_simh = linear_scaling(m_obs, m_simh, m_simh)           # Eq. 4.10
6      LS_simp = linear_scaling(m_obs, m_simh, m_simp)          # Eq. 4.9
7
8      VS1_simh = LS_simh - np.nanmean(LS_simh)                 # Eq. 4.16
9      VS1_simp = LS_simp - np.nanmean(LS_simp)                 # Eq. 4.17
10
11     VS2_simp = VS1_simp * (np.std(m_obs) / np.std(VS1_simh)) # Eq. 4.19
12
13     return VS2_simp + np.nanmean(LS_simp)                     # Eq. 4.20
14
15     vs_corrected = variance_scaling(m_obs, m_simh, m_simp)    # cf. Tab. 4.2

```

Listing 4.3: Implementation of the variance scaling method for temperatures

Here it should be noted again that `m_obs`, `m_simh` and `m_simp` must be separated into monthly groups in line 15 according to Table 4.2. This means that the predefined method is applied to, for example, all Januaries in a time series of 30 years. A detailed example can be found in Section 4.6.

4.4. The Quantile Mapping Method

Distribution-based procedures, in contrast to the scaling-based techniques, do not aim to directly match the modeled time series in mean or variance to that of the observed period, but rather attempt to minimize distributional biases by transferring the values corresponding to the quantiles of the distribution function from one time series to another.

This is achieved in the simple quantile mapping (QM) procedure by inserting a temperature value of the time series to be adjusted ($T_{sim,p}(i)$) into the cumulative distribution function (CDF) of the modeled time series of the control period ($F_{sim,h}$). Hereby a quantile value is determined, which tells in which quantile the temperature value of the predicted time series is located in the simulated data of the control period. This quantile value is then inserted into the inverse CDF of the historical observations $F_{obs,h}^{-1}$, providing the adjusted temperature value for the time step of $T_{sim,p}(i)$.

The adjustment of a temperature value, using QM, can be performed with the following equation:

$$T_{sim,p}^{*QM}(i) = F_{obs,h}^{-1} \{F_{sim,h} [T_{sim,p}(i)]\} \quad (4.33)$$

There is no need to differentiate between curly and square brackets, nor is separation by month necessary. The application of the quantile mapping is most easily illustrated with visualizations, therefore a detailed example follows, in which the procedure is clarified.

Example

Figure 4.3 represents the first step in QM. Here, the temperature value of 4.34°C of the predicted time series $T_{sim,p}$ at index i with $i = 6190$ (i.e. index of December, 17. 1997) is inserted into the CDF of the modeled time series of the historical period $F_{sim,h}$ to obtain the corresponding y-value representing the 0.28 quantile (i.e. 28% quantile of the modeled time series of the control period).

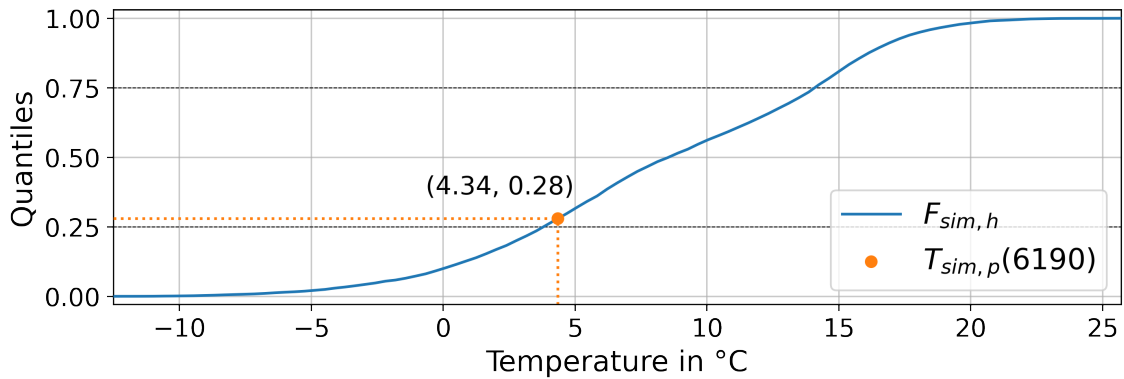


Figure 4.3.: Determination of the quantile value by inserting $T_{sim,p}(6190)$ into the CDF of the modeled time series of the control period; black lines indicate the 25% and 75% quantile respectively

(Fig. 4.4) Then, the value of 0.28 (i.e. 28% quantile) is inserted into the inverse CDF of the historical observations $F_{obs,h}^{-1}$ to determine the adjusted temperature value of 4.74°C

of the corresponding quantile in the historical period of the observed data. To adjust the entire predicted time series, this must be repeated for each value of $T_{sim,p}$.

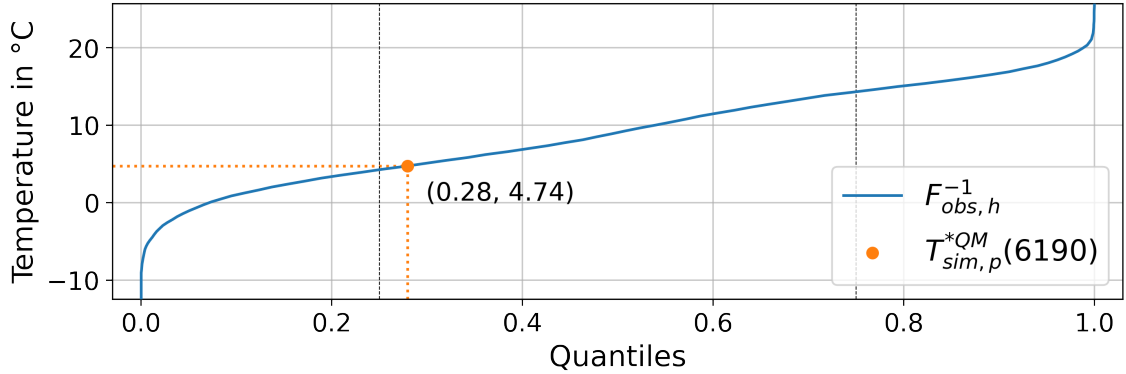


Figure 4.4.: Inserting the quantile value of 0.28 into the inverse CDF of historical observational data ($F_{obs,h}^{-1}$) to equate the adjusted temperature value of $T_{sim,p}(6190)$; black lines indicate the 25% and 75% quantile respectively

The term mapping is chosen for this reason, as the quantile value of $F_{sim,h}[T_{sim,p}(i)]$ is applied to the CDF of the historical observed time series to adjust a temperature value from the scenario time series according to this quantile.

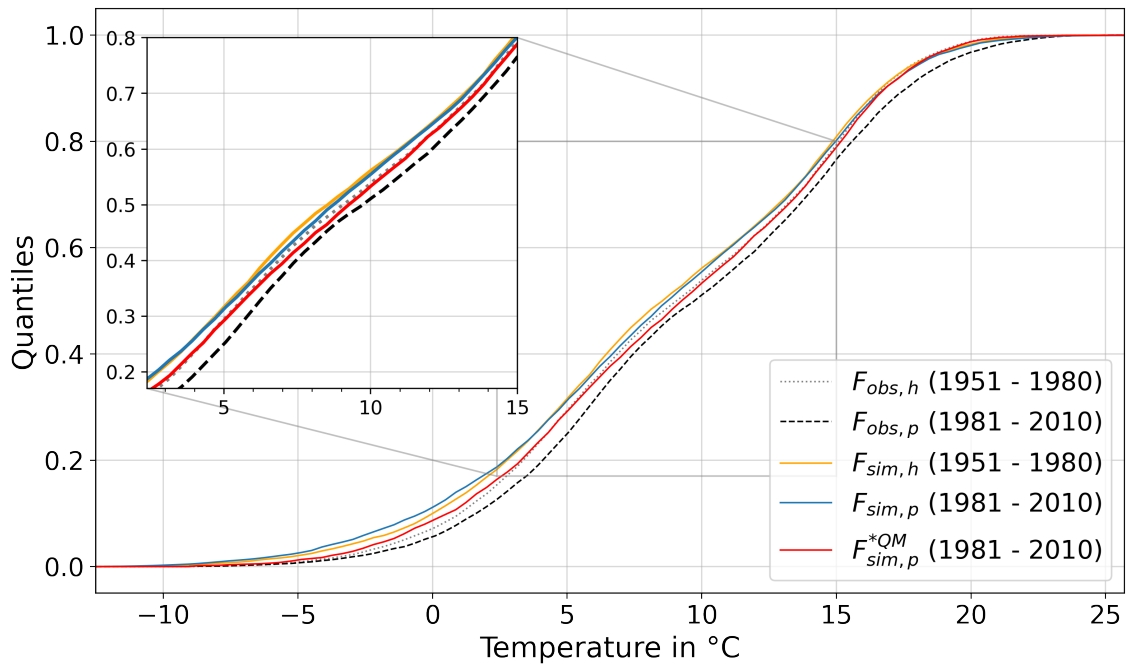


Figure 4.5.: Cumulative distribution functions of observed, modeled and QM-adjusted daily mean 2m air temperatures in Bremerhaven, Germany for different time periods

Figure 4.5 plots the cumulative distribution functions of the observed, as well as modeled and QM-adjusted time series for the location of Bremerhaven, Germany. It can be clearly seen that $F_{obs,p}$ has shifted to the positive range of values, which in turn is due to climate change and increasing temperatures. At the same time, it can be seen that the CDF of the QM-adjusted time series is similar to one of the observations of the control period. Since

$F_{sim,h}$ is significantly further to the left of $F_{obs,h}$, the procedure has applied this deviation to the modeled data of the predicted period. Therefore, the time series adjusted by QM has on average warmer values than $T_{sim,p}$. However, these still have a clear difference to $F_{obs,p}$.

In the highlighted subplot it can be seen that the historical modeled (yellow line), as well as the predicted data (blue line) initially, have very similar values in the quantiles between 0.2 to 0.3. Here, quantile mapping has minimized the historical bias of the data in the predicted time series, which is why the adjusted data (red line) almost match those of the historical observations (gray dotted line). In the quantiles between 0.4 to 0.5 a similar behavior can be seen, because here there is also a deviation between $F_{sim,h}$ and $F_{obs,h}$, which is compensated by QM using higher temperatures in the adjusted time series. However, as the distance between $F_{sim,h}$ and $F_{sim,p}$ has increased, $F_{sim,p}^{*QM}$ is shifted even further to the right, causing it to no longer match the *CDF* of $T_{obs,h}$, as it did shortly before.

Since the climate is changing and thus the data of the observations do not remain the same, the QM adjustment technique can only provide an approximation of the distributions. Also, the changes between historical and predicted modeled data are not considered here. However, if the modeled data are identical, the adjustment can ensure that the distributions can produce a very high agreement with the observed data of control period.

The missing trends between the modeled time series, which also include the climate change, are adjusted when the simple quantile mapping procedure is applied. Although the data can be adjusted using historical values, their extreme values are not adequately taken into account (Cannon et al., 2015). Also, values produced by QM can never be higher or lower than the maximum or minimum value of the observed time series of the control period, since for example, the 100% quantile corresponds to the highest temperature value in $F_{obs,h}^{-1}$ (cf. Fig. 4.4), so this procedure is only suitable for adjusting predicted data whose range of values is approximately equal to that of the historical observations.

Implementation

The QM technique can be implemented in Python by first determining the probability boundaries (xbins) by dividing the range between the minimum and the maximum temperatures of the control period into n equally-sized sections:

```

1  def get_xbins(obs: [float], simh: [float], n: int=100) -> np.array:
2      global_max: float = max(np.amax(obs), np.amax(simh))
3      global_min: float = min(np.amin(obs), np.amin(simh))
4      wide: float = abs(global_max - global_min) / n
5      return np.arange(global_min, global_max + wide, wide)
6
7  xbins: np.array = get_xbins(obs, simh)

```

Listing 4.4: Define monotonously increasing vector xbins, containing equally sized bins between the maximum and minimum historical modeled and observed temperature values

These probability boundaries are needed to determine the probability density function (*PDF*) of the modeled, as well as of the observed time series (cf. Fig. 4.6 and Lst. 4.5l. 2). In this thesis, the value of 100 is always used for the parameter *n*, thus *xbins* is a monotonically increasing vector with 101 values, which define the limits of the probability boundaries based on the temperature values of the control period. Therefore, *xbins* defines the boundaries in which the bars of a density function are represented in a histogram, so 101 *xbins* values will show 100 histogram bars:

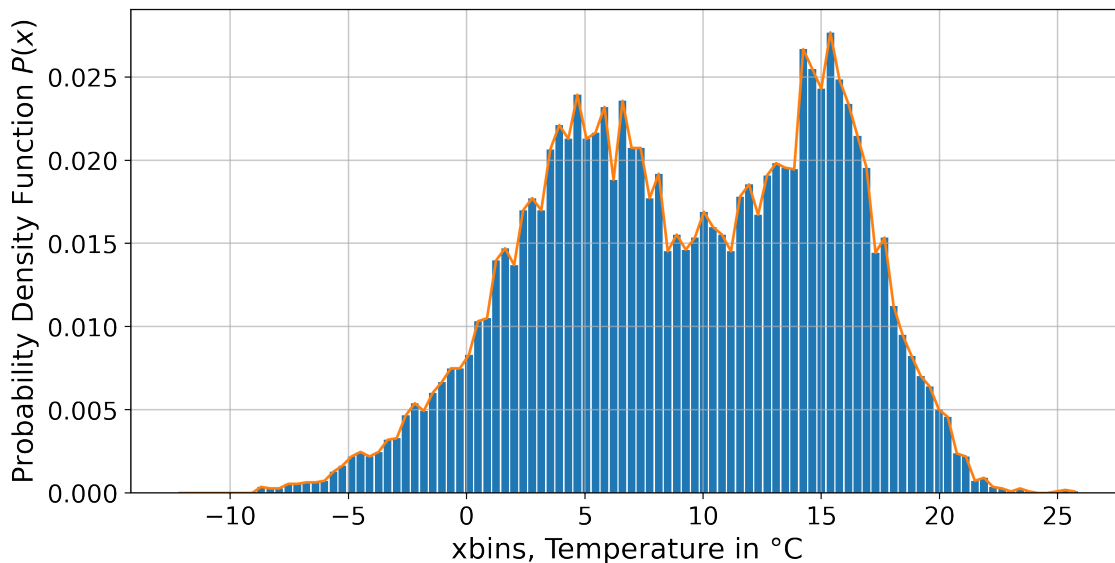


Figure 4.6.: Probability density function as line and bar plot of observed daily mean temperatures in Bremerhaven, Germany between 1951 and 1980 as a representation of the need of *xbins*

The probability density function of a time series is needed to determine the corresponding *CDF* by summing up the values (i.e. densities) of the *PDF* (cf. Lst. 4.5l. 3).

```

1  def get_cdf(a: [float], xbins: [float]) -> np.array:
2      pdf, _ = np.histogram(a, xbins)                                # compute P(x)
3      return np.insert(np.cumsum(pdf), 0, .0)                       # compute F(x)

```

Listing 4.5: Define function *get_cdf* that computes the *PDF* of a vector (i.e. 1D sequence) by given probability boundaries (i.e. *xbins*) to equate and return the *CDF*

The summation of the values is achieved with the help of the function *cumsum* from the Numpy module. At the same time, a zero is shifted to the first position of the array determined in this way. This is important in order to interpolate between 0% quantile and the minimum occurred value later.

Since now a function is defined, with which the *CDF* of a time series can be determined, it is time to bundle all this in a final function (cf. Lst. 4.6f.). Here, in the first step, the previously described boundaries *xbins* are determined with the help of the function `get_xbins` (l. 4), in order to subsequently determine the cumulative distribution functions of $T_{obs,h}$ and $T_{sim,h}$ (l. 5f.):

```

1  def quantile_mapping(
2      obs: [float], simh: [float], simp: [float], n_quantiles: int=100
3  ) -> np.array:
4      xbins: np.array = get_xbins(obs, simh, n_quantiles)           # Lst. 4.4
5      cdf_obs: np.array = get_cdf(obs, xbins)                       #  $F_{obs,h}$ 
6      cdf_simh: np.array = get_cdf(simh, xbins)                    #  $F_{sim,h}$ 

```

Listing 4.6: Implementation of the quantile mapping method (Part 1)

(Lst. 4.7) To calculate according to the Equation 4.33 $F_{sim,h}[T_{sim,p}]$, the function `interp` from the module `Numpy` can be applied with the parameters `x`, `xp` and `fp`. Here, for `x`, the values to be interpolated from `simp` ($simp = \{T_{sim,p}(i) \mid \forall i \in \{0, 10949\}, i \in \mathbb{N}\}$) are used to determine the associated quantile values (`y_values`) with the help of linear interpolation. `xp` corresponds to the probability boundaries `xbins` and thus forms the `x`-values of the interpolation function. The parameter `fp` is passed `cdf_simh` ($F_{sim,h}$), because this vector contains the `y`-values (i.e. quantile values) of the function. Line one to three is the execution of what is shown in Figure 4.3. The last line realizes the subsequent insertion into $F_{obs,h}^{-1}$ (cf. Fig. 4.4).

```

1      y_values: np.array = np.interp(                               # Eq. 4.33; Fig.4.3
2          x=simp, xp=xbins, fp=cdf_simh
3      )
4      return np.interp(x=y_values, xp=cdf_obs, fp=xbins)          # Eq. 4.33; Fig.4.4

```

Listing 4.7: Implementation of the quantile mapping method (Part 2)

Unlike the previous bias adjustment implementation examples, quantile and quantile delta mapping do not require `obs`, `simh`, and `simp` to be separated at equal months as shown in Table 4.2. For this reason, this method can be applied directly to the full-time periods.

```

1      qm_corrected: np.array = quantile_mapping(
2          obs=obs,                                                 #  $T_{obs,h}$ 
3          simh=simh,                                              #  $T_{sim,h}$ 
4          simp=simp                                              #  $T_{sim,p}$ 
5      )

```

Listing 4.8: Execute the quantile mapping method

The number of probability boundaries (i.e. length of `xbins`), results from the value, which is passed for `n` to the function `get_xbins` (cf. Lst. 4.4). This value not only specifies

how many bins the *PDF* is divided into, but is also crucial when linear interpolation is applied (e.g. when inserting $T_{sim,p}(i)$ into $F_{sim,h}$; Eq. 4.33). If this value is too low, for example only 10, the range of $P(x)$ is only divided into 10 equal parts, which can cause inaccuracies in the further course of the calculations, because the distributions cannot be mapped sufficiently, since the distances between which interpolation is performed are too large.

Since temperatures in the data presented here have approximately a range of values from -50°C to $+50^{\circ}\text{C}$, this range can be divided into 100 parts, whereby the spacing of the values in the vector `xbins` is approximately 1°C , provided that a region is adjusted which also covers such a range of values. If the values of a time series to be adjusted are only between -10°C and $+30^{\circ}\text{C}$, this will also be divided into 100 equal parts, resulting in smaller distances of the values in `xbins`.

Since data for 30 years with a total of 10950 days per grid cell are used here, this value can also be set significantly higher, such as 250 or even more. Since most of the cells adjusted here only have temperatures between -10°C and 25°C , which means that the distances in `xbins` are only about 0.35°C between which a value is interpolated (when length of `xbins` = 101), the value of 100 is assumed to be sufficient and is used in both quantile mapping and quantile delta mapping. As an aside, Figures 4.3f. and 4.6 map the *CDFs* of various time series. These have exactly 100 values.

Determining the optimal number of boundaries is a subfield unto itself and certainly requires some further investigation, as it may be defined differently depending on the variable, time period, location, and quality of the data.

4.5. The Quantile Delta Mapping Method

The quantile delta mapping procedure (QDM) for temperatures is characterized by taking into account the absolute changes in the quantiles between historical and future simulated climate model output and thus obtaining the model-specific trends, which are adjusted in the simple quantile mapping (cf. Section 4.4). By trend, in terms of the data used here, the the absolute change in temperatures between historical and projected modeled period is meant. Here, the simple quantile mapping technique is applied in a modified form as part of the transfer function.

(Eq. 4.34) First, the temperature value of $T_{sim,p}$ at index i is inserted into the cumulative distribution function $F_{sim,p}$. This gives the value of the quantile for the temperature at index i . Since this value is used several times in the following equations, it is assigned to the variable ε .

$$\varepsilon(i) = F_{sim,p} [T_{sim,p}(i)], \quad \varepsilon(i) \in \{0, 1\} \quad (4.34)$$

(Eq. 4.35) This is followed by minimizing the bias between the observations of the control period and the time series to be adjusted by inserting the previously calculated quantile value $\varepsilon(i)$ into the inverse *CDF* of the historical observations $F_{obs,h}^{-1}$. This is done similarly to Figure 4.4, but in QDM $F_{sim,p} [T_{sim,p}(i)]$ is inserted into $F_{obs,h}^{-1}$ instead of $F_{sim,h} [T_{sim,p}(i)]$.

This is called detrending, because the trends or the absolute changes of the quantiles between the control and predicted period of the modeled time series are ignored (Tong et al., 2021). $T_{sim,p}^{QDM(1)}(i)$ now represents an adjusted temperature value, which was determined based only on the observational time series of the control period and the modeled time series of the scenario period.

$$T_{sim,p}^{QDM(1)}(i) = F_{obs,h}^{-1} [\varepsilon(i)] \quad (4.35)$$

(Eq. 4.36) Now, to determine the absolute change in quantiles between historical and predicted modeled time series, the quantile value $\varepsilon(i)$ is inserted into the inverse *CDF* of $T_{sim,h}$ to obtain the temperature value of the historical modeled time series corresponding to this quantile.

This value is then subtracted from the value to be adjusted to determine the difference. $\Delta(i)$ is thus a temperature value representing the absolute change in quantiles between the historical and predicted values.

$$\Delta(i) = F_{sim,p}^{-1} [\varepsilon(i)] - F_{sim,h}^{-1} [\varepsilon(i)] \quad (4.36)$$

$$= T_{sim,p}(i) - F_{sim,h}^{-1} \{F_{sim,p} [T_{sim,p}(i)]\} \quad (4.37)$$

(Eq. 4.38) Finally, the previously determined temperature value $\Delta(i)$, which represents the change in quantiles, is added to the temperature value $T_{sim,p}^{QDM(1)}(i)$ adjusted in Equation 4.35, which was determined based on the observed time series of the control period and

the time series to be adjusted only.

$$T_{sim,p}^{*QDM}(i) = T_{sim,p}^{QDM(1)}(i) + \Delta(i) \quad (4.38)$$

By adding the change between the quantiles of the simulated time series, the results of quantile delta mapping are no longer bound to the range of values of the historical observations, as was previously the case with the simple quantile mapping technique.

To adjust the bias in precipitation time series by QDM, it is recommended to determine the relative change in Equation 4.36 by dividing instead of subtracting. Subsequently, multiplication must be performed in Equation 4.38. However, since this does not apply in this thesis, the reader is therefore referred to the article by Tong et al., 2021.

Example

To adjust a temperature value in $T_{sim,p}$ at time step i using QDM, so that it ideally matches the observed value of $T_{obs,p}(i)$, the value of the quantile of $T_{sim,p}(i)$ must be determined. This is accomplished by inserting the temperature value that occurred on that day, from the predicted modeled time series into the *CDF* of $T_{sim,p}$ (cf. Eq. 4.34):

$$\varepsilon(i) = F_{sim,p}[T_{sim,p}(i)] = 0.1997 \quad (4.39)$$

where:

$$\begin{aligned} i &= 6190 \text{ (index of December, 17. 1997 in predicted period)} \\ T_{sim,p}(i) &= 4.34^\circ\text{C} \end{aligned}$$

The 0.1997 quantile thus determined, which indicates in which quantile the temperature of 4.34°C in $T_{sim,p}$ is located in the predicted time series, is now inserted into the inverse *CDF* of the historical observations (cf. Eq. 4.35) to adjust the value, similar to the simple quantile mapping, but without mapping between the modeled time series nor taking into account the absolute change of the quantile between historical and projected modeled data:

$$T_{sim,p}^{QDM(1)}(i) = F_{obs,h}^{-1}(0.1997) = 3.35^\circ\text{C} \quad (4.40)$$

The value of 3.35°C obtained in this way represents the adjusted temperature value, which does not take into account the change in the quantiles between the modeled time series. It indicates the temperature of the 0.1997 quantile in the observed time series of the control period.

(Eq. 4.41ff.) In parallel, the value of $\varepsilon(i)$ is also inserted into $F_{sim,h}^{-1}$ (cf. Eq. 4.36) to determine the historical modeled temperature value corresponding to quantile 0.1997. The difference of this determined value with $T_{sim,p}(i)$ results in $\Delta(i)$ and represents a temperature value describing the absolute change of the quantile between the control and

predicted modeled period:

$$\Delta(i) = T_{sim,p}(i) - F_{sim,h}^{-1}[\varepsilon(i)] \quad (4.41)$$

$$= 4.43^{\circ}\text{C} - F_{sim,h}^{-1}(0.1997) \quad (4.42)$$

$$= 4.43^{\circ}\text{C} - 2.75^{\circ}\text{C} \quad (4.43)$$

$$= 1.59^{\circ}\text{C} \quad (4.44)$$

The temperature value $\Delta(i)$ is then added to the detrended-adjusted value of Equation 4.40 to obtain, in addition to the temperature value adjusted based only on the observational control period time series, the absolute change in the quantile (i. e. change between $F_{sim,h}[T_{sim,p}(i)]$ and $F_{sim,p}[T_{sim,p}(i)]$):

$$T_{sim,p}^{*QDM}(i) = T_{sim,p}^{QDM(1)}(i) + \Delta(i) \quad (4.45)$$

$$= 3.35^{\circ}\text{C} + 1.59^{\circ}\text{C} \quad (4.46)$$

$$= 5.04^{\circ}\text{C} \quad (4.47)$$

Finally, it can be seen that the temperature value of December 17, 1997, adjusted using the quantile delta mapping method, is 0.7°C warmer than the model reported for that date.

Implementation

Since the inverse *CDF* of a time series must be formed several times in the quantile delta mapping procedure, it is worthwhile to write a method for this at the beginning. This also makes it easier to understand the subsequent source code section. This is implemented by interpolating again with the help of the function `interp` from the Numpy module.

```

1  def get_inverse_cdf(
2      base: [float], insert: [float], bins: [float]
3  ) -> np.array:
4      return np.interp(x=insert, xp=base, fp=bins) # F_base^{-1}[insert]
```

Listing 4.9: Implementation of function `get_inverse_cdf` that computes the inverse *CDF* and enables inserting values in one step

(Lst. 4.10) The quantile delta mapping procedure can now be performed by first determining the probability boundaries `xbins` (l. 4) using the function defined in Listing 4.4. This number of quantiles, whose default value here is 100, has the same influence on the result as in the simple quantile mapping. Afterward, the cumulative distribution functions of the historical time series and also of the time series to be adjusted can be calculated (l. 7ff.).

```

1  def quantile_delta_mapping(
2      obs: [float], simh: [float], simp: [float], n_quantiles: int=100
3  ) -> np.array:
```

```

4     xbins: np.array = get_xbins(obs, simh, n_quantiles)
5
6     # compute CDFs
7     cdf_obs: np.array = get_cdf(obs, xbins)           #  $F_{obs,h}$ 
8     cdf_simh: np.array = get_cdf(simh, xbins)        #  $F_{sim,h}$ 
9     cdf_simp: np.array = get_cdf(simp, xbins)        #  $F_{sim,p}$ 

```

Listing 4.10: Implementation of quantile delta mapping method (Part 1)

After the *CDFs* are determined, the Equations 4.35ff. can be applied by realizing the insertion via linear interpolation:

```

1     # calculate exact CDF values ( $F_{sim,p}[T_{sim,p}]$ , i.e.  $\varepsilon$ )
2     epsilon: np.array = np.interp(                   # Eq. 4.34
3         x=simp, xp=xbins, fp=cdf_simp
4     )
5
6     # insert  $\varepsilon$  into inverted  $F_{obs,h}$ 
7     QDM1: np.array = get_inverse_of_cdf(             # Eq. 4.35
8         cdf_obs, epsilon, xbins
9     )
10
11    # compute  $\Delta$ 
12    delta: np.array = simh - get_inverse_cdf(        # Eq. 4.36f.
13        cdf_simp, epsilon, xbins
14    )
15
16    return QDM1 + delta                             # Eq. 4.38

```

Listing 4.11: Implementation of quantile delta mapping method (Part 2)

As with simple quantile mapping, it is not necessary to separate the months beforehand. Therefore, the quantile delta mapping method can be applied directly to the full period of the time series, just like the simple QM. These must be passed to the method `quantile_delta_mapping` in the form `list` or `numpy.array`:

```

1     qdm_corrected = quantile_delta_mapping(
2         obs = obs,                                   #  $T_{obs,h}$ 
3         simh = simh,                                 #  $T_{sim,h}$ 
4         simp = simp                                  #  $T_{sim,p}$ 
5     )

```

Listing 4.12: Application of the quantile delta method

The variable `qdm_corrected` contains the same number of values after execution as the passed 1D sequence `simp` and includes all temperature values adjusted by QDM ($qdm_adjusted = T_{sim,p}^{*QDM}$).

4.6. Extensive Implementation

In order to show how the presented scaling-based procedures can be applied using Python, a detailed example follows, in which the linear scaling procedure is carried out in order to adjust the time series from 1981 to 2010 of the modeled data with the help of modeled and observed data of the period 1951 to 1980. Here, it is particularly important to note that the adjustment is performed for a specific location. Thus, the loaded data sets must have a resolution of 1 x 1 longitude/latitude and in this example, they have 10950 daily mean 2m air temperature values spanning 30 consecutive years, not including days that occur only in leap years. If it is desired to bias-adjust data sets with higher spatial resolutions, a loop can be constructed around the following example, which iterates over all desired locations or even vertical dimensions. Depending on the resolution, it is advisable to split the calculations into several processes, as the adjustments of different locations can be calculated independently of each other.

First, the data sets need to be loaded into the Python script. For this, the module Xarray is used, which must be imported as shown in Listing 1.2. In the same step, the variable “tas” (i.e. temperatures) is selected, which changes the data type of the loaded data set from `xarray.core.dataarray.Dataset` to `xarray.core.dataarray.DataArray`. This simplifies more direct access to the values and at the same time defines that only temperature values are used.

```

1  obs: xr.core.dataarray.DataArray = xr.open_dataset(           #  $T_{obs,h}$ 
2      'path/to/observational_dataset_1951-1980.nc'
3  )['tas']
4  simh: xr.core.dataarray.DataArray = xr.open_dataset(        #  $T_{sim,h}$ 
5      'path/to/modeled_dataset_1951-1980.nc'
6  )['tas']
7  simp: xr.core.dataarray.DataArray = xr.open_dataset(       #  $T_{sim,p}$ 
8      'path/to/modeled_dataset_1981-2010.nc'
9  )['tas']

```

Listing 4.13: Loading $T_{obs,h}$, $T_{sim,h}$ and $T_{sim,p}$ into a Python script

Once this data has been successfully loaded, it is time to specify which of the methods presented should be used to adjust the data. The following functions are available for selection:

Method	Listing
<code>delta_method</code>	4.1
<code>linear_scaling</code>	4.2
<code>variance_scaling</code>	4.3

Table 4.4: Available methods for long-term monthly scaling-based bias adjustment implemented in Python

In the following example, the linear scaling method is chosen by assigning the function `linear_scaling`, described in Listing 4.2, to the variable `method`. Furthermore, the variable `results` is initialized by copying the data set, which has all information of the time series to be adjusted, into it. The parameter `deep` is set to `True` here, so that a copy and no reference is created from the object. Then the function `load` is called so that this 1D sequence is completely loaded into memory. This is necessary for longer time series in order to set values at specific indexes as the sequence progresses. This is because Xarray uses the Dask backend, which only loads data sets into working memory using chunks whose values cannot be accessed directly without first loading them completely into the working memory.

```

1  # define adjustment method
2  method: function = linear_scaling           # cf. Lst. 4.2
3
4  # prepare result data set by deep copying the scenario data set
5  result: xr.core.dataarray.DataArray = simp.copy(deep=True).load()

```

Listing 4.14: Selection of the desired adjustment method and preparation of the results data set by deep copying all attributes and values of the $T_{sim,p}$ time series

(Lst. 4.15) The Xarray module provides many useful functionalities to select specific spatial and temporal scales. For example, using the `groupby` method, which is available by default to all objects of the `xarray.core.dataarray.DataArray` class, it is possible to create 12 groups containing all indices of the temperature values of the given month. To subsequently assign these grouped values to a variable in the `dict` data type, the `groups` attribute is accessed.

```

1  groups: dict = simh.groupby('time.month').groups

```

Listing 4.15: Create dictionary containing indices grouped by month

In some cases, for example, when adjusting precipitation, it may even lead to better results if not grouped by month but by season using `time.season`. Grouping to `time.dayofyear` is also possible, but leads to problems with the current version of Xarray, because days in leap years cannot be considered sufficiently by this approach due to their less frequent occurrence. After applying the grouping described in Listing 4.15, the variable `groups` contains 12 keys corresponding to integer values from 1 to 12. These keys are assigned the indices of the corresponding days in the respective month.

```

1  groups: dict = {
2      1: {                                     # Januaries
3          0, 1, 2, 3, ..., 30,               # year 1 (1951)
4          395, 396, ..., 425,               # year 2 (1952)
5          ..., ..., ...,
6          10585, 10586, ..., 10615         # year 30 (1980)
7      },

```

```

8     2: { ... },           # Februaries
9     ...,
10    ...,
11    ...,
12    12: { ... }         # Decembers
13 }

```

Listing 4.16: Pseudocode representing the grouped key-value pairs containing monthly separated indices

Since now the respective indices of all days of a certain group, which indicates the respective month, are assigned, it can be iterated over each key ($\text{month} = \text{key} \in \{1, 12\}$) (cf. Lst. 4.17l. 1).

(Lst. 4.17l. 2) Then the variables `m_obs`, `m_simh` and `m_simp` are declared. They are initialized by iterating over all indices assigned to the key month of the dictionary groups in another loop (l. 4ff.). Thus, the temperature values at the corresponding index, from the time series loaded in Listing 4.13, can be selected and assigned to the corresponding list. After this initialization is performed, `m_obs`, `m_simh`, and `m_simp` are 1-dimensional sequences of float values containing all temperature values of a month, over the entire 30-year period under consideration (cf. Eq. 4.1).

```

1     for month in groups.keys():
2         m_obs, m_simh, m_simp = [], [], []
3
4         for i in groups[month]:
5             m_obs.append(obs[i])
6             m_simh.append(simh[i])
7             m_simp.append(simp[i])

```

Listing 4.17: Declaration and initialization of the 1D sequences containing all temperature values of all same named months: $m(T_{obs,h}(i))$, $m(T_{sim,h}(i))$ and $m(T_{sim,p}(i))$

(Lst. 4.17) Now that the 1D sequences of a particular month (e.g. all Januaries if $\text{month} = 1$) are prepared over the entire 30-year period, these vectors of data type `list` can be passed to the method to adjust the bias (l. 1ff.).

```

1     computed_result: np.array = method(
2         m_obs = m_obs,           # all e.g. Januaries between 1951 - 1980
3         m_simh = m_simh,        # -- // -- between 1951 - 1980
4         m_simp = m_simp,        # -- // -- between 1981 - 2010
5         kind = '+'              # set additive method
6     )
7
8     for i, index in enumerate(groups[month]):
9         result[index] = computed_result[i]

```

Listing 4.18: Application of the adjustment method and mapping of the computed results to the right position in the result sequence

(Lst. 4.18) The variable `computed_result` contains the adjusted values, for example of all Januaries. To transfer these to the final dataarray `result` prepared in Listing 4.14, these adjusted values must be set at the correct indexes. This is achieved by again iterating over all indices of the respective month (l. 8f.) to determine the corresponding index (i.e. `index`) in the source data set `simp` for each adjusted value and assigning it afterward.

After the outer loop defined in Listing 4.17 finishes, the variable `result`, which is of data type `xarray.core.dataarray.DataArray`, contains 10950 adjusted values. This completes the bias adjustment for this procedure.

```

1  print(obs[:10])
2  >  [
3  >    -1.766821,  1.51333 ,  1.356714,  2.278741,  3.902856,  4.230951,
4  >    5.146173,  3.950952,  4.558404,  3.350427,  5.191095,  4.790887
5  >  ]
6  print(simh[:10])
7  >  [
8  >    3.086204,  3.941287,  3.566806,  1.656013, -3.553522, -4.855335,
9  >    1.640700, -5.968388, -8.783854, -7.284368, -2.95557 , -1.361990
10 >  ]
11 print(simp[:10])
12 >  [
13 >    -0.677857, -1.611131, -1.641309, -3.011029, -1.313527,  1.022544,
14 >    5.259728,  5.509796,  1.314042,  0.981995,  1.003031, -0.400324
15 >  ]
16 print(result[:10])
17 >  [
18 >    -0.602686, -1.535959, -1.566137, -2.935858, -1.238356,  1.097715,
19 >    5.334899,  5.584968,  1.389213,  1.057166,  1.078203, -0.325152
20 >  ]

```

Listing 4.19: Output first 10 values of $T_{obs,h}$, $T_{sim,h}$, $T_{sim,p}$ and $T_{sim,p}^{*LS}$

Listing 4.19 shows the first 12 values of the historical, as well as predicted and linear scaled time series. These correspond to the first 12 days in January of the years 1951 and 1981, respectively. Here it can be seen that the adjusted time series has about 0.075°C warmer values on each day than the predicted time series. This is because the long-term monthly mean of the modeled data of the control period is slightly colder than in the observed time series. Thus, all Januaries in this grid cell during the period from 1981 to 2010 inclusive, which are adjusted with these historical data using the linear scaling procedure, are 0.075°C warmer than the predicted time series for this period.

To save this adjusted time series as a NetCDF file, the function `to_netcdf` can be called at the end of the script:

```

1  result.to_netcdf('linear_scaling_result.nc', name='tas')

```

Listing 4.20: Save adjusted data set to file

This data set now contains 10950 adjusted daily mean 2m air temperature values for the period 1981 to 2010 inclusive. In addition, this has various meta-information addressed in Chapter 3, such as the exact timestamps of the entries, the history of the data set, and, most importantly, the geographic coordinates of these values.

Since grouping by month is not applied for quantile and quantile delta mapping, adjusting a time series using these methods can be performed as described in Listing 4.8 and 4.12 respectively.

4.7. Hands on C++

In the final part of the implementations, the presented methods will be shown in the programming language C++. Here however the focus will be exclusively on the methods, this means that neither the loading nor the saving is represented, since this does not work as in Python, using a module, which loads the data set into the program by indication of the file path alone.

The complete C++ implementations, as well as executable examples, are available in the GitHub repository listed in Table 1.2. There are also data structures for loading and saving, as well as a main program that implements the application of the methods to multidimensional NetCDF data sets. The methods presented in the following are part of the class `CMethods`. In the repository, this can be found in the `lib` directory. This class is declared in the header file `CMethods.h` and implemented in `CMethods.cpp`. All functions inside are declared with `static`. An executable file named `DoBiasAdjustment` allows the adjustment of climate data with all methods presented in this thesis. The source code of this program is also available as a `.cpp` file.

Exception handling, as well as warning and hinting when parameters are wrong or invalid, have not been considered in the following examples, as in the Python implementations, so that the focus on the methods for adjusting the bias is not lost. Furthermore, there is a class `MyMath`, which has functions like `mean` and `sd`, which calculate the mean of a time series, respectively the standard deviation.

All analyses performed in Section 5 are based on data sets whose bias has been minimized using these methods implemented in C++. As mentioned in Section 1.3, the g++ compiler (Apple clang version 13.1.6) with the standard library `std` version 17 is used. Since this is done on a macOS operating system, this is optimized for the target system `x86_64-apple-darwin21.5.0`.

Since the scaling-based bias adjustment procedures, in contrast to the quantile and quantile delta mapping procedures, require that they are applied to long-term monthly data sets, they are presented separately below. It is also assumed here that these methods are each applied to a single time series. This means that a time series of a grid cell is adjusted by using the same location for the historical observed and modeled time series as well as for the time series to be adjusted, and their values are stored in float arrays. These arrays must have the same length. An example is shown at the end in Listing 4.33.

4.7.1. Scaling-based Bias Adjustment Techniques in C++

As with the implementation of the scaling-based methods in Python, it also applies in C++ that the observed, as well as the modeled time series, which are passed to the function call of an adjustment method, must be separated into monthly separate groups. This can be achieved with the presented tool `cdo` by running a loop before the call of the program, which contains the adjustment methods, to iterate over the input files, separates these in each case into 12 individual data sets and stores them under temporary paths.

```

1  declare -a data_sets=("obs.nc" "simh.nc" "simp.nc")    # input file paths
2  declare -a tmp_paths=("tmp_obs" "tmp_simh" "tmp_simp" "tmp")
3  for d in "${tmp_paths[@]}; do mkdir -p $d; done
4  for (( month=1; month<13; month++ ));do
5      declare -i index=0
6      for dataset in "${data_sets[@]}; do
7          cdo -f nc -s                                \ # silent .nc forced output
8              -selvar,tas                             \ # select variable "tas"
9              -selmon,$month $dataset                 \ # selection of month
10             "${tmp_paths[index]}/${month}.nc" & \ # execute in background
11         (( ++index ))
12     done
13     wait
14 done

```

Listing 4.21: Data set preparation for scaling-based bias adjustment

In Listing 4.21 Bash code is shown, in which the first two arrays are declared and initialized, which contain on the one hand the respective paths to the observed and modeled data, on the other hand for each of these three data sets a directory path, likewise in an array is stored and afterward created (l. 2f.). Finally, two loops iterate over the numbers from 1 up to and including 12, to separate the respective month for each data set in the inner loop using `cdo` and to store the separated data in the respective temporary directory. This creates 36 data sets, 12 for each input file. Thus, for example, each January of the observed data can be found in a file named “1.nc” in the directory `tmp_obs`. This separation can be done in parallel since the execution of the `cdo` commands are not dependent on each other.

The scaling-based adjustment procedures have almost all the same parameters in the C++ implementation. These include the output pointer, which points to a float array with the same length, as well as the `m_obs`, `m_simh` and `m_simp` parameters. This length is passed via the `n_time` parameter. The array pointed to by the pointer `output` is the adjusted time series after applying the respective method. Again, it should be noted that this is done per month, which means that the variables and parameters prefixed with `m_` each represents one month over the full period of, say, 30 years. The parameter `kind` specifies whether the adjustment is to be performed additively or multiplicatively.

Delta Method

The procedure of the delta method in C++ is almost the same as in the Python implementation (cf. Lst. 4.1). In the first step, the long-term monthly mean values of the modeled time series $m(T_{sim,h}(i))$ and $m(T_{sim,p}(i))$ are calculated by calling the method `mean` from the class `MyMath`. This class is not shown here. Then, depending on the value of the parameter `kind`, a loop iterates over all time steps to perform the adjustment calculation and stores the result in the main function (cf. Lst. 4.33) using the pointer `output` at the corresponding position of the array that is used to store the adjusted values. Since C++ can work with pointers and references, this function has no return values.

```

1  void CMethods::Delta_Method(
2      float* output, float* m_obs, float* m_simh, float* m_simp,
3      unsigned n_time, std::string kind
4  ) {
5      const float
6          simh_mean = MyMath::mean(m_simh, n_time),           //  $m(T_{sim,h}(i))$ 
7          simp_mean = MyMath::mean(m_simp, n_time);           //  $m(T_{sim,p}(i))$ 
8
9      if (kind == "add" || kind == "+") {
10         for (unsigned ts = 0; ts < n_time; ts++)
11             output[ts] = m_obs[ts] + (simp_mean - simh_mean); // Eq. 4.3
12     } else if (kind == "mult" || kind == "*") {
13         for (unsigned ts = 0; ts < n_time; ts++)
14             output[ts] = m_obs[ts] * (simp_mean / simh_mean); // Eq. 4.8
15     }

```

Listing 4.22: Implementation of the additive and multiplicative delta method in C++

Linear Scaling

The implementation of the linear scaling method for climate data is very similar to the delta method just presented, but here the first step is not to take the long-term monthly mean of the time series to be adjusted $m(T_{sim,p}(i))$, but that of the time series from the control period $m(T_{obs,h}(i))$ and $m(T_{sim,h}(i))$. This then allows to determine the difference, or ratio, between the observed and modeled data to add or multiply this by the values of $m(T_{sim,p}(i))$ to be adjusted.

```

1  void CMethods::Linear_Scaling(
2      float* output, float* m_obs, float* m_simh, float* m_simp,
3      unsigned n_time, std::string kind
4  ) {
5      const float
6          obs_mean = MyMath::mean(m_obs, n_time),           //  $m(T_{obs,h}(i))$ 
7          simh_mean = MyMath::mean(m_simh, n_time);         //  $m(T_{sim,h}(i))$ 
8
9      if (kind == "add" || kind == "+") {
10         for (unsigned ts = 0; ts < n_time; ts++)
11             output[ts] = m_simp[ts] + (obs_mean - simh_mean); // Eq. 4.9
12     } else if (kind == "mult" || kind == "*") {
13         for (unsigned ts = 0; ts < n_time; ts++)
14             output[ts] = m_simp[ts] * (obs_mean / simh_mean); // Eq. 4.11
15     }
16 }

```

Listing 4.23: Implementation of the additive and multiplicative linear scaling method in C++

Here, too, the results of the calculations are stored with the help of the variable `ts` at the corresponding index of the array to which the pointer `output` points. One of the strengths of C++ shows up here because while Python must initialize and return almost always and

everywhere new variables, which possess all also a multiplicity of attributes and functions, with again own variables, it is possible with C++ to initialize pointers, which point to certain memory addresses so that their values can be set and read, without the entire contents of the array or also object must be passed between different functions.

Variance Scaling

The variance scaling procedure for temperatures requires the implementation of the additive linear scaling procedure since this is an extension of the linear scaling procedure in that it adjusts the long-term monthly variance and standard deviation in addition to the long-term monthly mean. Therefore, in the first step, the additive linear scaling procedure is run for both the modeled control period `m_simh` and the time series to be adjusted `m_simp`. Here it can be seen that two float arrays are previously declared, which are then passed to the first parameter output of the function `Linear_Scaling`. Since the `Linear_Scaling` and `Variance_Scaling` functions are in the same class and declared with `static`, the class name does not need to be prefixed when calling the `Linear_Scaling` function.

```

1   void CMethods::Variance_Scaling(
2       float* output, float* m_obs, float* m_simh, float* m_simp,
3       unsigned n_time
4   ) {
5       float LS_simh[n_time], LS_simp[n_time];
6       // Lst. 4.23
7       Linear_Scaling(LS_simh, m_obs, m_simh, m_simp, n_time);
8       Linear_Scaling(LS_simp, m_obs, m_simh, m_simp, n_time);
9
10      float
11          LS_simh_mean = MyMath::mean(LS_simh, n_time),
12          LS_simp_mean = MyMath::mean(LS_simp, n_time);
13
14      float VS_1_simh[n_time], VS_1_simp[n_time];
15      for (unsigned ts = 0; ts < n_time; ts++) {
16          VS1_simh[ts] = LS_simh[ts] - LS_simh_mean;           // Eq. 4.16
17          VS1_simp[ts] = LS_simp[ts] - LS_simp_mean;          // Eq. 4.17
18      }
19
20      float
21          obs_sd = MyMath::sd(m_obs, n_time),
22          VS1_simh_sd = MyMath::sd(VS1_simh, n_time);
23
24      float VS2_simp[n_time];
25      for (unsigned ts = 0; ts < n_time; ts++) {
26          VS2_simp[ts] = VS1_simp[ts] * (obs_sd / VS1_simh_sd); // Eq. 4.19
27          output[ts] = VS2_simp[ts] + LS_simp_mean;           // Eq. 4.20
28      }
29  }

```

Listing 4.24: Implementation of the variance scaling method in C++

After the modeled time series are linearly adjusted, the mean is subtracted from each of them (l. 15ff.). This gives these two time series a mean of zero (cf. Eq. 4.18). After both the standard deviation of the observed time series, as well as for the modeled data of the control period are determined (l. 20ff.), the Equations 4.19f. can be applied starting at line 25 by iterating again over all time steps, scaling the standard deviation, and then adding back the previously subtracted mean. The results are pushed into the array pointed to by the pointer output, as in the previously described methods. This can be done in one step, but is presented separately for a better understanding of the equations and their execution.

Execution of the scaling-based Bias Adjustment Techniques implemented in C++

If the methods `Delta_Method`, `Linear_Scaling` and `Variance_Scaling` (cf. Lst. 4.22ff.) are implemented in the program code, it is necessary for the following listing that this program can take arguments. These arguments are required to pass the necessary parameters, such as the input and output paths, as well as the selected method to the program start.

The first step is to define the method and the type of adjustment. In this example, the additive linear scaling method is chosen. Then a loop is created again, which iterates over the 12 months. Within this loop, the program in which the previously described methods are implemented is executed. Additionally, the required input parameters are passed to the program call. For this it is necessary, as shown in Listing 4.21, to have monthly separated data sets. In this case these are located in the respective temporary directory.

```

1  method="linear_scaling"; kind="+"           \ # define method and kind
2  for (( month=1; month<13; month++ )); do
3      ./adjustment_program                   \ # execute the program
4          --obs "tmp_obs/${month}.nc"        \ #  $m(T_{obs,h}(i))$ 
5          --simh "tmp_simh/${month}.nc"      \ #  $m(T_{sim,h}(i))$ 
6          --simp "tmp_simp/${month}.nc"      \ #  $m(T_{sim,p}(i))$ 
7          --method $method                   \ # set method
8          --kind $kind                       \ # "+" or "*"
9          --variable tas                      \ # variable to adjust
10         --output "tmp/${month}_${method}.nc" & \ # output file path
11     done
12     wait
13
14     # Merge LS-adjusted data sets
15     cdo -f nc                               \ # force .nc output
16         -mergetime "tmp_path/"*${method}.nc \ # merge all months
17         "tas_${method}_kind-${kind}.nc"     \ # output file path
18     for d in "${tmp_paths[@]"; do rm -rf $d; done \ # cleanup

```

Listing 4.25: Execution of monthly separated adjustment and merge to final adjusted data set using Bash and cdo

After each month has been adjusted separately and its adjusted time series has been saved in the `tmp` directory, these separate data sets must be merged. For this purpose, the `cdo`

tool can be used again. Its `mergetime` method allows to merge several data sets using the time dimension (cf. Lst. 4.25l. 15f.). Once this is done, the directory where this script has been executed contains a data set named `tas_linear_scaling_kind++.nc` and contains the fully adjusted time series. After that, the previously generated monthly separated data is no longer needed and will be deleted with the created directories in line 18.

Since the adaptation of the separated months can be done independently, parallel execution using “&” (cf. Lst. 4.25l. 10) is also possible here. The complete implementation can be found in the prepared repository of the C++ implementation in Table 1.2.

4.7.2. Distribution-based Bias Adjustment Techniques in C++

The quantile and quantile delta mapping methods do not require separation at equal months, but additional methods are needed here to determine, for example, the probability density function, as well as the cumulative distribution function.

First, however, as in the Python implementation, a monotonically increasing vector `v_xbins` is created. This vector has also the function to divide the value range of the observed and modeled time series of the control period into a defined number of equally large distances. This is necessary to determine the *PDF* of a time series on the one hand, on the other hand this vector is also used for the interpolation.

(Lst. 4.26) The function `get_xbins` is passed two pointers, each pointing to a float array. In addition, two further values are passed, which define on the one hand the number of quantiles to be considered, and on the other hand the length of the arrays. Within the function, the minimum and maximum of the values pointed to by `a` and `b` are determined to divide this range of values into `n_quantiles` of equal size. For this purpose, a vector of the data type `double` is created. This data type makes it possible to map larger numbers than `float`. This is important later for the interpolation in order to achieve particularly accurate results.

```

1   std::vector<double> CMethods::get_xbins(
2       float* a, float* b, unsigned n_quantiles, unsigned length
3   ) {
4       const double
5           a_max = *std::max_element(a, a + length),
6           a_min = *std::min_element(a, a + length),
7           b_max = *std::max_element(b, b + length),
8           b_min = *std::min_element(b, b + length);
9
10      const double
11          global_max = std::max(a_max, b_max),
12          global_min = std::min(a_min, b_min);
13      const double wide = std::abs(global_max - global_min) / n_quantiles;
14
15      std::vector<double> v_xbins(0);
16      v_xbins.push_back(global_min);
17      while (v_xbins[v_xbins.size() - 1] < global_max)
18          v_xbins.push_back(v_xbins[v_xbins.size() - 1] + wide);

```



```

19     return v_xbins;
20 }

```

Listing 4.26: Function to create a monotonically increasing double vector

The vector `v_xbins` is returned at the end of the function.

Listing 4.27 contains the function which allows determining the probability density function of the array pointed to by the pointer `arr`. For this, a vector is needed which specifies how many bins should be used. In addition, the length of the array is also passed.

The first step of the `get_pdf` function creates the vector, which at the end of the function contains integer values that indicate how many values of the array pointed to by the `arr` pointer are to be found in each bin.

```

1     std::vector<int> MyMath::get_pdf(
2         float* arr, std::vector<double> bins, unsigned length
3     ) {
4         std::vector<int> v_pdf(bins.size() - 1);
5         for (unsigned ts = 0; ts < length; ts++) {
6             for (unsigned i = 0; i < v_pdf.size() - 1; i++) {
7                 if (i == 0 && arr[ts] <= bins[i]){
8                     ++v_pdf[i]; break;
9                 } else if (arr[ts] >= bins[i] && arr[ts] < bins[i + 1]){
10                    ++v_pdf[i]; break;
11                } else if (i == v_pdf.size() - 2 && arr[ts] >= bins[i + 1]){
12                    ++v_pdf[i + 1]; break;
13                }
14            }
15        }
16        return v_pdf;
17    }

```

Listing 4.27: Function to create the probability density function in C++

These values are determined by iterating over the length of the array in a loop. Within this loop, it is then checked whether a value is to be found within a bin. If this is the case, the value of this bin is incremented by one and jumps out of the inner loop. If all values are assigned, the vector `v_pdf` is returned at the end. This now contains the y-values of the *PDF* of the array pointed to by the pointer `arr`. This function is comparable to the Numpy function `histogram(a, bins, range)`.

(Lst. 4.28) The last function to highlight, which is needed for distribution-based bias adjustment procedures, is the `get_cdf` method. This method allows to determine the cumulative distribution function of an array pointed to by the pointer `arr`. To do this, the probability density function is first determined by calling the previously defined method `get_pdf` (cf. Lst. 4.27), with the appropriate arguments. Subsequently, the sum of the values of the previous indexes is stored at each index. This function also returns the created vector at the end.

```

1  std::vector<int> MyMath::get_cdf(
2      float* arr, std::vector<double> bins, unsigned length
3  ) {
4      std::vector<int> v_pdf = MyMath::get_pdf(arr, bins, length);
5      std::vector<int> v_cdf(v_pdf.size() + 1);
6      v_cdf[0] = 0;
7      for (unsigned i = 0; i < v_pdf.size(); i++)
8          v_cdf[i + 1] = v_cdf[i] + v_pdf[i];
9      return v_cdf;
10 }

```

Listing 4.28: Function to create the cumulative distribution function in C++

Once all these functions are defined, the implementation of the quantile and quantile delta mapping procedures can begin. Before that, however, it is important to mention that the following implementations are optimized for oscillating values, such as air temperatures. Since this thesis only deals with air temperatures at 2m height, no variant of the methods is designed for precipitation or even wind speed. The difference is that time series with precipitation have an exponentially increasing *CDF* at the beginning, whose slope stagnates when advancing into the positive value range. This offers the possibility to extrapolate at the positive end of the value range of the vector `v_xbins`. For temperatures, however, this can produce unrealistic values.

Quantile Mapping

The quantile mapping procedure for the adjustment of a time series has a similar workflow in C++, as in the programming language Python.

(Lst. 4.29) Here in the first step a vector is determined, which determines the probability boundaries `v_xbins` (l. 5ff.). This can be determined with the previously defined function `get_xbins` and the corresponding parameters. Once this is done, the *CDFs* of the observed and modeled data of the control period must be determined by calling the function `get_cdf` in each case (l. 10ff.). Since this vector has values of the data type `int` but is to be interpolated with these, equivalent vectors of the data type `double` are subsequently declared and initialized (l. 15ff.).

```

1  void CMethods::Quantile_Mapping(
2      float* output, float* obs, float* simh, float* simp,
3      unsigned n_time, unsigned n_quantiles
4  ) {
5      std::vector<double> v_xbins = get_xbins(
6          obs, simh, n_quantiles, n_time
7      );
8
9      // create CDFs
10     std::vector<int>
11         v_obs_cdf = MyMath::get_cdf(obs, v_xbins, n_time), // F_obs,h
12         v_simh_cdf = MyMath::get_cdf(simh, v_xbins, n_time); // F_sim,h

```

```

13
14 // create CDF vectors of type double
15 std::vector<double>
16     obs_cdf(v_obs_cdf.begin(), v_obs_cdf.end()), // Fobs,h
17     simh_cdf(v_simh_cdf.begin(), v_simh_cdf.end()); // Fsim,h

```

Listing 4.29: Implementation of the quantile mapping bias adjustment procedure in C++ (Part 1)

Now the exciting part begins; the determination of the quantile values of each value of $T_{sim,p}$ inserted in $F_{sim,h}$ (cf. Lst. 4.30l. 2ff.). This can be used to determine which temperature value of the time series to be adjusted corresponds to which quantile in the modeled data of the control period. For this the function `interpolate` from the class `MyMath` is used. This contains the implementation of linear interpolation, without extrapolation. Here, the first two parameters correspond to two vectors containing the x- and y-values of the function between which the last parameter is to be interpolated. This implementation of the linear interpolation function is not shown here.

```

1 // compute Fsim,h[Tsim,p]
2 std::vector<double> q_values;
3 for (unsigned ts = 0; ts < n_time; ts++)
4     q_values.push_back( // Eq. 4.33; Fig.4.3
5         MyMath::interpolate(v_xbins, simh_cdf, (double)simp[ts]
6     );
7
8 // insert Fsim,h[Tsim,p] into Fobs,h-1
9 for (unsigned ts = 0; ts < n_time; ts++)
10    output[ts] = (float)MyMath::interpolate( // Eq. 4.33; Fig.4.4
11        obs_cdf, v_xbins, q_values[ts]
12    );
13 }

```

Listing 4.30: Implementation of the quantile mapping bias adjustment procedure in C++ (Part 2)

Once all the values of `simp` are inserted into the cumulative distribution function `simh_cdf`, the vector `q_values` results, which contains the quantiles of the temperature values of the time series to be adjusted corresponding to the *CDF* of the modeled data of the control period.

In Listing 4.30, starting at line 9, iterating again over all time steps is needed to insert these previously determined values into the inverse *CDF* of the historical observations. The results are inserted into the final array at the appropriate index, using the pointer `output`. Thus, this function has no return value. Here, a conversion from `double` to `float` is done, since temperature data is usually stored in the `float` data type.

Quantile Delta Mapping

The most complex of the methods presented here for adjusting bias in climate data is the quantile delta mapping technique. This is also implemented in the C++ programming language and offers the possibility to adjust a time series similar to the quantile mapping,

but the crucial difference is that as described in Section 4.5, the change of the quantiles between the modeled time series of the control period and the time series to be adjusted is taken into account.

(Lst. 4.31) The first steps in quantile delta mapping are very similar to QM because the probability boundaries `v_xbins` are also determined to subsequently determine the cumulative distribution functions. In QDM, however, the *CDFs* of the control period, as well as those of the time series to be adjusted, are needed. These are determined using the function `get_cdf` defined at the beginning, starting at line 10, and then stored in vectors whose values are of type `double` (l. 16ff.).

```

1  void CMethods::Quantile_Delta_Mapping(
2      float* output, float* obs, float* simh, float* simp,
3      unsigned n_time, unsigned n_quantiles
4  ) {
5      std::vector<double> v_xbins = get_xbins(
6          obs, simh, n_quantiles, n_time
7      );
8
9      // create CDFs
10     std::vector<int>
11         v_obs_cdf = MyMath::get_cdf(obs, v_xbins, n_time),           // F_obs,h
12         v_simh_cdf = MyMath::get_cdf(simh, v_xbins, n_time),       // F_sim,h
13         v_simp_cdf = MyMath::get_cdf(simp, v_xbins, n_time);       // F_sim,p
14
15     // create CDF vectors of type double
16     std::vector<double>
17         obs_cdf(v_obs_cdf.begin(), v_obs_cdf.end()),               // F_obs,h
18         simh_cdf(v_simh_cdf.begin(), v_simh_cdf.end()),           // F_sim,h
19         simp_cdf(v_simp_cdf.begin(), v_simp_cdf.end());           // F_sim,p

```

Listing 4.31: Implementation of the quantile delta mapping bias adjustment procedure in C++ (Part 1)

Once these preparatory steps are done, the application of the Equations 4.34ff. can begin by determining ε , by inserting the values of $T_{sim,p}$ into $F_{sim,p}$ (cf. Lst. 4.32l. 3ff.). The interpolation function of the `MyMath` class used here is the same function that was used for the quantile mapping in Listing 4.30. This means that the first two parameters are the x- and y-values of the function, between which the third parameter is interpolated linearly, without extrapolation. After all values of epsilon (i.e. ε) are determined, this vector can be used to determine a new vector `QDM1`, which corresponds to an adjusted time series based only on the observed time series, as well as the time series to be adjusted. For this purpose, interpolation is performed again to insert the previously determined values of epsilon into the inverse *CDF* of the observations of the control period (l. 10ff.).

```

1  // calculate exact CDF values (F_{sim,p}[T_{sim,p}], i.e. ε)
2  std::vector<double> epsilon;
3  for (unsigned ts = 0; ts < n_time; ts++)

```

```

4     epsilon.push_back(                                     // Eq. 4.34
5         MyMath::interpolate(v_xbins, simp_cdf, simp[ts])
6     );
7
8     // insert  $\varepsilon$  in inverted  $F_{obs,h}$ 
9     std::vector<double> QDM1;
10    for (unsigned ts = 0; ts < n_time; ts++)
11        QDM1.push_back(                                     // Eq. 4.35
12            MyMath::interpolate(obs_cdf, v_xbins, epsilon[ts])
13        );
14
15    // compute and add  $\Delta$ 
16    for (unsigned ts = 0; ts < n_time; ts++){
17        double delta = simp[ts] - MyMath::interpolate(     // Eq. 4.36f.
18            simh_cdf, v_xbins, epsilon[ts]
19        );
20        output[ts] = (float)(QDM1[ts] + delta);           // Eq. 4.38
21    }
22 }

```

Listing 4.32: Implementation of the quantile delta mapping bias adjustment procedure in C++ (Part 2)

(Lst. 4.32) Finally, the quantile delta mapping procedure determines Δ for each time step by subtracting $F_{sim,h}[\varepsilon(i)]$ from each value of the time series to be adjusted (simp; l. 16ff.). This difference is then added in line 20 to the corresponding value of the vector QDM1, adjusted in lines 10ff. and stored in the final array using the pointer output and the iteration variable ts. In the following subsection is a pseudocode example that shows how this method can be used.

Execution of the distribution-based Bias Adjustment Techniques implemented in C++

Once all required methods and functionalities for the execution of the distribution-based bias adjustment procedures are implemented, the adjustment can begin. The following listing shows an example of how these methods can be called. First, the number of quantiles to be considered as well as the length of the time series are defined, and then the input data sets are loaded. This is shown in a very simplified way for demonstration purposes.

```

1     ...
2     int main(int argc, char** argv){
3         unsigned
4             n_quantiles = 100,
5             n_time = 10950;
6         float
7             adjusted_data[n_time], // declare array to store adjusted data
8             obs = load("obs.nc"), // load data set into program
9             simh = load("simh.nc"), // ----
10            simp = load("simp.nc"); // ----
11 }

```

```
12     // apply the quantile delta mapping method
13     CMethods::Quantile_Delta_Mapping(
14         adjusted_data, obs, simh, simp, n_time, n_quantiles
15     );
16     // adjusted_data now contains the QDM bias-adjusted time series
17     ...
18     return 0;
19 }
```

Listing 4.33: Pseudocode to demonstrate the quantile delta mapping function call

(Lst. 4.33) After executing the `Quantile_Delta_Mapping` function of the `CMethods` class, the `adjusted_data` array has the fully bias-adjusted time series. It should be noted here that all input data sets must have exactly `n_time` entries. This approach can also be applied to the scaling-based adjustment methods presented earlier, as long as the corresponding parameters of the functions as well as the monthly separations are taken into account.

4.8. Method Discussion

All five presented methods aim at minimizing the bias between modeled and observed climate data of a past or future period, based on past observed and modeled time series. This means that the methods are only calibrated based on past simulated and observed data and can then be applied to predicted and also past modeled data. All methods also assume that there is a relationship between modeled and observed data, as well as between the historical and future time series, in that both the modeled and the observed data represent the same region (i.e. grid box), which is subject to similar to identical topographic and atmospheric conditions.

The methods differ in their approach, because the delta method, as well as the linear and variance scaling method try to minimize the bias directly by scaling mean values. The quantile and quantile delta mapping methods, on the other hand, try to achieve the adjustment on the basis of changed distributions by not scaling values but redefining them according to historical distribution properties. The linear and variance scaling methods, as well as the quantile mapping method also try to project the bias between the historical observed and modeled data onto the predicted time series. The delta method, as well as the quantile delta mapping method, on the other hand, also try to take into account the change between the modeled data at different periods.

(Equations 4.12ff., 4.21ff., 4.39ff. and Figures 4.3f.) During the exemplary execution of the methods for adjusting the temperature on December 17, 1997, it was noticed that all methods, except for the delta method, produced a warmer adjusted temperature value. The delta method is also the only method that includes a specific value from the control period ($T_{obs,h}(i)$). This value, which corresponds to the index 6190 and is assigned to the December 17, 1967 in the control period, thus has a very high influence on the result, although it only originates from the observations of the past. Thus, if there is no change between the modeled data of the control and predicted period, the result will always be the historically observed value at the corresponding index.

5. Results

In each of the adjustment procedures presented, historical observed and modeled daily mean 2m air temperatures of the period from 1951 to 1980 inclusive are used to adjust values of modeled time series of the period from 1981 to 2010 inclusive. This adjustment is applied per grid cell. This means that each adjustment procedure is performed 5550 times to fully adjust the modeled data set of the MPI-ESM1-2-HR with a resolution of 111 x 50 cells (longitude/latitude), presented in Section 3.1.

This is achieved by applying the methods discussed in Chapter 4. Due to the better performance of C++, the presented methods are implemented in this language to perform the adjustment for each grid box (i.e. spatial cell). Excerpts are shown in Section 4.7. The complete source code of the bias adjustment techniques, as well as usage examples can be found in the public repositories listed in Table 1.2.

The scaling-based adjustment techniques are carried out in such a way that the same months are adjusted together. This means that the results of the methods are not dependent on individual months, but are calculated based on long-term monthly means and standard deviations. Code snippets are shown in Section 4.7.1.

Quantile and quantile delta mapping were performed as shown in Section 4.7.2 by adjusting the full period without monthly separation. Furthermore, a value of 100 was passed to the respective method for the parameter `n_quantiles`.

In the first part of the evaluations, various statistical methods are applied to check the results and the success of the implemented methods. In the second part, further analyses will be performed in which individual time series will be compared, and the effect on quantiles and distributions will be addressed. In addition, it will be determined whether and to what extent correlations between locations within a data set are affected by the procedures and how this affects the teleconnections described later.

5.1. Method Evaluation

To investigate the effects of the adjustment techniques, three statistical methods are used to compare the results of the adjustment procedures. The aim is to show whether and to what extent the deviations between modeled, observed and adjusted data have changed. This should provide information on whether the procedures were applied correctly, whether an adjustment was necessary at all, and whether the adjusted data resemble the values of the observations more closely than the unadjusted, modeled data do.

In addition to considering Europe as a whole, five regions are also defined to examine in detail how far the error between observed and modeled data differs before and after the adjustment. Thus, one does not only have roughly averaged data of the whole of Europe and its surroundings but can examine the effects at different locations in more detail. These regions are listed in the table below.

	Oslo (NO)	Bremerhaven (DE)	Zurich (CH)	Milan (IT)	Lyon (FR)
Longitude	10.7579°E	8.5833°E	8.5392°E	9.1885°E	4.8343°E
Latitude	59.9115°N	53.5499°N	47.3686°N	45.4647°N	45.7634°N

Table 5.1.: Coordinates of selected locations in Europe

Since the selected locations are not in the center of a grid box in the data sets, the closest box is used for the investigation. The following table shows the coordinates used.

	Oslo (NO)	Bremerhaven (DE)	Zurich (CH)	Milan (IT)	Lyon (FR)
Longitude	11°E	9°E	9°E	9°E	53°E
Latitude	60°N	54°N	47°N	45°N	46°N

Table 5.2.: Coordinates of the nearest cell of selected locations

The selected location Oslo covers the grid cell from 11°E to 12°E and 60°N to 61°N, since all grid boxes in this data are 1.0° x 1.0° longitude/latitude. This region thus covers much more than just the city of Oslo, but in the further course, only the respective city name will be referenced when talking about the selected grid cells defined in Table 5.2.

When Europe is referred to in subsequent studies, e.g., in tables and descriptions, the entire region from -45°W to 65°E, 22°N to 72°N as described in Chapter 3 is meant.

In the analysis of climate data, the weighted mean is often used, because the surface area of grid cells in data sets becomes smaller the further away they are from the equator, due to the curvature of the earth's surface. A weighting of the values then ensures that values or also the mean values of such cells, which have a smaller earth surface, are less important if for example the mean is determined over a large region (e.g. Fig. 3.4).

This weighting of an individual cell can be determined by the following equation:

$$w(T_{sim,p}) = \cos\left(x \cdot \frac{\pi}{180}\right) \quad (5.1)$$

where:

$$x = \text{Latitude of 1D sequence } T_{sim,p}$$

If the time series $T_{sim,p}$ is on the latitude 23°N, then a weighting of $w = 0.92$ results. For very high or low latitudes, this weighting decreases. To determine the weighted mean temperature value of a data set that has time series at different geographic locations, the regular mean value of each time series must be weighted according to its geographic

location.

$$A = \begin{bmatrix} \mu T_{90,-160} & \dots & \mu T_{90,160} \\ \vdots & \ddots & \vdots \\ \mu T_{-90,-160} & \dots & \mu T_{-90,160} \end{bmatrix} \quad (5.2)$$

$$\mu A = \frac{\sum_{i=-90}^{90} \sum_{j=-160}^{160} A_{i,j} \cdot w(A_{i,j})}{\sum_{i=-90}^{90} \sum_{j=-160}^{160} w(A_{i,j})} \quad (5.3)$$

where:

$A_{i,j}$ = Temperature at a specific longitude and latitude in A

The value of each cell of matrix A corresponds to the mean value of the temperatures of a grid cell over the entire observation period (e.g. 30 years). This mean value can be determined on a daily, but also a monthly or even annual basis. By applying Equation 5.3, values in cells covering only a fraction of the earth's surface have a smaller impact on the overall mean value of A than cells covering a larger region.

However, since bias adjustments are performed for each individual cell and it should be determined which effects these procedures have taken on the data, it does not make sense, in this case, to use weighting when determining global values, since possible inaccuracies in higher latitudes are not sufficiently taken into account here.

The `cdo` tool takes this weighting into account. To disable this behavior, `weights=FALSE` can be passed to the respective calculation argument.

```

1   cdo -output                \ # enable command line output
2   -fldmean,weights=FALSE    \ # mean without spatial weight
3   mbe_no_correction_19810101_20101231.nc \ # input file

```

Listing 5.1: Compute the European unweighted mean

5.1.1. Mean Bias Error

The mean bias error (*MBE*) is a statistical method that is suitable for estimating the mean error between two time series. Thus, it can be recognized in advance whether an adjustment has to be made or whether the data of the observed and modeled time series show high similarity in the mean. Since the *MBE* only calculates the mean error, it does not provide information on how strong individual deviations occur. Thus, extreme values are not sufficiently represented, which is why partly high differences can result in a low *MBE*. For this reason, this method is usually only used to estimate whether an error exists and whether it is less or greater than zero. The following equation can be used to calculate the mean bias error between two time series:

$$MBE(T_{obs,p}, T_{sim,p}) = \frac{1}{n} \sum_{i=1}^n (T_{sim,p}(i) - T_{obs,p}(i)) \quad (5.4)$$

To determine the *MBE* on a daily basis, the difference between the modeled or bias-adjusted and observed value is taken for each day of the time series from 1981 to 2010 inclusive. This is followed by dividing by the number of all entries. Thus, a negative *MBE* means that the model is simulating temperature values that are too cold on average. A positive value, on the other hand, means that the model-generated (or bias-adjusted) data includes temperatures that are too warm on average. If the value is close to zero, the data are very similar on average, but can still differ greatly. The unit of *MBE* calculated here is degrees Celsius, since all data sets used here have temperature values in this unit.

Method	Oslo	Bremerhaven	Zurich	Milan	Lyon	Europe
No Correction	-1.4858	-1.1661	-0.2571	-1.5312	-1.2646	-0.7881
DM	-0.7317	-0.6923	-0.3087	-0.2739	-0.3797	-0.1551
LS	-0.7318	-0.6923	-0.3087	-0.2739	-0.3797	-0.1551
VS	-0.7318	-0.6923	-0.3087	-0.2739	-0.3797	-0.1551
QM	-0.7525	-0.6789	-0.3177	-0.3064	-0.3734	-0.1749
QDM	-0.7345	-0.6944	-0.3082	-0.2727	-0.3790	-0.1550

Table 5.3.: *MBE* of daily mean 2m air temperatures between observed, modeled and adjusted data respectively in deg. Celsius at different locations in Europe (1981-2010)

(Tab. 5.3) The mean bias error between the observed and raw modeled data is always negative. When considering Europe as a whole, the unweighted *MBE* has a value of -0.7881°C , which means that the model calculated values of almost one deg. Celsius colder on average than observed for the same period. This indicates that an adjustment of this data set is appropriate to minimize this deviation in the mean. After applying the adjustment procedures, the *MBE*, relative to Europe as a whole, has moved more than half a degree toward zero. This is the first sign of the successful implementation and application of the adjustment procedures. Furthermore, it is noticeable that the *MBE* for the locations Oslo, Bremerhaven, Milan as well as in Lyon have improved significantly. Only in Zurich has it deteriorated. The scaling-based methods have produced the same

results, except for Oslo, but they hardly differ from the results of the quantile and quantile delta mapping.

	No Cor- rection	DM	LS	VS	QM	QDM
Minimum	-5.8870	-3.1945	-3.1945	-3.1945	-3.1658	-3.1944
Mean	-0.7881	-0.1551	-0.1551	-0.1551	-0.1749	-0.1550
Maximum	5.7449	1.0348	1.0348	1.0348	1.0063	1.0336

Table 5.4.: Minimum, mean and maximum *MBE* of daily mean 2m air temperatures between observed, modeled and bias-adjusted data in deg. Celsius in Europe (1981-2010)

(Tab. 5.4) When looking at the minimal and maximal values of the *MBE*, related to Europe, it is noticeable that the unadjusted values not only show the highest deviation on average but also the minimal and maximal values deviate strongly from zero. The procedures, on the other hand, have been able to reduce this difference. It is interesting to note that the minimum and maximum of all *MBE* in the data adjusted by the quantile mapping method are closer to zero, although the mean *MBE* shows the highest distance to zero compared to the other methods.

The *MBE* related to the whole period from 1981-2010, whether calculated with daily or monthly means, has of course no great significance, because just as the climate behaves differently at different times, the error is also not the same at every season. For this reason, it makes sense to determine the long-term monthly *MBE* to find out in which months particularly high or low deviations are to be found:

Month	No Correction	DM	LS	VS	QM	QDM
Jan	-1.3403	-0.5404	-0.5404	-0.5404	-0.4812	-0.4740
Feb	-0.8770	-0.0993	-0.0994	-0.0994	-0.0175	-0.0090
Mar	-0.8966	-0.2675	-0.2675	-0.2675	-0.0781	-0.0662
Apr	-0.9951	-0.0017	-0.0017	-0.0017	-0.2675	-0.2509
May	-0.7343	0.1275	0.1275	0.1275	-0.1785	-0.1541
Jun	-0.5037	-0.0007	-0.0007	-0.0007	-0.1567	-0.1292
Jul	-0.3764	-0.0399	-0.0399	-0.0399	-0.2107	-0.1790
Aug	-0.3277	-0.0053	-0.0053	-0.0053	-0.1567	-0.1206
Sept	-0.5047	-0.0527	-0.0527	-0.0527	-0.0577	-0.0243
Oct	-0.7250	-0.3194	-0.3194	-0.3194	-0.0252	-0.0052
Nov	-1.0207	-0.2361	-0.2361	-0.2361	-0.1713	-0.1584
Dec	-1.1595	-0.4088	-0.4088	-0.4088	-0.2799	-0.2713

Table 5.5.: Mean *MBE* per month based on daily mean 2m air temperatures in Europe in deg. Celsius (1981-2010)

(Tab. 5.5) It can be seen that both the modeled and the adjusted data have the highest deviations from the observed data in the winter months. In February, April, and June to September, the *MBE* of the scaling-based methods, is even below 0.1°C. Furthermore,

despite their different approaches, the scaling-based methods almost consistently have the same values. This is due to their goal of adjusting the long-term monthly means. The fact that their daily values of individual locations and time series are not always the same due to the different approaches will be shown later.

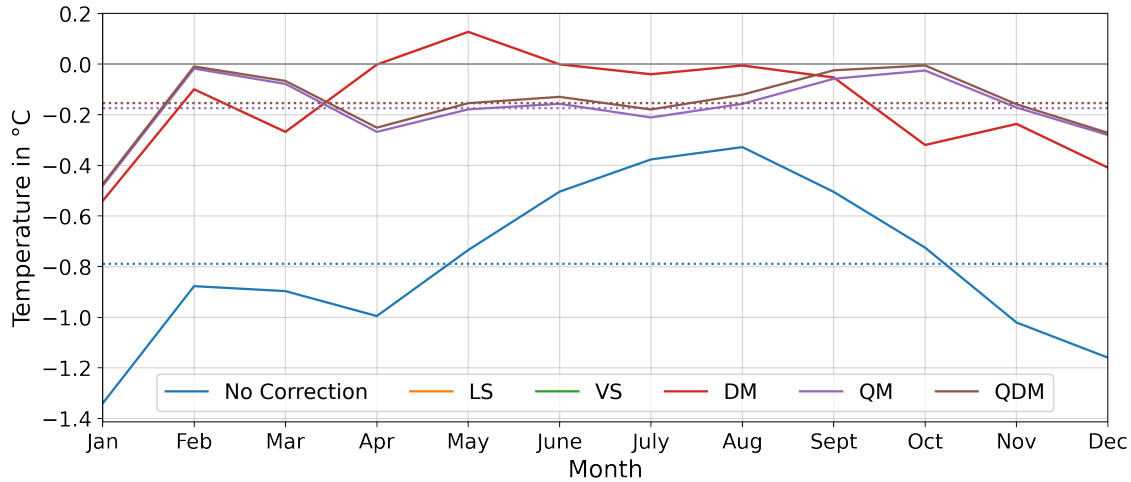


Figure 5.1.: Long-term monthly mean MBE in Europe based on daily mean 2m air temperatures (1981-2010)

Figure 5.1 serves as a visual representation of the Table 5.5 and shows once again how much the methods have reduced the mean bias error related to the whole of Europe and its surroundings. The scaling-based methods have the same values and therefore cannot be differentiated. The quantile and quantile delta mapping procedures also have very similar results, but those of the quantile delta mapping are minimally better, which can be seen from the closer mean to zero. The distribution of the bias has also changed. The unadjusted 2m air temperatures of MPI-ESM1-2-HR have a MBE with a range of about 1°C , whereas the bias-adjusted data sets have a range of only 0.67°C , with a little variation here from 0°C in the months of February through October.

Now it is clear that the procedures have changed the monthly mean temperatures. But how is this reflected in daily values? Figure 5.2 serves to illustrate this topic. Here the modeled (blue line), as well as observed (black dashed line) and DM-adjusted temperatures (red line), averaged to day of the year are shown. It is now clearly visible that the modeled data always have colder values on average than the observed data. This must also have been the case in the control period (1951-1980) because the delta method provided higher temperatures here. From April to the end of September, there are hardly any differences between the observed and the adjusted values. This has already been shown in Table 5.5. In winter, the highest differences between the adjusted and observed values can be seen. This may be due to the fact that the difference between the modeled winter months of the control and scenario period was somewhat smaller than the difference between the observed and modeled time series of the scenario period, which caused the delta method to add a too small difference to the predicted time series.

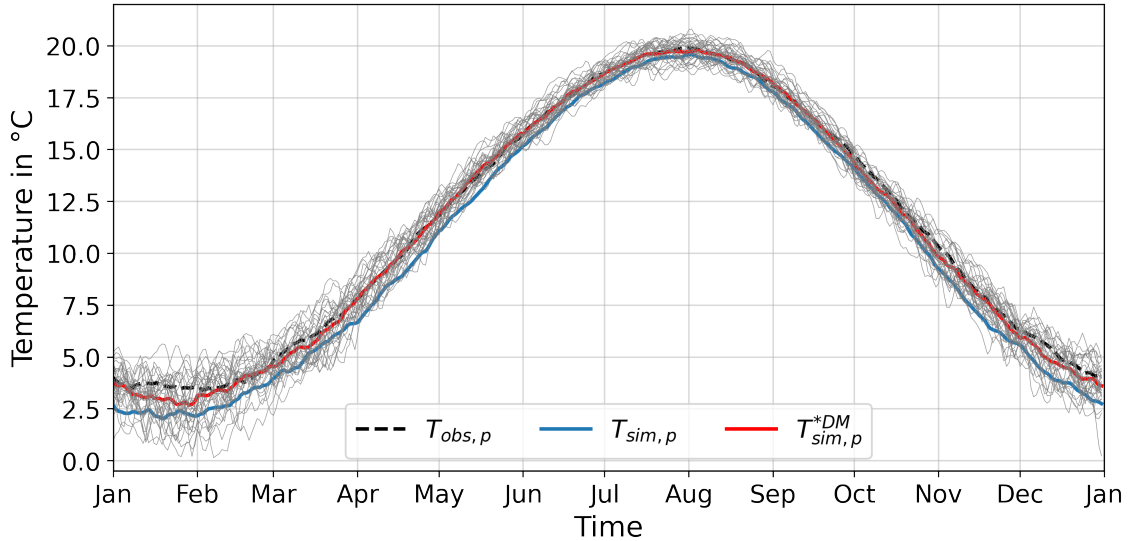


Figure 5.2.: Mean 2m air temperatures per day of the year in Europe between 1981 and 2010; grey lines represent the daily mean 2m air temperatures of each individual year in the DM-adjusted data set

This representation is of course also available for the other bias-adjusted data sets, but since these hardly differ, they are attached to the appendix (cf. Fig. A.1ff.). However, it is to be mentioned that with this representation method a conspicuousness has come, because by the monthly-separated adjustment of the temperatures, the first and last day of a month possess often a disproportionately high difference to each other, whereby it comes for example in Figure A.1 to jumpy behavior of the mean temperatures by day per year. This can be observed in early April, June, and October. Also in Figure A.2 this can be observed especially in April and October. This is due to the month-dependent scaling, which in the scaling-based techniques procedures a different value depending on the month, which is added to the predicted or observed time series (depending on the method), causing these deviations to be observed especially at the change of month. Thus, this is a weakness of these adjustment procedures.

At the beginning of this thesis, the quantile and quantile delta mapping procedures were also adjusted, just like the scaling-based methods, not with the entire period at once, but in monthly separated groups. Here, this conspicuousness was even more noticeable at almost every turn of the month. This is to be justified with the QM-technique in the fact that for example, an adjusted May can possess only temperature values from the historical observed value range of the May months (cf. Section 4.4), since the May possesses however on the average warmer temperatures than the April and the quantiles of the one month have nothing more in common with those of the other, it can lead so to strong deviations. Also, if the control period is too far in the past in relation to the period to be adjusted, or one of the periods is too long or too short, this error can be amplified, since the adjustment here is bound to the range of values of the observations of the control period. Nevertheless, it is worth mentioning here that the QM and QDM techniques were also tried for monthly separated time series and these partly yielded better values for the mean bias error and statistical procedures listed later. Since the monthly separation of the temperatures of

QM and QDM has not been a topic in any publication found so far, it has been omitted in this thesis.

As an aside, Figure 4.2 plots similar data to Figure 5.2, but it plots the monthly means of the linearly scaled data rather than the mean of the DM-adjusted data for the mean day of the year.

Figure 5.2 has shown that the DM-adjusted data set is very similar to those of the observed time series. But it was not possible to see exactly how much this difference is at certain points in time. For this reason, the difference between the modeled and observed, and the bias-adjusted and observed data was calculated. Again, this is done per day of the year, so that it can be shown, analogous to the previous figure when which deviations occurred.

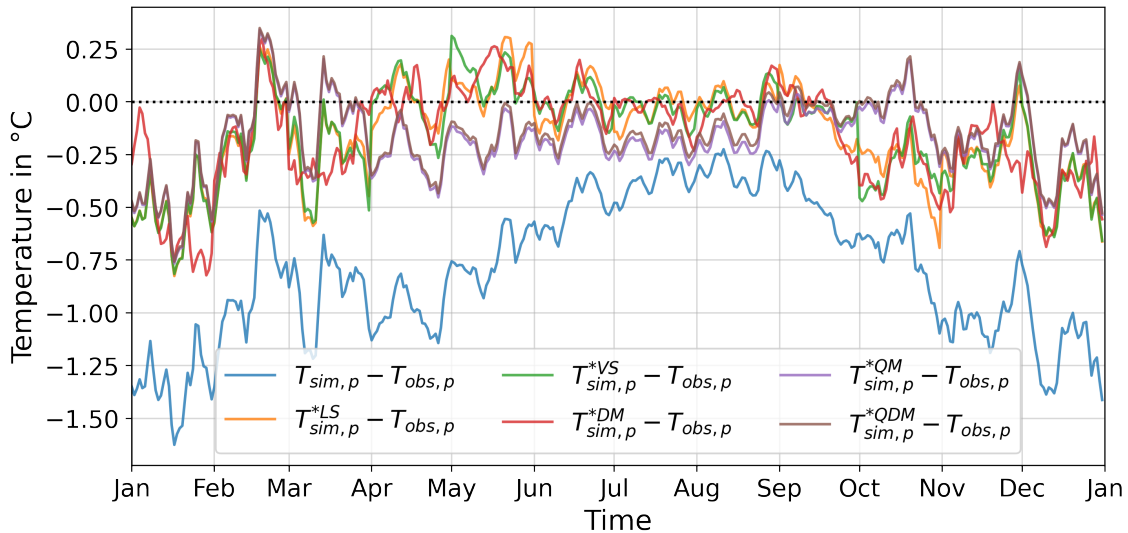


Figure 5.3.: Mean 2m air temperature difference between modeled and observed and bias-adjusted and observed per day of the year in Europe (1981-2010)

(Fig. 5.3) Now it can be seen how similar the mean deviations of the adjusted time series are. Especially the similarity between the quantile (violet line) and quantile delta mapping (brown line) results can be observed. The scaling-based methods have a smaller deviation from the observed data, especially in the summer months, this is also shown in Table 5.5. The linear and variance scaling methods have very similar values and intensities of the movements at many time points, the delta method, on the other hand, has produced a slightly different symmetry, even if it is not too different overall. In the winter months, as previously noted several times, the highest difference from the observed data can be seen. All the differences shown between the adjusted and observed time series, which represent the mean value per day of the year, are always smaller than the modeled data of the same period. The application of the adjustment procedures has provided significant improvements here.

(Fig. 5.4) When Europe is visualized on the map, it can be seen that the strongest discrepancies between modeled and observed data occur primarily in the North Atlantic. This may be due to the North Atlantic Oscillation, which may not be adequately accounted for in the model, and therefore colder temperatures are shown by the model.

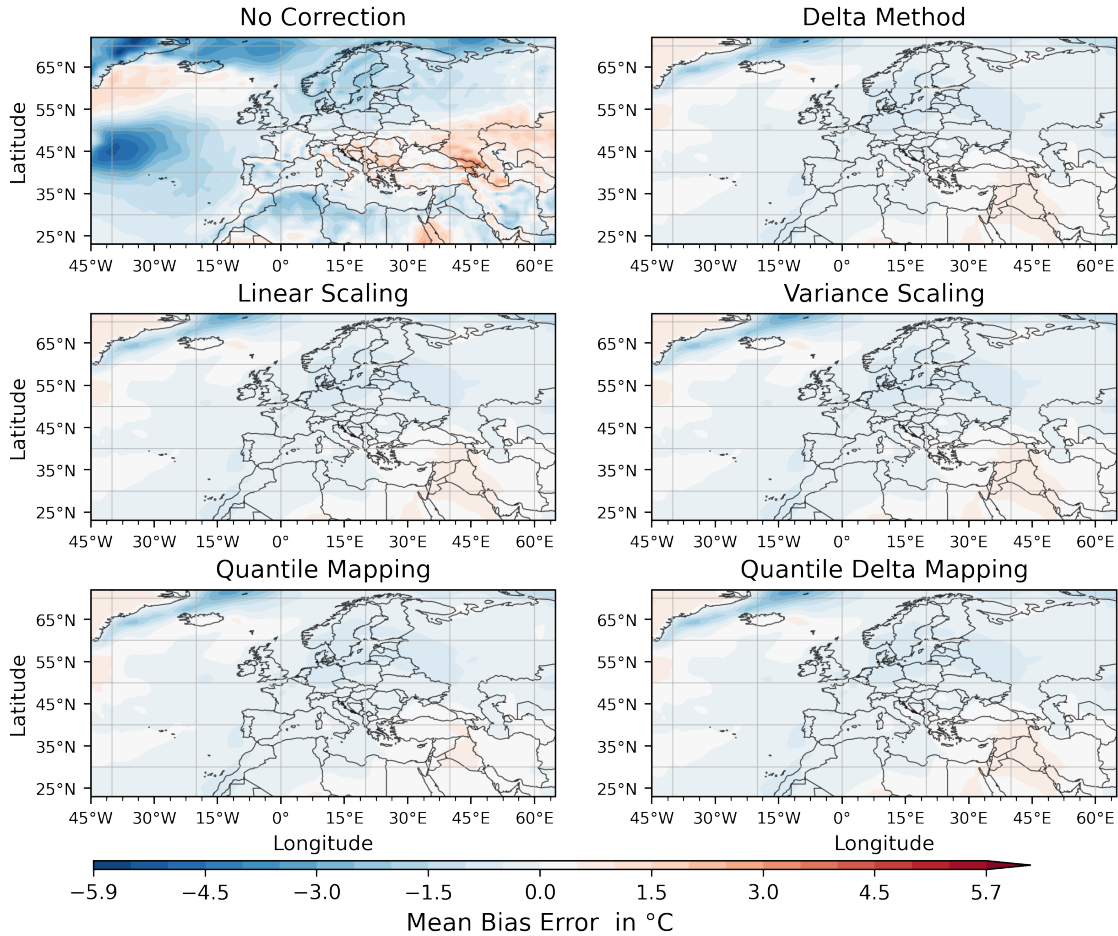


Figure 5.4.: Mean bias error of daily mean 2m air temperatures in Europe (1981-2010)

Also in the northern regions, Iceland, Greenland, as well as in the Scandinavian countries and North Africa, the modeled data show significantly colder temperatures than the observed data. The adjustment procedures have been able to reduce these errors significantly so that hardly any differences between the results of the procedures can be found here. Only in Islandsea are even higher deviations to be found. Since the minimum and maximum values of MBE vary greatly between modeled and adjusted values (cf. Tab. 5.4), which is why the land contours, which seem to range from -1°C to $+1^{\circ}\text{C}$, are hardly distinguishable, the separate consideration of the land and sea regions, as well as the adjustment of the color bar, is appropriate.

(Fig. 5.5) In contrast to Figure 5.4, it is particularly easy to see that the mean bias error at almost every location has a difference of less than 1°C . Regions whose mean bias error values are higher or lower than the maximum or minimum in the adjusted data sets are marked with yellow horizontal lines. This type of plot allows close examination of the change in mean errors. All methods give roughly similar results and have significantly reduced the deviations in many regions. Especially in the north of Africa, Iceland, and the Scandinavian countries, but also in central-eastern Europe and south of Russia the mean error has been reduced.

However, it is striking that in Greenland the previously too cold temperatures are now

somewhat too warm. This change of sign is since the differences between the modeled and observed data of the scenario period are no longer as large as they were in the control period so the adjustment procedures here provided too much compensation for the cold temperatures, which has now led to somewhat too warm values.

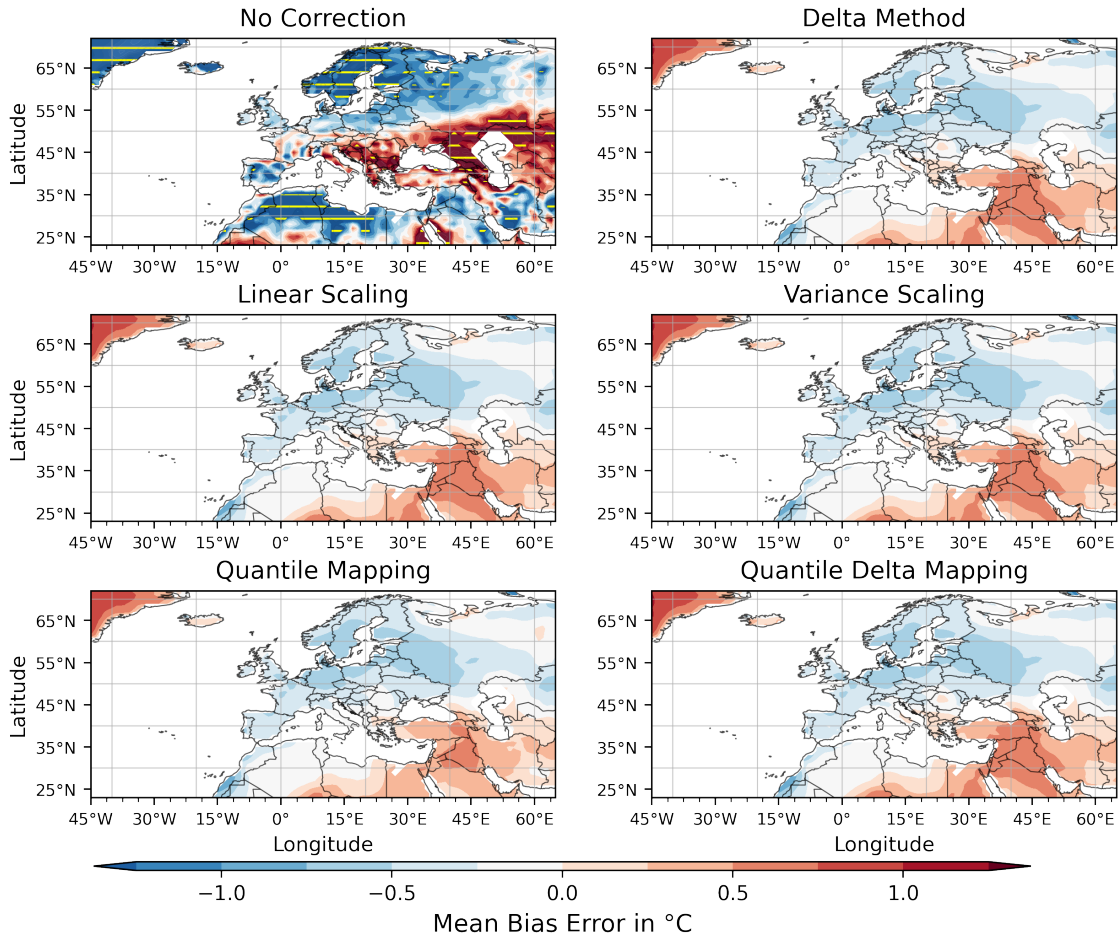


Figure 5.5.: Mean Bias Error of daily 2m air temperatures in Europe (1981-2010)

(Fig. 5.6) Not only on land but also the sea regions have, after applying the methods presented in Chapter 4, significantly lower mean deviations compared to the observed data. All methods have been able to visibly reduce the particularly high differences in the North Atlantic. Also above Iceland, the error was reduced, nevertheless, strong deviations of up to -3.25°C are found in all adjusted data sets. The coastal region of Greenland is too warm in the modeled data compared to the observed data, but too cold after the procedures were applied. This may indicate that there has been a large change there between the observed data of the 1951 - 1980 and 1981 - 2010 time series, which was not simulated in this form in the modeled time series.

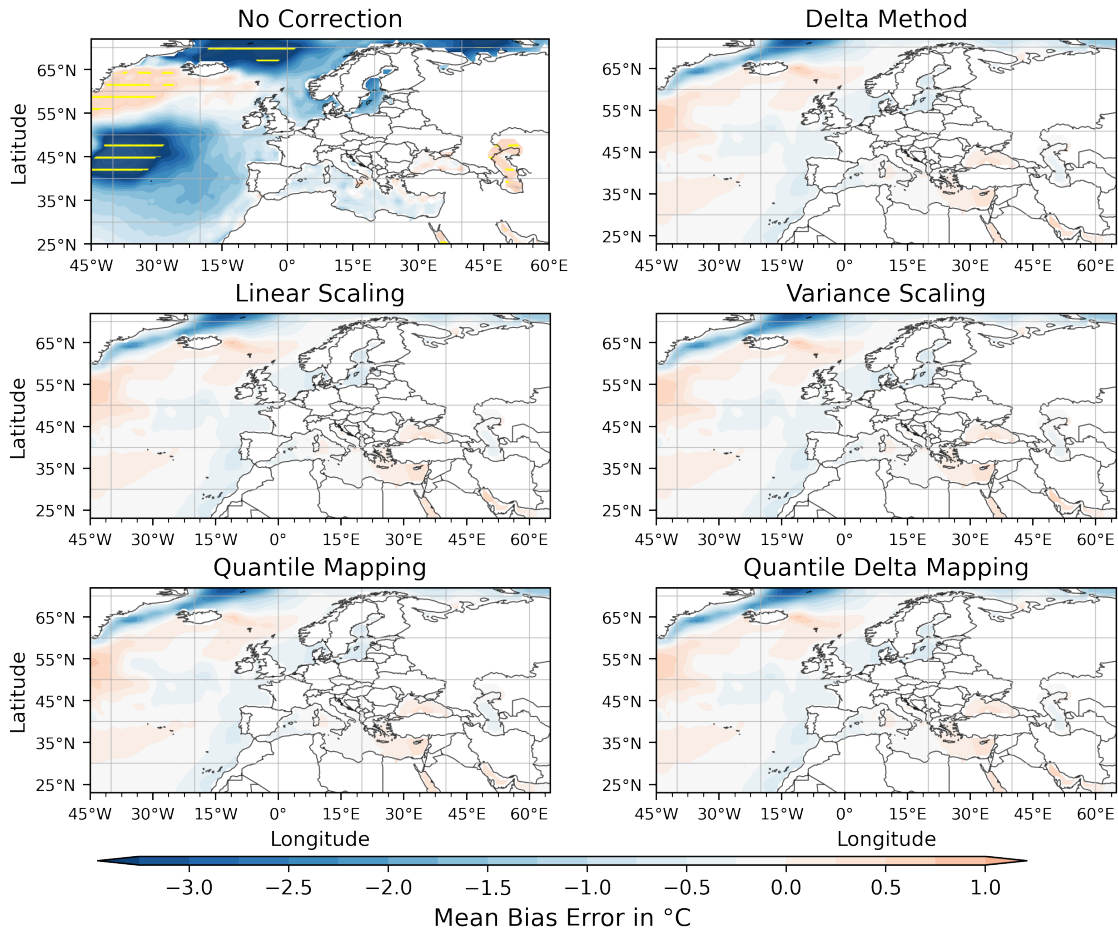


Figure 5.6.: Mean bias error of daily mean 2m air temperatures in Europe (1981-2010)

Interestingly, the air temperatures near Greenland have changed their sign in all adjusted data sets. This means that the observed data show significantly warmer values than the adjusted time series for the period 1981-2010. This may be due to the fact that the modeled data for the control period was also already too warm in this region (cf. Fig. 5.7).

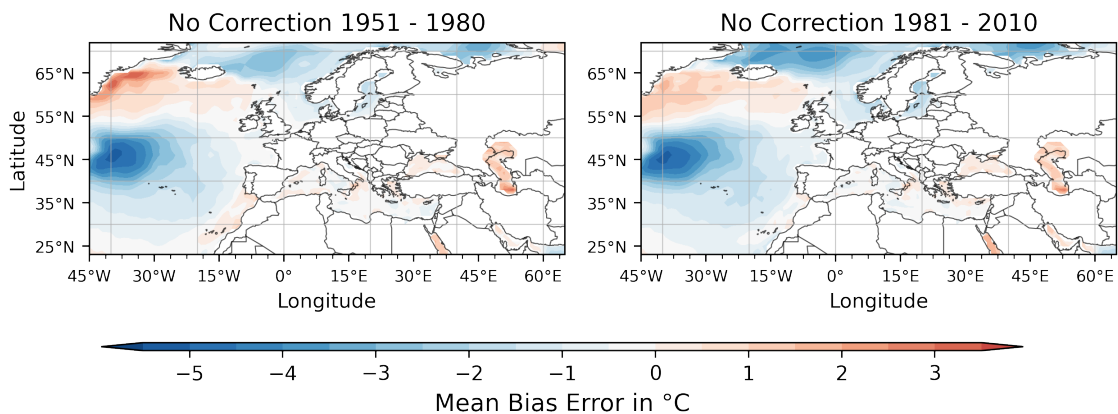


Figure 5.7.: Mean bias error of daily mean 2m air temperatures between modeled and observed data in the control and scenario period

(Fig. 5.7) Due to the already too warm temperatures of the modeled data of the control

period in this region, the methods have projected colder values onto the predicted time series. Climate change, which is causing temperatures to rise, especially in the polar regions, is also present in the modeled data and can cause the *MBE* of the adjusted data to become negative because the observed temperatures, which are now also warmer, are subtracted from the modeled temperatures, which have been cooled by the procedures. This can create a difference with a negative sign if the temperatures of the observed values are higher than those of the adjusted data.

It has been shown that the bias adjustment techniques have significantly reduced the error in the daily mean temperatures, as well as the long-term monthly means between the observed and modeled data related to the whole of Europe. The application of the scaling-based methods, have provided roughly similar results with respect to the mean bias error (cf. Tab. 5.4f. and Fig. 5.1ff.), but it has also been shown that these techniques can also change the previously too cold values to too warm values (cf. Fig. 5.5) or to warm values to too cold values (cf. Fig. 5.6).

Since the mean bias error cannot tell how strong individual deviations are due to particularly high or low temperatures on specific days, further steps are needed to examine the effects of the bias adjustment procedures on the data.

5.1.2. Root Mean Square Error

The Root Mean Square Error (*RMSE*) is, like the *MBE*, a statistical method that can be used to measure the errors between two time series and to summarize them in one value. These errors are again the differences between a modeled or bias-adjusted and an observed time series. The *RMSE* is used as a measure of accuracy because the root is taken from squared differences. Thus, in contrast to the *MBE*, particularly high differences between observed and modeled or predicted data (at a given time) can also be weighted disproportionately.

$$RMSE(T_{obs,p}, T_{sim,p}) = \sqrt{\frac{1}{n} \sum_{i=1}^n (T_{sim,p}(i) - T_{obs,p}(i))^2} \quad (5.5)$$

When calculating the *RMSE*, the reference value at index i , which here always represents an observed time series, is always subtracted from the corresponding modeled or adjusted value. In the following the value which is determined by inserting the respective time series from one of the adjusted data sets instead of $T_{sim,p}$ into Equation 5.5 is called the *RMSE* of the adjusted time series. In addition, squaring ensures that the *RMSE* only yields results greater than or equal to zero. A *RMSE* of zero means that there are no differences between the time series and that they are identical. This usually never occurs, but values close to zero are desired. The unit is always the measure in which the data between which the error is calculated. Since in this thesis, the errors between time series of temperature data are used and these are present in the unit deg. Celsius, this is also the unit of the *RMSE*.

This procedure was also applied to each cell of the data sets presented here. Since the goal of the adjustment procedures is, among other things, to minimize the mean temperature differences between $T_{sim,p}$ and $T_{obs,p}$, the *RMSE* applied to the adjusted time series should have lower values.

The root mean square error results applied to the full 10950 days in each data set are shown in Table 5.6. Again, the bias-adjusted results give better results than the unadjusted data. The *RMSE* of the bias-adjusted data sets is always lower, though sometimes just barely, than between the observed values and the raw model output.

Method	Oslo	Bremerhaven	Zurich	Milan	Lyon	Europe
No Correction	5.7899	4.4331	5.2652	4.5349	5.1652	4.4775
DM	5.0387	3.5868	5.0884	3.7486	4.8362	4.0144
LS	5.5607	4.3442	5.2626	4.2224	5.0159	4.1473
VS	5.1653	3.7602	5.1482	3.8262	4.9237	4.0526
QM	5.3905	4.1039	5.2137	3.9995	5.0723	4.1189
QDM	5.3838	4.1261	5.2101	4.0278	5.0537	4.1247

Table 5.6.: *RMSE* between observed, modeled and bias-adjusted daily mean 2m air temperatures in deg. Celsius in Europe (1981-2010)

It is striking that the application of the linear scaling method resulted in the lowest

improvement of the *RMSE*. The Europe column represents the unweighted mean of all cells in the respective data sets, i.e. the mean of the *RMSE*, which are shown in Figure 5.9 later. Thus, the bias adjustments have been able to reduce the root mean square error by almost half a degree on average.

This improvement is not as big compared to the effects on the *MBE*. This is because the methods mainly try to reduce the deviations in the mean, whereby the adjustment of particularly high deviations between individual days of two time series are also only shifted or scaled by one mean value in the scaling-based methods.

(Tab. 5.7) The consideration of the minimal and maximal values of the *RMSE* must not be omitted, because this shows that the procedures not only reduced the mean, but also the minimal and maximal value of the respective *RMSE*.

	No Cor- rection	DM	LS	VS	QM	QDM
Minimum	0.9408	0.8439	0.9345	0.7989	0.9095	0.9027
Mean	4.4775	4.0144	4.1473	4.0526	4.1189	4.1247
Maximum	10.0985	9.7370	9.2791	9.9495	9.5632	9.5497

Table 5.7.: Minimum, mean and maximum *RMSE* of daily mean 2m air temperatures in deg. Celsius in Europe between observed, modeled and bias-adjusted data sets (1981-2010)

As shown in the investigation of the mean bias error, the deviations vary strongly when the months are considered separately. (Tab. 5.8) The root mean square error is lowest in the summer months, as is the mean bias error. This trend exists in the raw model output as well as in the bias-adjusted data sets. In addition, the procedures have also provided a reduction in the mean *RMSE* in each month.

Month	No Correction	DM	LS	VS	QM	QDM
Jan	5.6292	5.3975	5.3522	5.3173	5.3896	5.3926
Feb	5.5929	5.2358	5.3333	5.3286	5.2984	5.3062
Mar	5.1347	4.6726	4.8115	4.7661	4.7774	4.7866
Apr	4.6180	3.9216	4.2694	4.0751	4.2539	4.2607
May	3.9217	3.3390	3.5568	3.3806	3.5241	3.5247
Jun	3.5838	2.9580	3.1486	3.0194	3.1090	3.1131
Jul	3.3868	2.7406	2.8641	2.7250	2.7852	2.7961
Aug	3.3092	2.7806	2.8629	2.7049	2.7553	2.7596
Sept	3.4400	2.9240	3.1280	2.9506	3.0365	3.0415
Oct	3.8842	3.4766	3.6951	3.5046	3.6508	3.6560
Nov	4.5311	4.1306	4.2907	4.2145	4.2977	4.3023
Dec	5.0957	4.8345	4.8370	4.8685	4.8514	4.8539

Table 5.8.: Mean *RMSE* per month based on daily mean 2m air temperatures in Europe in deg. Celsius (1981-2010)

In contrast to the mean bias error, the differences between the scaling-based techniques

(DM, LS, and VS) are clearly visible, but the distribution-based procedures, QM, and QDM now have very similar values.

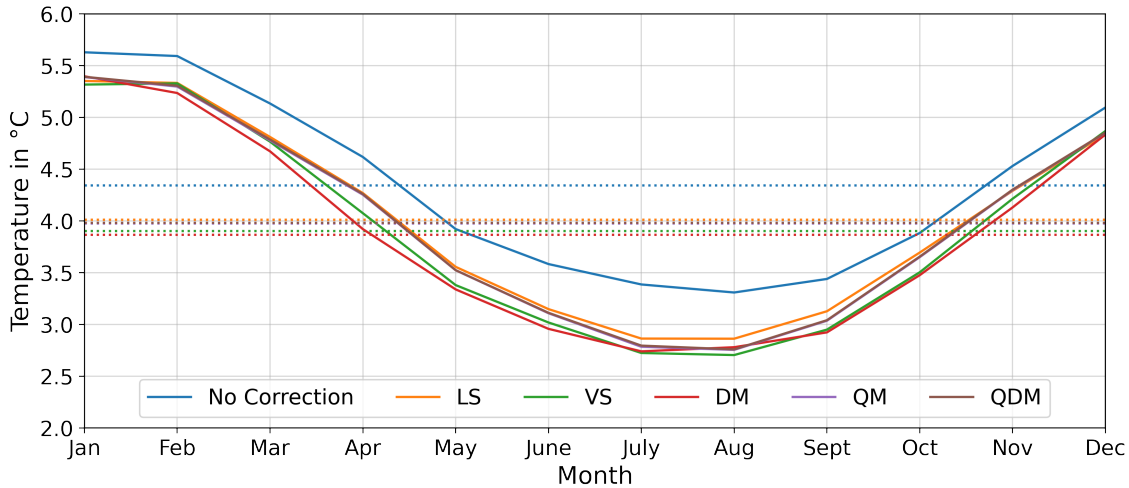


Figure 5.8.: Mean $RMSE$ per month based on daily mean 2m air temperatures in Europe (1981-2010); dotted lines indicate the mean

Figure 5.8 serves as an illustration of the values of Table 5.8 and shows how similar the values of the root mean square error of the different matched data sets are. The reduction in error compared to the unadjusted values is also evident here. Especially in summer, the methods presented in Chapter 4 could provide minimization of the squared deviations. This means that the adjusted time series have smaller temperature differences from the observed temperatures in summer than in winter. This has already been the case in the unadjusted data, but these differences could be reduced, even if in the winter months there are still sometimes particularly high-temperature differences, which result in a high $RMSE$.

This means at the same time that the MPI-ESM1-2-HR can either represent warm 2m air temperatures better than the colder ones or that in the winter months the model follows different or shifted trends, which did not occur in the observed time series. The delta method has the best values here from February to June, but these also hardly differ from those of the variance scaling method and the other methods.

Once the values of $RMSE$ between observed and modeled, or adjusted, time series have been calculated for each grid cell of the data presented in Chapter 3, this can also be presented as a contour plot. In this case, there is no consideration between land and sea regions separately, since only positive values are produced by squaring the errors, and their range of values maps well to land and sea masses.

Figure 5.9 shows the $RMSE$ of each grid cell, calculated based on daily mean 2m air temperatures over the period from 1981 to 2010 between observed and modeled or bias-adjusted time series. Contrary to expectations, the $RMSE$ is significantly lower in the North Atlantic than on the landmass of the mapped region, although the highest differences for the mean bias error can be found in the North Atlantic (cf. Fig. 5.4).

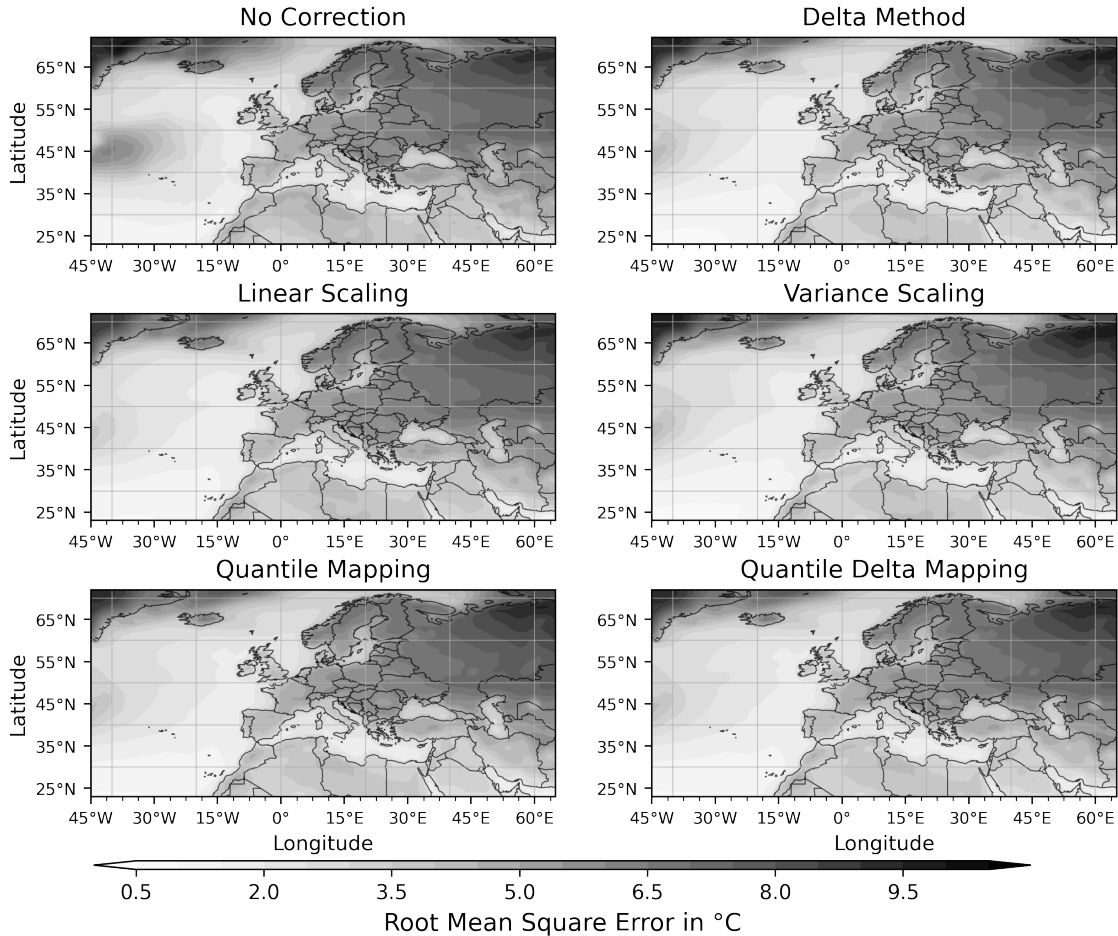


Figure 5.9.: *RMSE* between observed, modeled and adjusted daily mean 2m air temperatures in Europe (1981-2010)

It can be seen that in the North Atlantic the unadjusted data show a *RMSE* of 2.5°C to 4°C, which was significantly reduced in the bias-adjusted time series. The highest differences can be seen in the northwest of Russia and eastern Europe, although the *MBE* there shows quite small deviations in the adjusted time series (cf. Fig. 5.5). This means that the temperatures are on average similar to those of the observed time series, but individual deviations are very high. These strong deviations only come to light through squaring during the calculation of the *RMSE*. Overall, there is hardly any change in the root mean square error in this region. High values can also be seen in Greenland, but since the *MBE* has already shown high differences there, this was to be expected.

To find out exactly in which regions the root mean square error has improved or worsened, the corresponding matrix of *RMSE* values of the adjusted time series is subtracted from the matrix of unadjusted data, which has the *RMSE* values of each cell (cf. Fig. 5.9 upper left “No Correction”). Thus, it can be determined exactly where the procedures have caused a change in the values compared to the raw model output.

(Fig. 5.10) As noted previously, differences in the North Atlantic have reduced by nearly 4°C in the region from 35°N to 49°N, 45°W to 25°W. Reductions are also evident above Iceland and Russia. This has also been shown in the mean bias error. Off the coast of

Greenland, however, minimally higher discrepancies have now appeared. Since the mean bias error even showed a sign change of the mean differences there, there seems to have been a change in the trends between the control and the adjusted period. As already mentioned for the *MBE*, this may be due to climate change and the changing temperatures in this region.

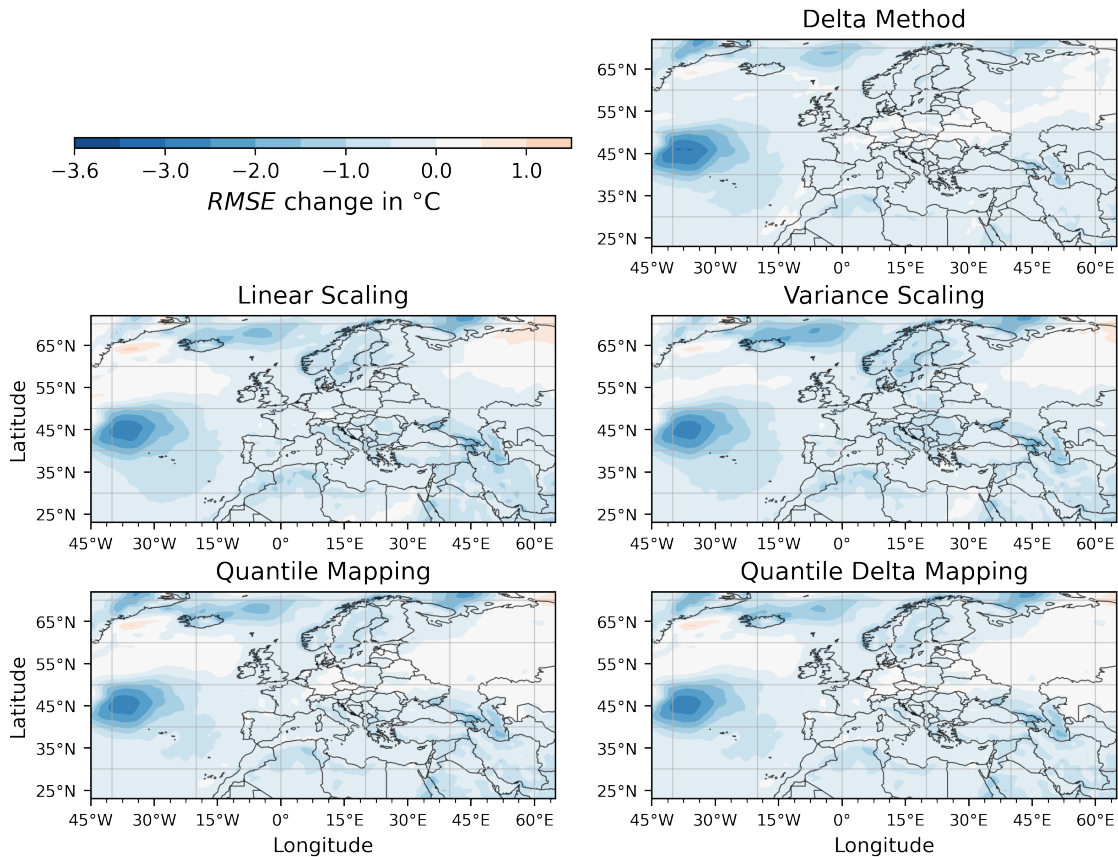


Figure 5.10.: Change of *RMSE* between bias-adjusted and modeled data based on daily mean 2m air temperatures in Europe (1981-2010)

Figure 5.9 has suggested that the *RMSE* has hardly changed in Russia. Now it turns out that it has become minimally worse in the linear and variance scaling methods, although the mean bias error has shown relatively good values there. The delta method, as well as the quantile and quantile delta mapping, have also caused a minimal deterioration of the *RMSE* in this region.

There are also many regions where the root mean square error has not improved at all or only minimally. All regions shown in white and light blue in Figure 5.10, for example, show that the differences have hardly changed when particularly high deviations are taken into account, although the mean bias error shows considerable improvements after applying the bias adjustment procedures, especially in the north of Africa, but also in the Scandinavian countries, southeastern Europe and south of Russia.

This shows that even if the mean bias error can be reduced by the procedures, the particularly high differences between observed and bias-adjusted or modeled time series cannot

always be reduced. Overall, a clear improvement of the *RMSE* in the bias-adjusted data sets can be seen, since these techniques have significantly reduced the bias in a large part of the regions.

5.1.3. Variance and Standard Deviation

Among the presented, implemented, and applied methods there is also a method, which already defines the goal, to minimize bias in variance between the modeled predicted data and the observed time series, in its name, the investigation of the effect of the methods on the variance, as well as the standard deviation must not be missing.

The standard deviation provides information about the average deviation from the mean and is therefore not applied to two time series, as was previously the case with the mean bias and root mean square error, but only to one. The mean value of the entire period under consideration is subtracted from each value in a time series. In order to take extreme values and outliers into account, this is squared and summed up for all values determined in this way. The squared variance determined in this way is the standard deviation.

$$\sigma(T_{obs,p})^2 = \frac{1}{n} \sum_{i=1}^n (T_{obs,p}(i) - \mu(T_{obs,p}))^2 \quad (5.6)$$

$$\sigma(T_{obs,p}) = \sqrt{\sigma(T_{obs,p})^2} \quad (5.7)$$

The variance of climate parameters is an important key figure in climate modeling, because it can provide information about changed climatic conditions with respect to extreme and mean values when its compared to different time series. A shift of mean and extreme values, as well as a changed range can have a negative influence, for example, on the development of different animal and plant species in nature.

Again, the starting point is by looking at the change in variance at the selected regions, as well as for Europe as a whole. Now it becomes visible that the variance in the modeled data (named ‘‘MPI-ESM’’ in the following) is higher than in the data provided by NOAA (i.e. reanalysis/observational data), both at the selected locations and averaged for Europe as a whole. Here, the procedures have mostly resulted in an approximation to the observed data, although in some cases, for example in Lyon, it can be seen that this is not always the case. In Bremerhaven, the linear scaling procedure produced even higher variance.

Data set	Oslo	Bremerhaven	Zurich	Milan	Lyon	Europe
NOAA	8.1226	6.2084	7.3259	7.4992	6.8820	6.6020
MPI-ESM	9.4526	6.7231	7.6931	8.4285	6.9570	7.0263
DM	8.6472	6.2939	7.3974	7.8512	6.9015	6.7419
LS	8.9411	6.7365	7.5529	8.1257	7.0270	6.8420
VS	8.6845	6.3448	7.4695	7.8915	6.9528	6.7900
QM	8.6782	6.3081	7.4622	7.8178	6.9742	6.7716
QDM	8.6715	6.3430	7.4653	7.8916	6.9524	6.7840

Table 5.9.: Variance of observed, predicted and bias-adjusted daily mean temperatures in deg. Celsius at selected grid boxes in Europe (1981-2010)

Now it is known that the variance of the modeled data is on average too high and the presented bias adjustment techniques could reduce this somewhat. Looking at Europe on

the map, it is possible to find out in which regions the variance is higher or lower than in the observed data.

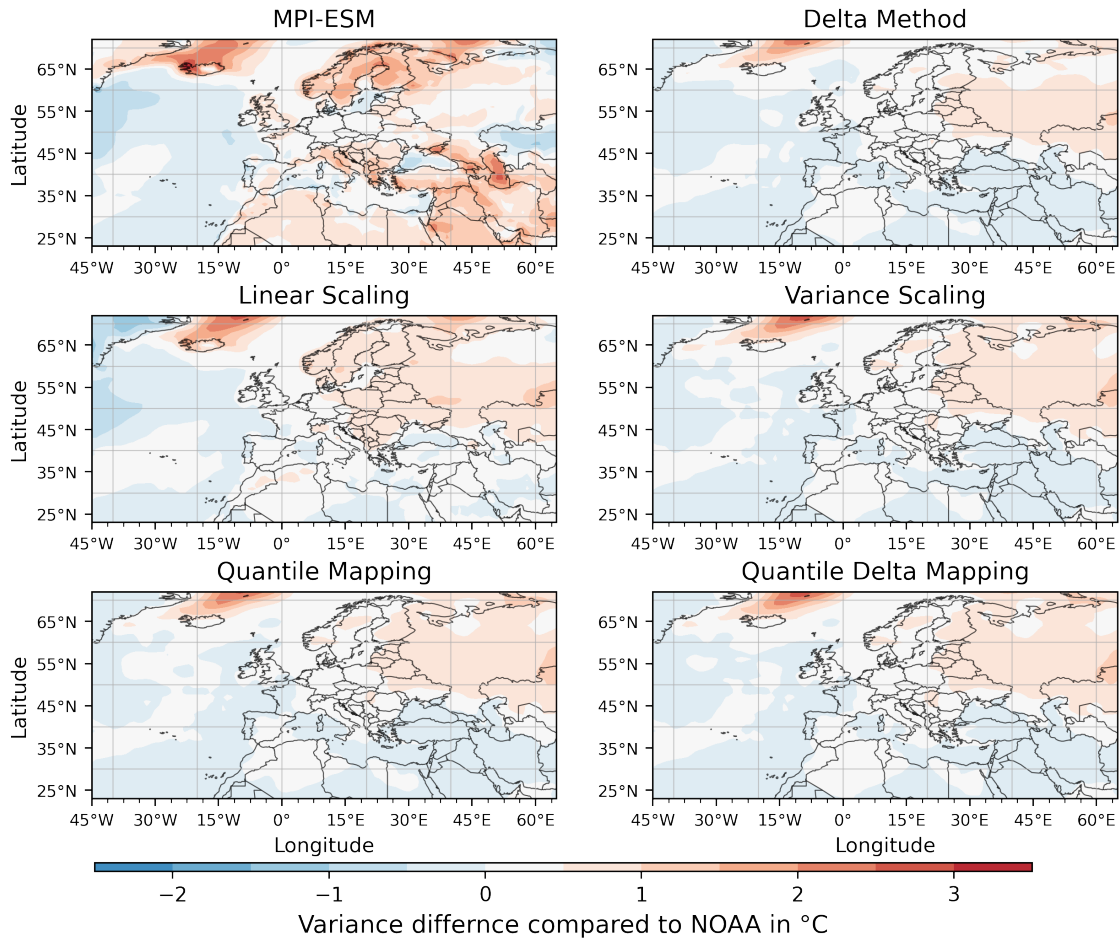


Figure 5.11.: Difference in variance between modeled and observed and bias-adjusted and observed data sets based on daily mean 2m air temperatures in Europe (1981-2010)

(Fig. 5.11) Thus, it can be seen that above Iceland, both in the unadjusted modeled data, as well as in all bias-adjusted data sets, a significantly higher variance can be found than in the observed data. The bias adjustments have been able to significantly reduce the deviation from the variance of the observed data both in the Scandinavian countries and in northern Africa. In Russia, however, the values have again worsened minimally, as it could also be noticed when examining the root mean square error. In the North Atlantic, however, there is almost no change at all, because here the variance of the modeled data, as well as in central Europe, was already quite close to the values of the observed time series.

Not only in Figure 5.11 minimal changes can be seen. Looking at the minimal, mean and maximal deviations to the observed data, it can be noticed that not only the mean deviation of the variance to the observed data has decreased by the adjustment techniques, but also the maximal and minimal values have moved somewhat towards zero:

	MPI-ESM	DM	LS	VS	QM	QDM
Minimum	-2.2077	-0.5018	-1.2392	-0.4728	-0.5497	-0.4902
Mean	0.4243	0.1400	0.2400	0.1880	0.1696	0.1821
Maximum	3.4773	2.3048	2.6475	2.7513	2.5572	2.7255

Table 5.10.: Minimum, mean and maximum deviation in variance of daily mean 2m air temperatures in deg. Celsius in Europe; observed minus modeled and bias-adjusted data sets (1981-2010)

(Tab. 5.11) Looking at the variance per month, it is surprising to see that the linear scaling procedure has the same values for the variance per month as the modeled data, although Figure 5.11 and Table 5.10 show that they do not have the same values everywhere. This is because the linear scaling procedure has changed the monthly means, for example by scaling each January by the same value. This has shifted the mean values of the months, but not directly the monthly variance, since the extreme values of the months have also been scaled by the same values. Although the delta method has a similar procedure, the variance is also influenced here, since the individual values of the observed time series from the control period have a particularly high weighting during the adjustment. In the contour plot shown, however, the variance has not been calculated for individual months, but over the entire period of 30 years (and then the matrix of the variance of the observed data has been subtracted).

Month	NOAA	MPI-ESM	DM	LS	VS	QM	QDM
Jan	3.6359	3.9108	3.7001	3.9108	3.8695	3.9846	3.9896
Feb	3.6724	3.8288	3.7825	3.8288	3.8394	3.8403	3.8489
Mar	3.2882	3.7003	3.5672	3.7003	3.6426	3.6493	3.6578
Apr	3.0227	3.4330	3.0470	3.4330	3.1694	3.3510	3.3595
May	2.6276	2.9007	2.5952	2.9007	2.6324	2.7860	2.7869
Jun	2.2910	2.4702	2.2126	2.4702	2.2846	2.3309	2.3341
Jul	1.9819	2.1440	1.9050	2.1440	1.9433	1.9894	1.9973
Aug	2.0195	2.1821	1.9157	2.1821	1.9561	2.0078	2.0117
Sept	2.2721	2.5947	2.3010	2.5947	2.3305	2.4012	2.4071
Oct	2.7159	3.0456	2.6767	3.0456	2.7535	2.9286	2.9351
Nov	3.1826	3.3748	3.1248	3.3748	3.2617	3.3357	3.3415
Dec	3.3978	3.5894	3.4563	3.5894	3.6448	3.6359	3.6403

Table 5.11.: Monthly mean variance of daily mean 2m air temperatures in deg. Celsius in Europe (1981-2010)

Once again, it can be seen that the values in the summer months are significantly lower than at the other times of the year. In addition, approximations to the observed variance have also not occurred everywhere. Thus, it can be seen that the scaling-based procedures in December, January, and February partially provided a greater distance from the variance of the observed data.

It may be noticed that the mean variance in Table 5.9 related to the whole of Europe is significantly higher than the individual values in Table 5.11. This is because in the first table the standard deviation is calculated by computing the distance from the mean, based on all temperature values for the 30 years. However, this mean may be far from the values in individual months. In the table where the variance is calculated per month, of course, the mean of the respective month is taken to determine the standard deviation and finally the variance. Therefore, a change in the variance of the linear scaled data can also be seen in Figure 5.11, but not when looking at the mean of all months.

The visual representation of the Table 5.11 is provided by Figure 5.12. It should be noted here that the variance of the modeled data is identical to those of the linearly scaled data on average (cf. Tab. 5.11), which is why the blue line is not visible. This shows that not only the variance scaling procedure is very close to the observed values, especially in the summer months, but also the values of the delta method and the distribution-based bias adjustment techniques.

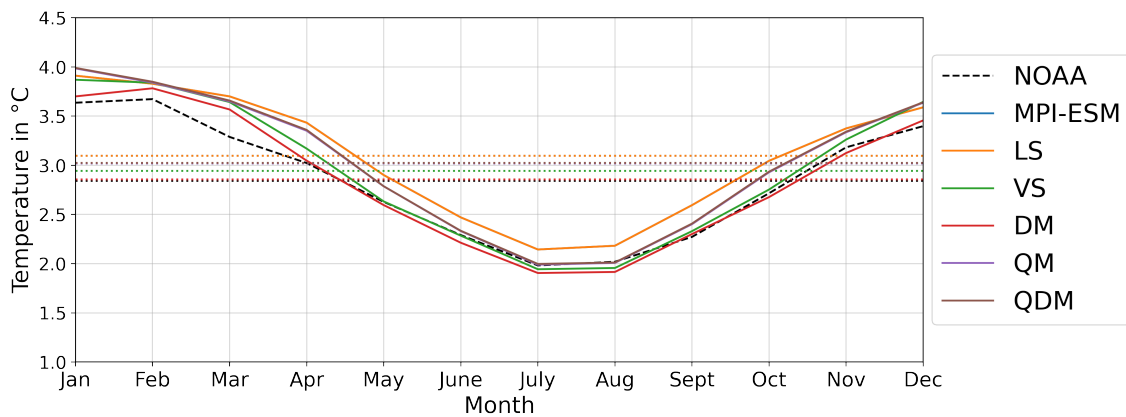


Figure 5.12.: Mean variance per month based on daily mean 2m air temperatures in Europe (1981-2010); dotted lines indicate the mean

Here, the delta method performs best, because the mean-variance is closest to that of the observed time series. Overall, it is again evident that the highest approximation to the observed values takes place in the summer months, whereby the linear scaling procedure has not caused any change on the monthly level but has had an effect on the variance of the entire observation period.

Not only the mean bias error and the root mean square error have improved significantly by applying the bias adjustments, but also the variance in the bias-adjusted data sets is now closer to the values of the observed time series. This improvement of the variance is, as in the previous studies, especially found in the summer months, but the procedures did not have such a strong influence on the North Atlantic as was the case in the mean bias and root mean square error. This is because the model there has a similar variance to that of the observed data, resulting in only minimal changes for this region. The Scandinavian countries and the regions near the Caspian Sea have benefited the most from the adjustment of the variance because there the difference to the observed time series has been reduced by up to 2°C so that there is hardly any difference anymore.

5.2. Change in Quantiles and Distribution

The mean bias error has shown that the mean values of the modeled time series adjusted with the help of the adjusting procedures are significantly closer to the observed data than the raw model output. However, mean values alone cannot be used to assess whether errors are also present at the extremes and whether these have been minimized by the procedures. Also, the investigation with the help of the root mean square error has shown that the results must not necessarily improve by the application of the adjustments. Therefore, two different approaches are now used to investigate the distributions and extremes.

In statistics, so-called quantile-quantile plots (Q-Q plots) can be used to check whether two samples or time series, have similar or even identical distributions. Here, the quantiles of the first data set are mapped against the quantiles of the second data set, so that the data points are shown, if they are identical, representing a diagonal with a 45° angle. This does not usually occur. For this reason, this reference diagonal is additionally mapped for support, to assess whether the time series have a similar distribution and how they differ from each other. Also, with Q-Q plots it is easy to see how the extreme values in the 1% and 99% quantiles behave, or if one distribution is very similar to the other, but their values are shifted slightly in one direction or the other (e.g.: If the time series have the same distribution, but one of them is shifted in the positive or negative direction).

Due to the large number of methods and the size of the data, there is an enormous range of visualization possibilities. For this reason, it was initially decided to create the Q-Q plots based on the entire time period of the respective data sets. This means that the quantiles of the observed data, as well as those of the modeled and adjusted time series, are plotted against each other over the full 30-year period between 1981-2010. It is also possible to make seasonally separate observations by, for example, plotting the quantiles of a given season against each other for each of the locations defined in Table 5.1. This will be shown later.

With the help of these Q-Q plots, it is now possible to find out how the bias adjustment techniques have affected individual quantiles. (Fig. 5.13) For the location Bremerhaven it can be seen that the predicted data of the model have a similar distribution as the observed data, but on average they have too cold values. This can be seen from the fact that the gray circles lie below the diagonally dashed line. It can also be seen here that the temperatures adjusted using the quantile delta mapping still have values that are slightly too cold, but show a high degree of similarity to those of the observed time series, especially in the middle quantiles.

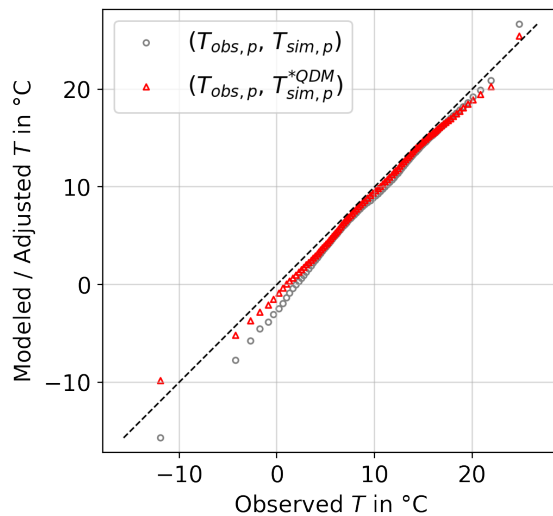


Figure 5.13.: Q-Q plot of daily mean 2m air temperatures in Bremerhaven (1981-2010)

In fact, the 1% quantile is no longer too cold after the adjustment procedure, as it was in the modeled data, but is now above the black dashed 45° reference line, suggesting that this quantile is too warm in the adjusted time series compared to the observed data.

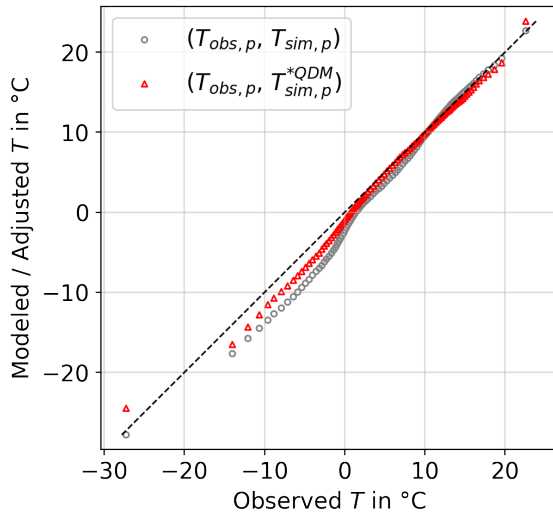


Figure 5.14.: Q-Q plot of daily mean 2m air temperatures in Oslo (1981-2010)

Similar results are seen for Oslo (Fig. 5.14) and Milan (Fig. 5.15). Again, the modeled temperatures of the 1981-2010 time series are slightly too cold on average. In both cases, the QDM-adjustment has ensured an approximation of the quantiles, although it is again noticeable that the 1% quantile in the QDM adjusted data is now depicted slightly too warm.

This points out a weakness of the bias adjustments, because if the temperatures in the modeled data of the control period may have been too cold and they are therefore mapped slightly warmer by applying the procedures, this quantile may deteriorate if the distribution of the historical observations does not also have the same relationship to the predicted time series in the future period.

(Fig. 5.16) In Zurich and Lyon, there is hardly any change in the quantiles, since the temperatures in the modeled time series already have very similar distributions to those of the observed time series. In Zurich, for example, the *MBE* has worsened minimally (cf. Tab. 5.3), this is not seen in the Q-Q plot, but the quantiles in both Lyon and Zurich have become somewhat more similar to those of the observations. Again, it can be seen that the 1% quantile has changed from too cold to too warm. The 99% quantile, on the other hand, has become closer to that of the observed time series in all Q-Q plots shown. Overall, the adjusted time series of the selected locations are still slightly too cool.

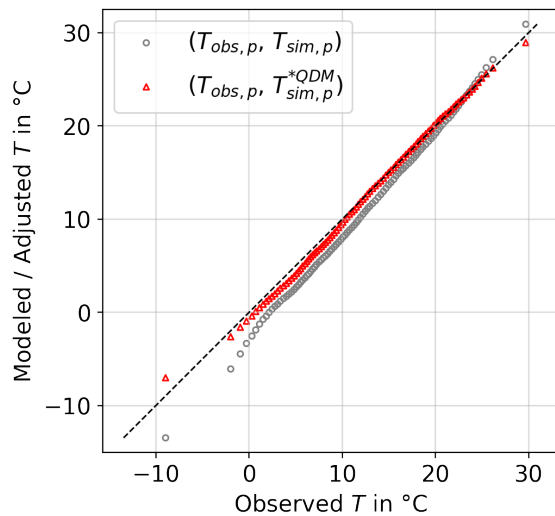


Figure 5.15.: Q-Q plot of daily mean 2m air temperatures in Milan (1981-2010)

This is shown by the fact that the quantiles of the bias-adjusted time series (red triangles) are almost always below the reference line. At the locations where the predicted data were significantly below this line, there was the greatest change, although the symmetry of the presented quantiles was mostly maintained.

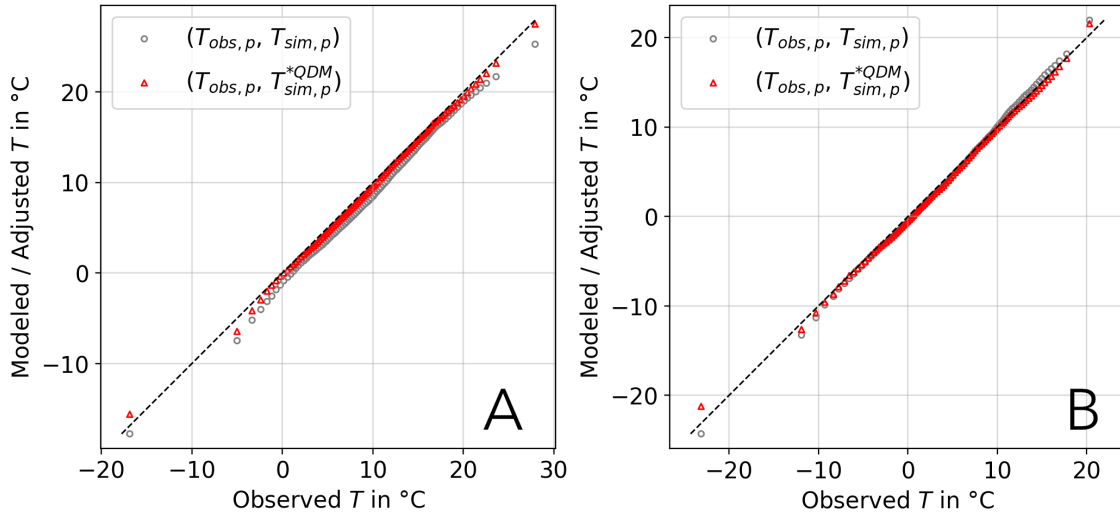


Figure 5.16.: Q-Q plot of daily mean 2m air temperatures in (A) Zurich and (B) Lyon (1981-2010)

Of course, it is also exciting to see how the quantiles behave or have changed not only over the entire time period, but how they behave in individual seasons. For reasons of space, the Q-Q plots for the winter and summer months can be found in the appendix (cf. Fig. A.5f.).

(Fig. A.5) In December, January and February, except for Zurich (D), a change in the quantiles in can always be seen, with the largest change also taking place between the 5% and 95% quantile. The quantile delta mapping has been able to bring the quantiles close to the 45° reference diagonal, in Lyon (E) even slightly above it.

Not only in winter but also in summer months, the modeled time series are colder than the observed ones in almost all quantiles. The bias adjustment techniques have been able to reduce this difference. However, the procedures did not always provide better results. Figure A.6 compares the distributions of the quantiles of the summer months June, July, and August between 1981 and 2010. In Oslo (Fig. A.6 (B)), the quantiles of the predicted time series are quite similar to those of the observed ones. When the quantile delta mapping was applied, this deteriorated significantly in the middle quantiles. This indicates that the observed and modeled time series for the control period for the Oslo site have a larger difference in the quantiles than in the 1981-2010 period, prompting the procedure to project these differences onto the predicted time series to compensate for the discrepancies. These deviations between modeled and observed values are now not as large, so the application of the adjustment technique provides for an even larger bias.

Overall, the Q-Q plots show that the 1% and 99% quantile have a particularly high distance from the remaining quantiles. The symmetry was changed minimally so that an approximation to the reference diagonal can be achieved. The quantile delta mapping almost always changed the distribution of the modeled quantiles for the better. These investigations can also be carried out for the other methods, but since this would go beyond the scope, selected figures can be seen in the appendix (cf. Fig. A.8ff.). Overall, these provided very similar results to the quantile delta mapping technique.

Apart from the investigation of the quantiles with the help of Q-Q plots, the changes in the distribution are also to be emphasized, because not only the distribution-based bias adjustment techniques have been able to influence the distributions, but also the scaling-based methods have caused changes in the mean values, as well as in all quantiles.

The Q-Q plots have shown that the quantiles of the selected time series, could be adjusted to those of the observed data. For this reason, the probability density functions are now plotted based on the whole region of Europe, to find out whether the distributions were indeed approximated to those of the observed time series not only in the selected regions but also with respect to the whole region.

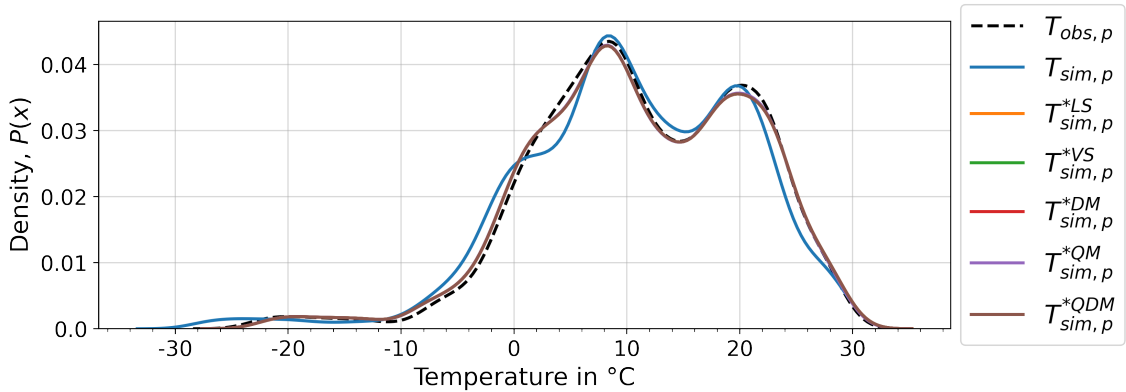


Figure 5.17.: Probability density functions of daily mean 2m air temperatures in observed, modeled and bias-adjusted data sets in Europe (1981-2010)

(Fig. 5.17) The probability density functions over the entire period from 1981 to 2010 inclusive of the observed and modeled, as well as bias-adjusted time series show that the modeled data do not differ that much from the observed data. Again, the modeled data have much colder values than the observed data set. This was also noticeable in the quantile-quantile plots, furthermore, it is now also visible that the adjusted time series not only seem to have identical distributions when looking at the 30 years as a whole but that they also do not lie so far into the negative range. This could also be shown in the Q-Q plots, since there the too cold 1% quantile was somewhat too warm after the application of the bias adjustment procedures compared to the observed values. So this particular behavior is not only found at the selected sites.

The distribution functions of the bias-adjusted data sets cannot be differentiated here, since their values are much too close to each other, but it can be clearly seen that they show higher similarity to the observed distribution function than the time series of the unadjusted modeled data set. This similarity can be observed especially in the warm value range between 23 - 30°C where the distributions are almost identical.

Since the distribution of temperatures for Europe over the entire adjustment period have also shown that the adjustment procedures have brought about a significant improvement, it is now necessary to investigate whether these also show such a high degree of similarity seasonally. For this purpose, the distribution functions have again been calculated based on daily mean values for all data sets, but these are now listed seasonally separately.

(Fig. 5.18) Also when looking at the distribution functions seasonally separated, it can be seen that the modeled data in any case have a similarity to the observed time series, but it is also noticeable that the modeled data in each season have much colder values than it is the case in the observed time series. These too cold temperatures have not only been changed in the winter months, but also at all other times of the year by the adjustment procedures, so that they are now even more similar to the distribution functions of the observed time series.

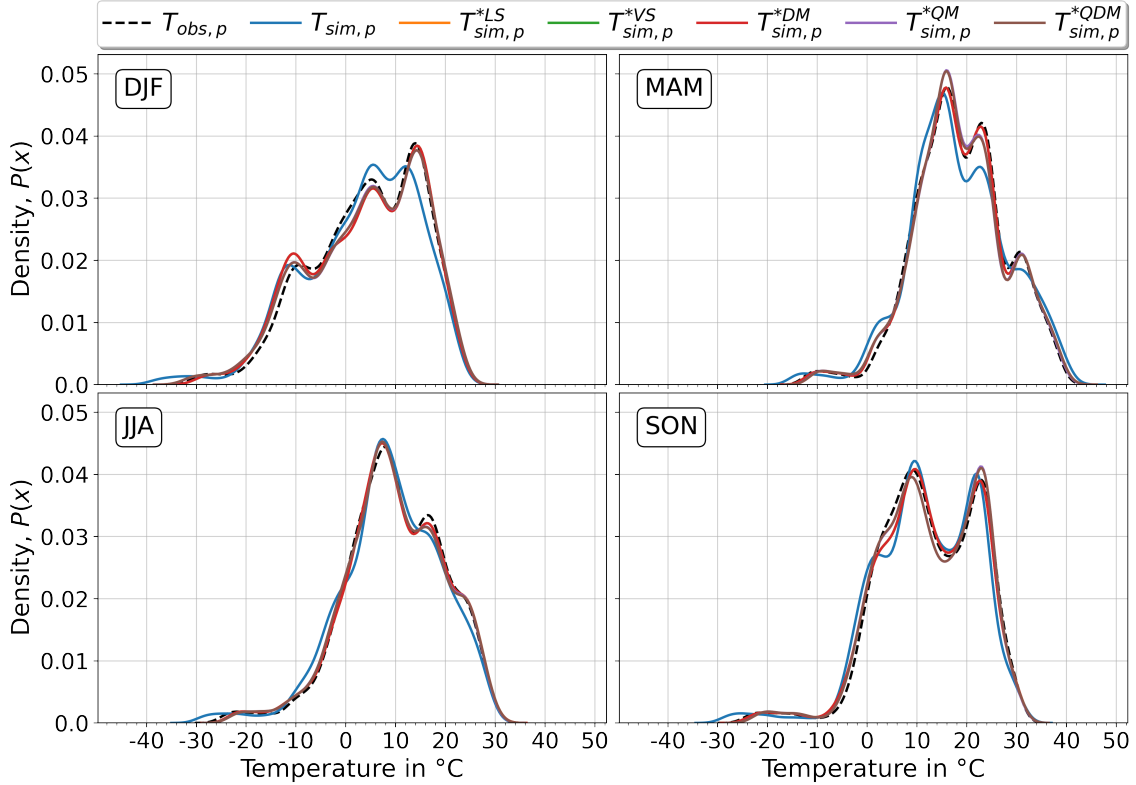


Figure 5.18.: Seasonal probability density functions of daily mean 2m air temperatures in observed, modeled and bias-adjusted data sets in Europe (1981-2010)

Furthermore, it is again noticeable here that the methods achieved very similar results overall. The scaling-based methods cannot be differentiated here, because the delta method, the linear as well as the variance scaling method are so similar to each other that they cannot be distinguished in this presentation. The distribution-based bias adjustment techniques have also produced extremely similar distributions, so that although a distinction from the scaling-based methods can be seen here, but the distribution functions of the QM and QDM-adjusted time series can hardly be distinguished from each other.

Not only the particularly low quantiles, but also in the medium and high value range all methods were able to produce an approximation to the distribution of the observed time series, so that here one can clearly speak of a minimization of the bias in the distributions. In the summer months, in which the modeled data not only have the smallest deviations from the observed data in *MBE*, *RMSE* and variance, the probability density functions are also most similar, so that here the bias-adjusted time series have even produced almost identical distributions to the observed data.

Both the quantile-quantile and distribution plots have shown that the application of the bias-adjustment techniques has always caused changes. For example, the mean quantiles of the adjusted time series in the Q-Q plots show significantly higher similarities with the observed time series than the raw data of the model. It should also be emphasized that not only the particularly cold values of the selected regions in the Q-Q plots often become too warm, but this behavior could also be observed in the distributions of the temperatures over Europe as a whole, as well as in the seasonal observation. This is a clear sign that in the control period the difference between the modeled and observed quantiles in the low-value range was higher than is the case in the period from 1981 to 2010, which now produces somewhat too warm values in these quantiles.

In the Q-Q plots, an approximation to the reference diagonal can almost always be demonstrated when observing the individual sites. This approximation to the observed data also emerged from the distributions shown, so that it is possible to speak here of a clear improvement in the distribution properties caused by the applied bias adjustment techniques, even if it is remarkable that the procedures achieved such very similar results.

5.3. Correlation and minimal Correlation Maps

The final part of the analysis of the results deals with correlations. Correlation allows making statements about the dependencies and similarities of two time series. This statistical method can also be used to determine the relationship of two time series based on the relative movements of their values. Thus a value can be generated, which states whether these time series behave similarly or even opposite to each other. This value is called the correlation coefficient, also known as the Pearson correlation coefficient. The correlation coefficient can take a value from the rational numbers, in the range of -1 to 1, where a value of -1 means that the time series have perfectly opposite movements. A value of 1 means that the time series perform exactly the same movements. Here, the values of the time series do not have to be the same or identical (like it was with the *MBE* and *RMSE*), but it is about the percentage intensity of each movement. Therefore, a value of 0 means that the time series or samples have nothing to do with each other.

$$r(x, y) = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}}, \quad r \in \{-1, 1\} \quad (5.8)$$

where:

x = First time series as 1D sequence

y = Second time series as 1D sequence

To calculate the correlation coefficient two time series are needed. These can be from the same data set, for example, if the correlation between two different locations within the observed data is to be determined, or it is to be found out how the correlation behaves between a time series from the modeled data and a time series of the same grid box from the observed data. In addition, the correlation can also be formed for specific periods, for example by using the entire observation period of 30 years for the calculation, or whether the correlation is formed separately for each year to find out how they have evolved and behaved over a longer period of time. It is important that x and y , as well as the parameters for the calculation of the *MBE* and *RMSE*, have the same length.

Similar to the first part of the evaluation, the starting point is by looking at how the correlations between the predicted and adjusted data relate to the observed time series. This provides an initial overview of whether and how the bias adjustment procedures have had an impact on the general correlations. In addition, it can be used to determine the degree of similarity between the model-generated data and the observed time series.

The correlation coefficients are determined based on the entire time frame of 30 years, i.e. 10950 daily values between observed and modeled or bias-adjusted data sets are shown in Table 5.12. It is noticeable that these correlation coefficients have hardly changed. At all selected locations, as well as averaged over the entire region, the correlations between the modeled and observed as well between the bias-adjusted and observed time series have changed only minimally.

Method	Oslo	Bremerhaven	Zurich	Milan	Lyon	Europe
No Correction	0.8076	0.7840	0.7558	0.8627	0.7381	0.7805
DM	0.8250	0.8416	0.7620	0.8824	0.7553	0.7925
LS	0.7954	0.7834	0.7511	0.8576	0.7416	0.7823
VS	0.8169	0.8268	0.7589	0.8783	0.7482	0.7872
QM	0.8001	0.7910	0.7525	0.8653	0.7335	0.7785
QDM	0.8002	0.7902	0.7529	0.8649	0.7347	0.7784

Table 5.12.: Correlation coefficients based on daily mean 2m air temperatures at different locations between observed, modeled and bias-adjusted data sets (1981-2010)

So do the bias adjustments affect the correlations of the data only minimally? For this, it is worthwhile to also look at the minimum and maximum correlation coefficients that occurred. These were determined in the same way as for the *MBE* and *RMSE*, by determining the correlation over 30 years to the observed data for all 5550 grid cells. The values in the table below correspond to the unweighted minimum, mean and maximum:

	No Cor- rection	DM	LS	VS	QM	QDM
Minimum	0.3230	0.4498	0.3757	0.4246	0.3390	0.3397
Mean	0.7805	0.7925	0.7823	0.7872	0.7785	0.7784
Maximum	0.9565	0.9592	0.9531	0.9611	0.9578	0.9578

Table 5.13.: Minimum, mean and maximum correlation coefficient of daily mean 2m air temperatures between observed, modeled and bias-adjusted data in Europe (1981-2010)

In Table 5.13 it is shown that the minimum correlation coefficients have increased after applying the procedures. This means that individual sites, which in the modeled data had only slightly similar relative movements as in the observed data, are now in the bias-adjusted time series somewhat more similar to the movements in the observed data. However, since this does not provide any information about the times at which the correlations were particularly high or low, it is worth taking a closer look at individual selected locations in the following.

5.3.1. Change in Correlation between different Locations

Since the correlations between observed and modeled, as well as bias-adjusted time series based on daily values, have not changed much, it is now appropriate to look at the correlations within a data set, because this allows to determine, for example, whether the correlations between Oslo and Zurich in the bias-adjusted data are still the same as they were previously in the modeled data, or whether they have diverged from the modeled values. If this is the case, it is important to find out how much this change in correlation is and whether these values have moved closer to or away from those of the observed data.

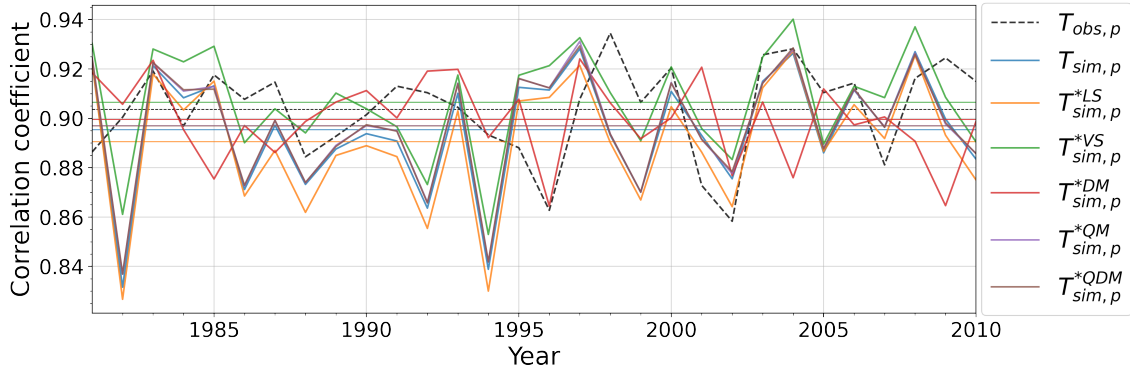


Figure 5.19.: Correlation coefficient per year based on daily mean 2m air temperatures in Zurich and Bremerhaven (1981-2010); horizontal lines represent the mean

Figure 5.19 shows the correlation coefficients per year between Zurich and Bremerhaven in the years 1981 to 2010 inclusive. Here, in the years 1981 to 1983, a high similarity between the values of the quantile mapping and linear scaling procedures, as well as the predicted data, can be seen. After that, similar but not as much excursive movements can be obtained. The data adjusted by the linear scaling technique has the lowest values overall here, but its movement is similar to those of the predicted, as well as the QM-adjusted values. The variance scaling method shows the highest correlation coefficients. The correlation of temperatures adjusted by the delta method seem to follow a different pattern than the values of the other two scaling-based methods.

All methods clearly changed the correlations between Bremerhaven and Zurich, but the mean deviations are minimal and it is difficult to say whether the variance scaling method or the delta method produced the better results. Because the mean of the VS-adjusted time series is highest, but the movements of the DM-adjusted data seems to represent the intensities of the movements better. The linear scaling method performs worst, as shown in Figure 5.19, and is the only method that shows the correlations even worse than the predicted time series.

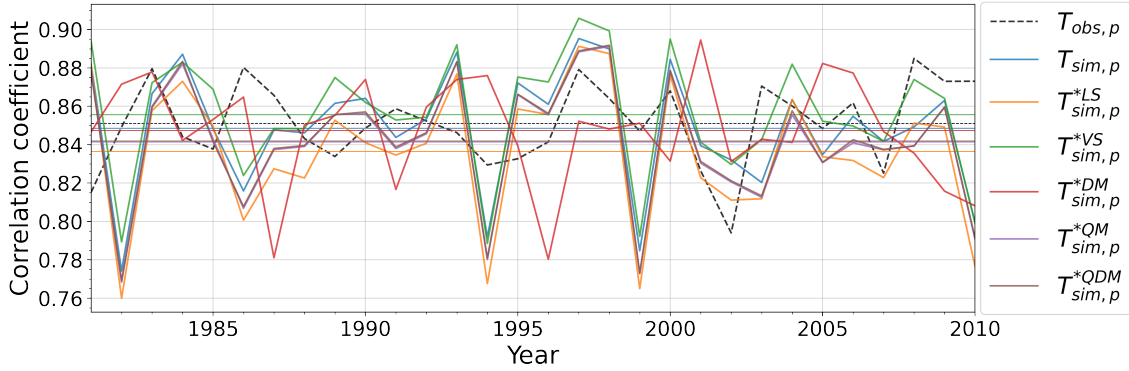


Figure 5.20.: Correlation coefficient per year; based on daily mean 2m air temperatures in Zurich and Oslo (1981-2010); horizontal lines represent the mean

(Fig. 5.20) Again, it can be seen that the methods presented in Chapter 4, have had an impact on the correlations. The linear and variance scaling methods have very similar behavior here as the modeled unadjusted time series. The distribution-based bias adjustment techniques, quantile and quantile delta mapping again show an almost identical behavior and the delta method seems to follow a different structure, however, it can be seen here as well that the methods show very similar values on average, even if they have a higher range and partly significantly lower values, than the correlation coefficients of the observed time series.

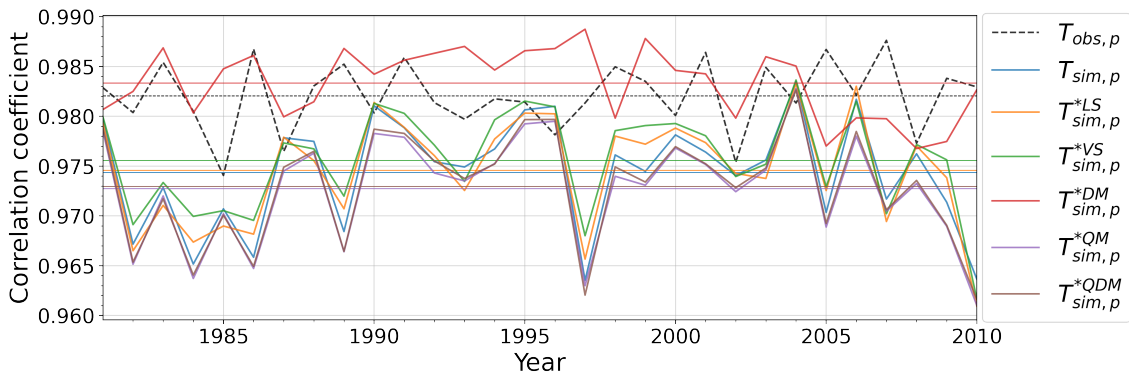


Figure 5.21.: Correlation coefficient per year; based on daily mean 2m air temperatures in Zurich and Lyon (1981-2010); horizontal lines represent the mean

(Fig. 5.21) In the correlations between Lyon and Zurich it can be seen particularly well how strongly the values differ between the methods. Except for the delta method, all other techniques show a very similar symmetry. The mean value of the delta method is much closer to the observed data than the other methods. However, since Zurich and Lyon are very strongly positively correlated, the deviations are minimal compared to Figure 5.20. The correlation coefficient of the modeled time series drops relatively sharply in 1997. This can always be seen with the exception of the DM-adjusted data in all bias-adjusted data sets.

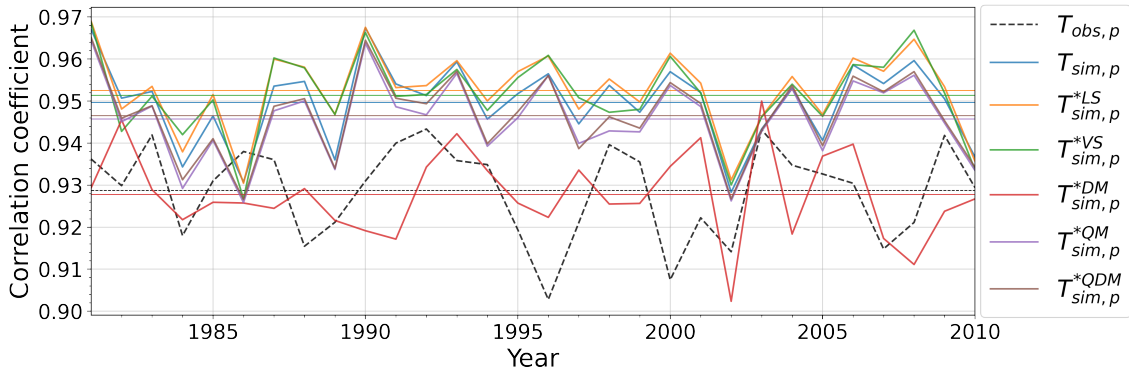


Figure 5.22.: Correlation coefficient per year; based on daily mean 2m air temperatures in Milan and Lyon (1981-2010); horizontal lines represent the mean

(Fig. 5.22) Milan and Lyon also correlate very strongly with each other, the symmetry of the methods is again very similar, and the delta method again stands out for its different behavior. Again, the mean correlations of the time series adjusted using the delta method for these sites are much closer to those of the observed correlations coefficients than the rest of the methods.

To investigate the influence of the bias adjustments outside of central Europe, the correlations between Bremerhaven and Iceland were determined (cf. Fig. 5.23). Here the correlation coefficients are not as strongly positive as seen in the previous figures, but it can also be seen that the procedures have also affected the correlations. In this case, the mean of the correlations of the predicted time series is significantly away from that of the observed time series. The procedures were able to reduce this deviation. It is interesting that the adjustments again show a similar symmetry, but this time also the delta method does not behave so differently compared to the other methods. Possibly this is since the values are not so strongly positively correlated this time.

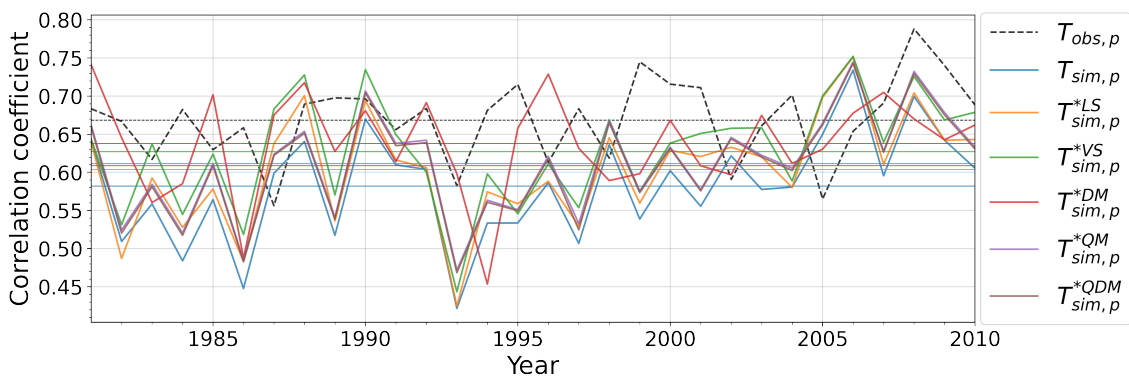


Figure 5.23.: Correlation coefficient per year; based on daily mean 2m air temperatures in Bremerhaven and Iceland (1981-2010); horizontal lines represent the mean

Except for the delta method, the correlations between the presented sites are quite similar to those of the modeled unadjusted time series. This will be because the linear and variance scaling method projects the long-term monthly means and resp. variance onto the predicted values. The delta method, on the other hand, adds the difference between

the long-term monthly means of the modeled data from the control and scenario period to the historical observations, whereby, as addressed in Section 4.1 and 4.8, the historical observed temperature value at the corresponding index is the determining factor for the final bias-adjusted value.

Of course, many more examples can be added here, because, with 5550 grid cells within one data set, a multitude of comparisons can be made. However, since it could be seen that the bias adjustment techniques had an influence on the data, another approach is taken in the last part of this analysis, which also deals with correlations, but in a slightly different way.

5.3.2. Minimal Correlation Maps

One method for determining dependencies and contrasting behavior of different regions is the creation of minimal correlation maps. These are matrices or contour plots that can be used to determine which regions and areas exhibit opposite behavior to each other. This is achieved by determining for each grid cell in a data set the correlation coefficient to every other grid cell. However, only the minimum value is of interest here.

Assume that there is a matrix A , which has the three dimensions longitude, latitude, and time. This matrix is a data set that represents, for example, the observed temperatures for the region of Europe over 30 years. To determine the minimal correlation map (i.e. minimal correlation matrix R) for this data set, for each time series $A_{i,j}$ (i.e. 1D sequence at a specific longitude and latitude) the correlation coefficient to every other time series in matrix A is determined, but only the minimum value is remembered. If $r(A_{0,0}, A_{0,1}) = -0.45$ and $r(A_{0,0}, A_{0,2}) = -0.59$ is determined afterwards, only the new, lower value is relevant. This is repeated for all further spatial dimensions until the minimum correlation coefficient of $A_{0,0}$ has been determined. Then this procedure is repeated for $A_{0,1}$, $A_{0,2}$ up to $A_{n,m}$.

The following equations describe this procedure, but the correlation coefficient of a time series to itself is also formed, which does not affect the final result:

$$A = \begin{bmatrix} T_{0,m} & T_{1,m} & \dots & T_{n,m} \\ \vdots & \vdots & \ddots & \vdots \\ T_{0,1} & T_{1,1} & \dots & T_{n,1} \\ T_{0,0} & T_{1,0} & \dots & T_{n,0} \end{bmatrix} \quad (5.9)$$

$$r_{min}(A_{0,0}) = \min(r(A_{0,0}, A_{i,j}) \text{ for } i = 0, \dots, n. \text{ for } j = 0, \dots, m.) \quad (5.10)$$

$$R_{i,j} = r_{min}(A_{i,j}) \quad (5.11)$$

where:

n = Number of longitudes

m = Number of latitudes

$A_{i,j}$ = 1D sequence of temperature values at a specific latitude and longitude

R = Minimal correlation matrix with same spatial resolution as A

Each cell of the two-dimensional matrix R contains a correlation coefficient, which indicates the minimum correlation of this cell to another cell in the same data set. This makes it possible to detect regions and areas that are slightly, negatively, or even strongly negatively correlated to each other. These relationships between regions that are strongly opposed to each other, so-called teleconnections, have already been addressed in an article by Wallace and Gutzler, 1981 by focusing on certain patterns, such as the North Atlantic Oscillation, North Pacific Oscillation and others, to study them using the sea level pressure of the northern hemisphere during the winter months.

In this article, regions that are strongly negatively correlated to each other have been marked with the help of lines to highlight the dependencies of these regions. This is also shown in the following, but only where the minimum correlation coefficients are significantly below zero. As in other contour plots, same color labels, map regions in a certain range of values. For example, regions whose grid cells have a minimum correlation coefficient of -0.3 or even lower may be shown very darkly. In order to distinguish these regions from the less strongly negatively correlated regions, it is necessary to perform a grouping or summary using a threshold. Only in this way it is possible to find out which non-contiguous regions have minimum correlation coefficients below or equal to the threshold and are related to each other (e.g. if the center of one of these regions correlates strongly negatively with another region and at the same time surrounding cells of this center correlate positively with the center). The determination of such a threshold depends on the range of values of R , therefore it cannot be defined in general. The determination of the threshold is justified at the appropriate place.

Thus, it can be determined which regions have negative correlations and with which other regions this correlation relationship is established. Furthermore, it can be compared whether these relationships exist not only in the observed data, but also whether they are represented in a similar way in the modeled time series and if not, whether the bias adjustment techniques have had an influence here.

The use of correlation matrices, which were determined based on strong positive correlations, does not make sense, because almost all cells are strongly positively correlated with their surrounding neighboring cells so no usable results can be obtained.

Since the data sets used here cover a period of 30 years, there is sufficient statistical significance, since the variability is no longer so strongly influenced by individual values due to the large time span.

In the following, the minimal correlation maps are determined and shown based on monthly means, daily means for the winter months, and for the summer months of the period from 1981 to 2010 inclusive. Here, the scales of the color bars are adjusted, so that these three different perspectives each have their own ranges of color spectra.

Minimal correlation based on monthly means

Since the scaling-based methods (DM, LS, VS) adjusted the modeled data set based on long-term monthly means, the first interest here is what such a minimal correlation map (i.e., minimal correlation matrix) looks like when it is determined based on monthly means. This means that for each month between 1981 and 2010, the mean is calculated so that $30 \cdot 12 = 360$ temperature values for each time series are used to determine the minimum correlation coefficient to each other grid box. The following table lists the minimum, as well as mean and maximum values of these minimal correlation matrices determined in this way:

	NOAA	MPI-ESM	DM	LS	VS	QM	QDM
Minimum	-0.0321	0.3517	0.3512	0.3323	0.3613	0.0430	0.0506
Mean	0.2102	0.5489	0.5515	0.5068	0.5533	0.2506	0.2525
Maximum	0.4135	0.7381	0.7489	0.6589	0.7339	0.4467	0.4598

Table 5.14.: Minimum-, mean- and maximum-minimum correlation based on monthly mean 2m air temperatures in Europe (1981-2010)

From Table 5.14 it is clear that the modeled, as well as the time series adjusted by the scaling-based methods, have significantly higher correlations overall. Not only the minimum correlation of, for example, 0.3512 in the delta method is significantly higher than in the observed time series, but also the maximum and mean values are significantly higher. The quantile and quantile mapping perform best here and have roughly similar values to that of the observations. This is due, to the fact that these two methods were not applied to time series grouped into months and their distribution-based adjustment approach.

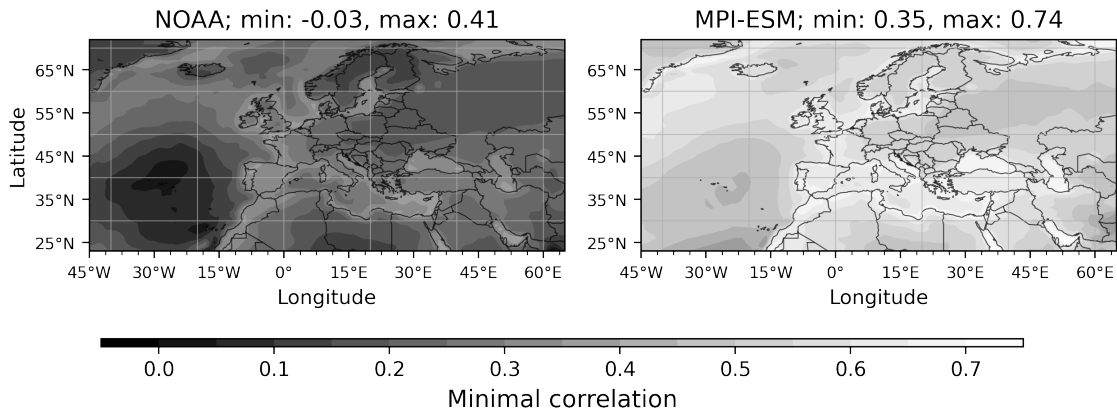


Figure 5.24.: Minimal correlation maps of the observed (NOAA) and modeled (MPI-ESM) data set based on monthly mean 2m air temperatures between 1981 and 2010

Figure 5.24 plots the minimum correlation maps of the observed/reanalysis data provided by NOAA and the raw model output of the 2m air temperatures from the MPI-ESM1-2-HR, respectively, for the 1981-2010 time period. Here the correlations were determined based on monthly means, as described before, by determining the correlation coefficient from each grid box to each other, but only the lowest value is saved at the corresponding position (i.e. index). This now allows revealing in which regions negative, neutral or even positive minimum correlations exist. In this example, in the observed data the minimum correlation is about -0.03, which means that some regions have no relationship to other regions, because their temperatures follow differently, but not opposite trends.

In these figures, no teleconnections are shown, because teleconnections are characterized by an opposite correlation ratio, but in none of the correlation matrices calculated on a monthly mean basis, particularly negative correlations occur. The lowest values occur in the observed time series in the North Atlantic. Since these values are close to zero, this

means that there are places to which this region is neither really positively nor negatively correlated. It can be seen that the modeled data has much higher values than the observed data set. This suggests that the monthly mean temperatures in the modeled data are much too correlated. However, the distance between the minimum- and maximum-minimum correlation coefficients is almost identical. The modeled data also have the lowest minimum correlations in the North Atlantic, but this region correlates positively with other regions, rather than not at all.

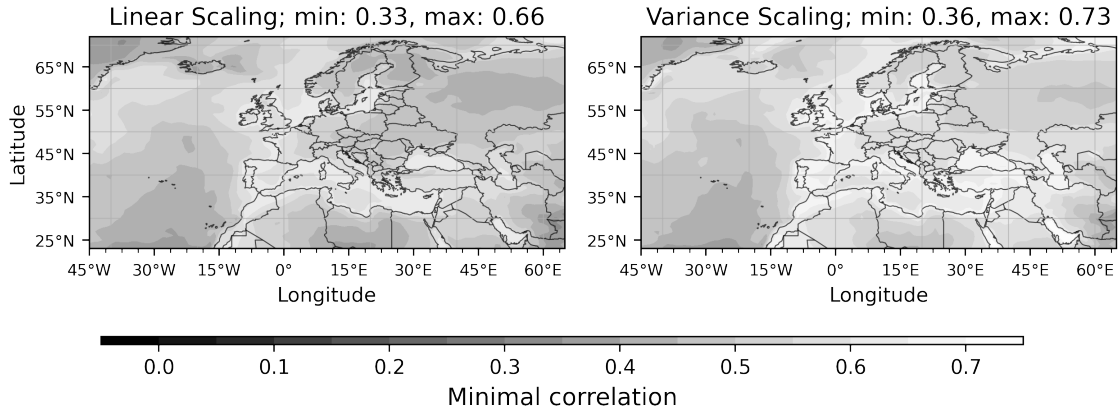


Figure 5.25.: Minimal correlation maps of LS- and VS-adjusted data set based on monthly mean 2m air temperatures between 1981 and 2010

(Fig. 5.25) The linear and variance scaling procedures also have significantly higher minimum correlation values (cf. Tab. 5.14), similar structures to the modeled data (cf. Fig. 5.24) can also be seen for both. This is not unusual because these procedures have directly affected the monthly means, for example, by subjecting each January to the same influence (cf. Section 4.2f.), which has hardly affected the correlations here.

The same applies to the time series adjusted using the delta method. Here, the monthly means were also changed, but not based on the difference in the data of the control period, but on the difference between the historical modeled data and the scenario time series. For this reason, even with the delta method, the minimum correlations are hardly distinguishable from those of the modeled data.

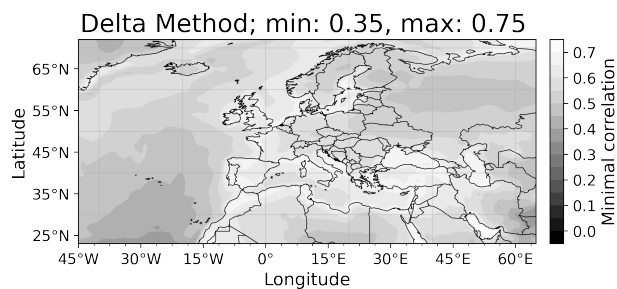


Figure 5.26.: Minimal correlation map of DM-adjusted data based on monthly mean 2m air temperatures between 1981 and 2010

The minimal correlation maps of the quantile and quantile delta mapping adjusted data sets look clearly different, because there is a very high similarity, with those of the observed data (cf. Fig. 5.24, left)). This is a massive improvement in the correlations based on monthly means compared to the unadjusted modeled data set. There is a very high similarity, not only in the Atlantic and Greenland but also in the Scandinavian countries, as well as in North Africa. Here,

the distribution-based bias adjustments have achieved significantly better results than the scaling methods.

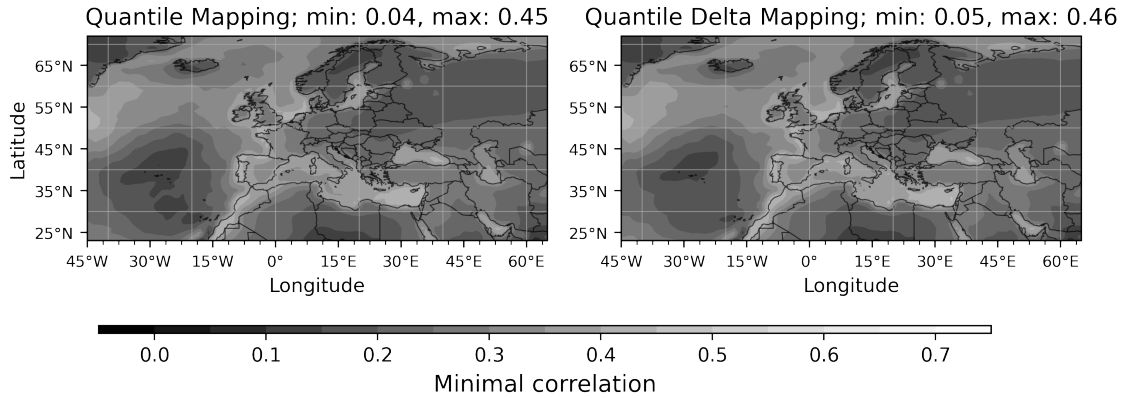


Figure 5.27.: Minimal correlation maps of QM- and QDM-adjusted data set based on monthly mean 2m air temperatures between 1981 and 2010

It has been shown that the minimum correlations based on monthly means of the modeled data are significantly higher than is the case in the observed time series. This suggests that not only are there many more regions that are positively correlated with each other but also that unrelated regions nevertheless exhibit at least rudimentary similar behavior in the modeled data. This could not be eliminated by the scaling-based bias adjustment techniques, but both the quantile mapping and the quantile delta mapping could provide an enormous improvement. With the QM- and QDM-adjusted data sets it is now possible to map these minimal correlations based on monthly means very similarly compared to the observed time series.

Minimal Correlation based on daily mean Temperatures in Winter

As in the article of Wallace and Gutzler, 1981, the minimal correlations of the winter months are now formed. However, here this is calculated based on daily mean 2m air temperatures, this means that for $(31 + 31 + 28) \cdot 30 = 2700$ values of each grid box the correlation coefficient to each other grid box is determined to find the minimum value.

(Tab. 5.15) The minimum, mean and maximum values of the modeled data are very similar to those of the observed data, therefore it is not surprising that the values of the bias-adjusted data sets also show quite similar values.

	NOAA	MPI-ESM	DM	LS	VS	QM	QDM
Minimum	-0.4111	-0.4108	-0.4112	-0.4430	-0.4090	-0.4057	-0.4064
Mean	-0.2410	-0.2507	-0.2555	-0.2678	-0.2533	-0.2484	-0.2491
Maximum	0.0114	-0.0379	-0.0510	-0.0072	-0.0591	-0.0319	-0.0343

Table 5.15.: Minimum-, mean- and maximal-minimum correlation based on daily mean 2m air temperatures in winter months in Europe (DJF, 1981-2010)

In the time series of the winter months between 1981 and 2010, significantly lower correlation coefficients can be seen overall than was previously the case with the matrices calculated on monthly means. This is mainly because seasonality plays a particularly important role when looking at minimal correlation maps based on monthly means, because this occurs at almost all locations and thus ensures, at least, a minimal correlation close to zero. These seasonal temperature changes are now no longer so strongly present since the focus is only on the winter months of December, January, and February. However, the delta method has also provided a distance in the mean from the observed data, just as it did on a monthly basis previously. The other two scaling-based methods also have worse values on average.

Since the correlation values are now sufficiently low to speak of negative correlation ratios, a threshold of -0.3 was defined. This is used to group regions together so that it can be determined to which other regions negative correlations exist. These teleconnections are marked by red lines. A cross marks the grid cell which has the most negative correlation within the respective region or to which another cell a negative correlation ratio less than or equal to the threshold exists. This makes it possible to find out which regions have an opposite behavior and at the same time to check if this can be found in the modeled and bias-adjusted data sets as well as in the observed ones.

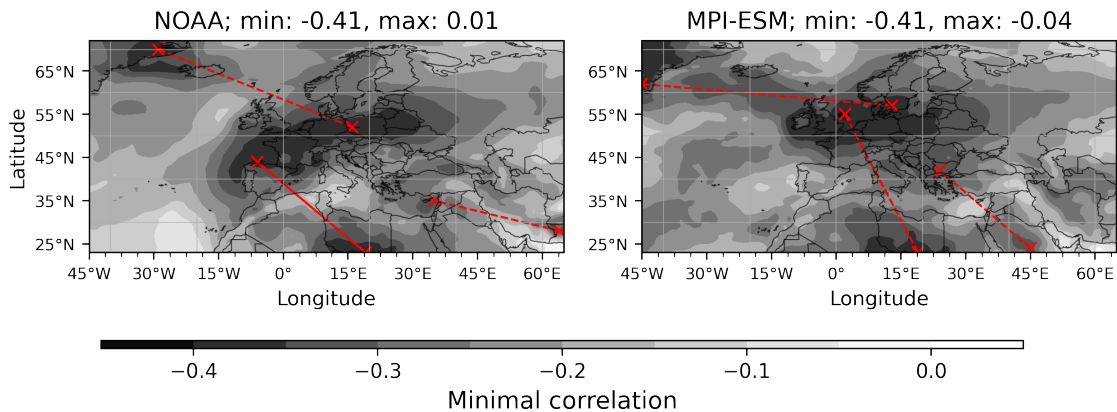


Figure 5.28.: Minimal correlation maps of the observed (NOAA) and modeled (MPI-ESM) data set based on daily mean 2m air temperatures in winter months between 1981 and 2010

It can be observed that in both the modeled (Fig. 5.28, right) and the linear and variance scaling adjusted data sets (Fig. 5.29) very similar contours can be found as in the observed data (Fig. 5.28, left). However, the patterns of the LS- and VS-adjusted data have a higher similarity to the raw modeled time series of the MPI-ESM1-2-HR, even if their teleconnections (red lines) are found in approximately the same regions. Thus, winter temperatures in Libya and Algeria are shown to correlate negatively with the Bay of Biscay in the observed data. However, in the modeled data, as well as those of the delta method, this correlation exists towards the Baltic Sea, and the LS- and VS-adjusted time series have their minimum correlation towards the African countries from Ireland. There is also an opposite behavior between the north of Europe and Greenland in all the data.

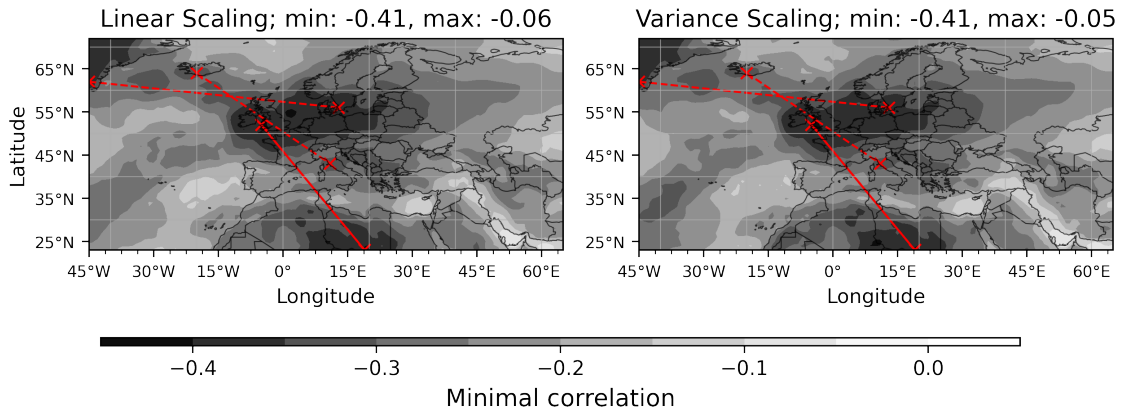


Figure 5.29.: Minimal correlation maps of LS- and VS-adjusted data set based on daily mean 2m air temperatures in winter months between 1981 and 2010

The minimum correlation matrix of the winter months within the data adjusted using the delta method differs somewhat from the time series adjusted using linear and variance scaling methods, because here the correlation coefficients in northwest Europe are somewhat lower, which is indicated by the lighter color scheme. Here also the connection between northern Europe and northern Africa can be seen, but also in the European North Sea.

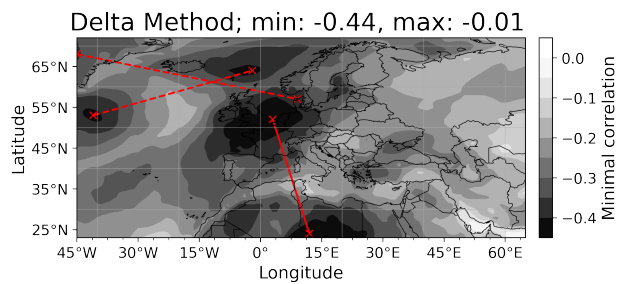


Figure 5.30.: Minimal correlation map of DM-adjusted data based on daily mean 2m air temperatures in winter months between 1981 and 2010

There is now a relatively large region, which has a negative correlation with the North Atlantic. These two negatively correlated regions are not found in such a form in either the modeled or the observed data. This possibly indicates that this regions in the observed data of the control period, which have a large influence on the adjusted values in the delta method, have a fundamentally different relationship to each other. The fact that the DM-adjusted time series has somewhat different patterns than the other methods has also been shown in the comparisons in Section 5.2.

(Fig. 5.31) The distribution-based bias adjustment techniques also produced similar results here as the modeled and observed data. The teleconnections between North Africa and northern Europe, as well as between northern Europe and Greenland can be seen. Furthermore, a connection between Tehran, near the Caspian Sea and the Baltic Sea is also present in the minimum correlation matrix of the quantile mapping, as well as in those of the quantile delta mapping adjusted time series. Not only the teleconnections but also the symmetry of the contours and their colorings are very similar here.

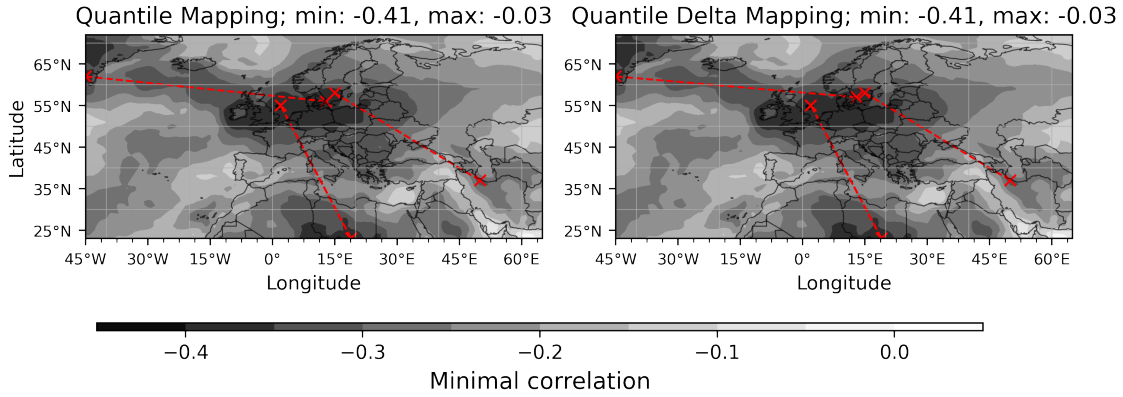


Figure 5.31.: Minimal correlation maps of QM- and QDM-adjusted data set based on daily mean 2m air temperatures in winter months between 1981 and 2010

In all minimal correlation matrices just graphically shown, teleconnections between different regions can be seen. This means that they have at least a slight negative correlation relationship due to the threshold of -0.3. Simply put, this means that when temperatures rise in one of these centers, temperatures fall at the other end of the teleconnection. Of course, this does not always have to be the case and these temperatures getting warmer and colder does not happen 1:1, but only minimally due to the low threshold, however, there is a relationship between these regions that cannot be overlooked, which do not have to influence each other but can provide information about what can take place at the other end of the connection.

Minimal Correlation based on daily mean Temperatures in Summer

Finally, the minimum correlation matrices for the summer months of June, July, and August should not be missed. As found in previous studies, both the mean bias and the root mean square error is lowest in these months. Also, the variance of 2m air temperatures is lowest in these months not only in the observed and modeled but also in the bias-adjusted time series.

For the minimal correlation matrices, $(30 + 31 + 30) \cdot 30 = 2730$ daily mean 2m air temperature values of the summer months between 1981 and 2010 were used to determine the minimum correlation from each cell to each other. The following table shows the minimum, average, and maximum correlations coefficients thus determined for the various data sets.

	NOAA	MPI-ESM	DM	LS	VS	QM	QDM
Minimum	-0.7465	-0.8374	-0.7518	-0.8441	-0.8607	-0.8011	-0.8002
Mean	-0.4008	-0.4524	-0.3935	-0.4464	-0.4526	-0.4248	-0.4247
Maximum	-0.0080	-0.0625	-0.0317	-0.0293	-0.0270	-0.0378	-0.0398

Table 5.16.: Minimum-, mean- and maximum-minimum correlation based on daily mean 2m air temperatures in summer months in Europe (JJA, 1981-2010)

What is immediately noticeable is that the summer months are characterized by particularly negative correlations, because both the mean and the maximum values of the minimum correlation matrices are always in the negative value range. Furthermore, the values of the modeled data are slightly lower than the observed ones, which means that this data set has grid cells that show a stronger opposite behavior than the actual observed time series. A threshold of -0.7 is set for the display of teleconnections.

The linear and variance scaling procedures have provided a minimal change here, but this is hardly worth mentioning. The minimum correlation coefficient in both the variance and linear scaling procedures have moved away from the minimum value in the observed time series.

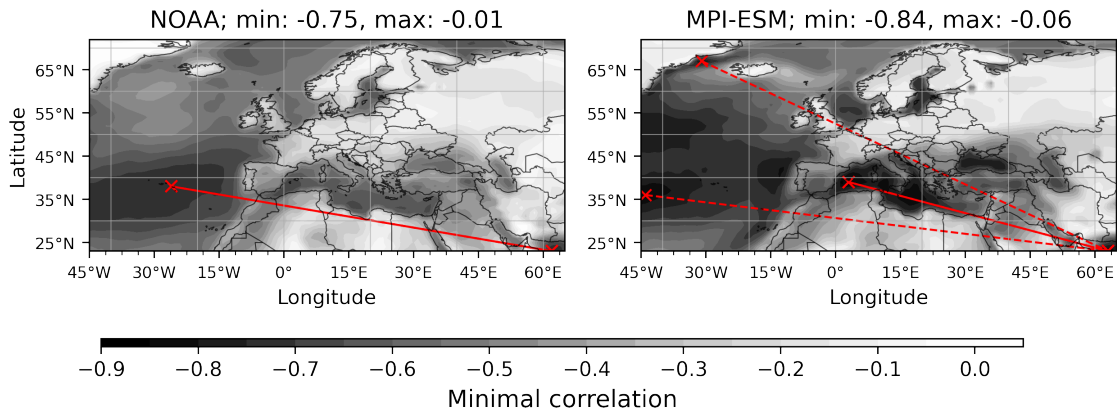


Figure 5.32.: Minimal correlation maps of the observed (NOAA) and modeled (MPI-ESM) data set based on daily mean 2m air temperatures in summer months between 1981 and 2010

The stronger negative correlations of the modeled time series can be seen on the right in Figure 5.32. There are clearly darker regions than in the observed values. Furthermore, it is now noticeable that in summer almost the entire land mass has minimum correlation coefficients close to zero. This means that there are no to few regions to which they have a strong negative trending behavior.

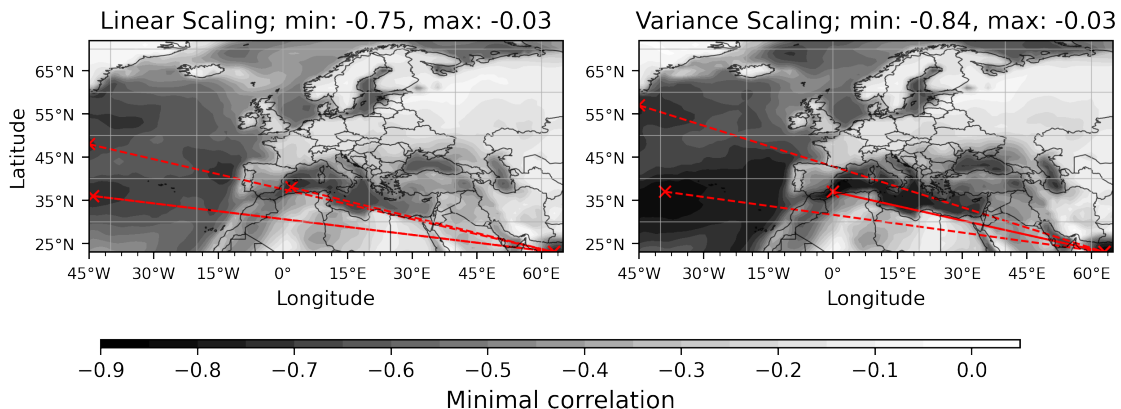


Figure 5.33.: Minimal correlation maps of LS- and VS-adjusted data sets based on daily mean 2m air temperatures in summer months between 1981 and 2010

On the sea mass, however, in the North Atlantic, both in the observed, as well as in the modeled and bias-adjusted data sets, is a region that correlates strongly negatively with the region around Pakistan. If the spatial section of the considered region were larger and not only tailored to Europe and its surroundings, one would perhaps find that this region is actually located rather near India, since at this time the monsoon prevails there for a climate that is contrary to that found in the North Atlantic during the summer season.

These strong negative correlation relationships (i.e. teleconnections) of the modeled data are also found in all bias-adjusted data sets, even if, for example, the region near the coast of Greenland has shifted somewhat. Also, a relationship between the monsoon region and the southern end of the Mediterranean Sea is always found.

(Fig. 5.34) In the DM-adjusted data set, the connection between the monsoon region and the coast of Greenland is missing. This is due to the fact that always only the three most disturbing teleconnections while respecting the threshold are mapped, otherwise this would overload the figures.

However, it is also clearly seen here that the North Pacific, as well as the Mediterranean and even part of the Caspian Sea have strong negative minimum correlations to the monsoon region. At the same time, this can also mean that the regions which have a strong negative correlation to the monsoon region have a very positive correlation ratio among themselves.

On the land regions, the nearly neutral minimum correlation coefficients can be justified by the fact that temperatures tend to become warmer in summer and, as previously noted, have a lower variance, making them particularly different from the monsoon and sea region.

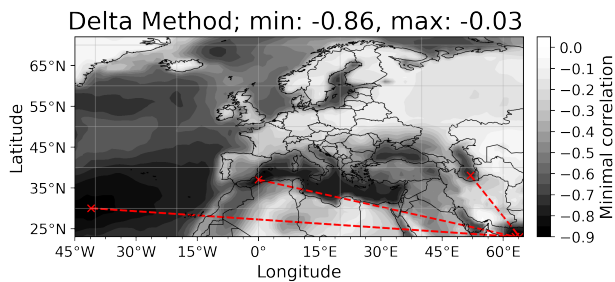


Figure 5.34.: Minimal correlation map of DM-adjusted data based on daily mean 2m air temperatures in summer months between 1981 and 2010

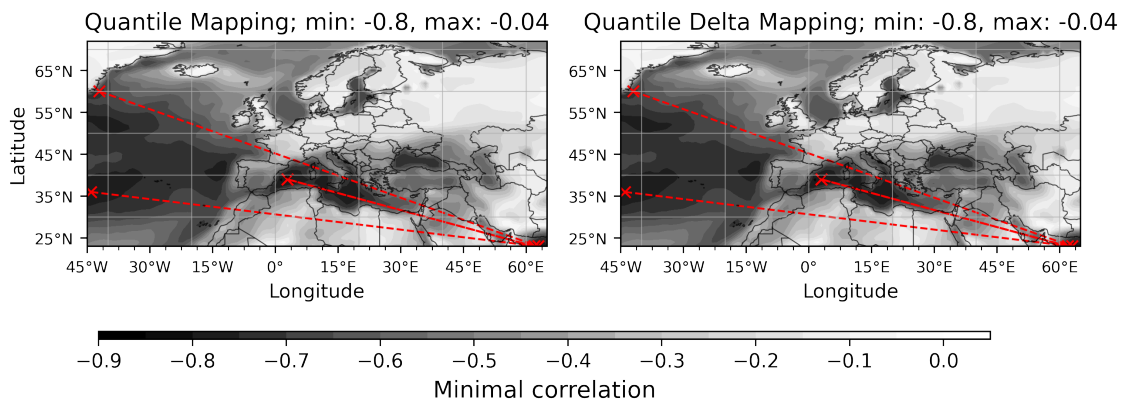


Figure 5.35.: Minimal correlation maps of QM- and QDM-adjusted data set based on daily mean 2m air temperatures in summer months between 1981 and 2010

The quantile and quantile delta mapping also provided very similar results. Here the values

are similar to those of the modeled data and are still visibly different from the observed values in the North Atlantic.

The minimum correlation matrix of the observed summer months between 1981 and 2010 have overall higher minimum correlation coefficients. This resulted in only two regions, which have a center with a correlation coefficient less than or equal to -0.7. In addition, there are not as negative values, near the coast of Greenland, as it is the case in the modeled and bias-adjusted data. Also the Baltic, Mediterranean and Red Sea do not have such strong negative correlations in the observed time series. In all minimal correlation matrices of the summer months between 1981 and 2010 inclusive, it can be seen that the land regions have minimal correlation coefficients close to zero and the regions that have strong negative correlations point towards the monsoon region of India.

The adjustment techniques have always been able to provide minimal changes, but only in the case of the distribution-based bias adjustment techniques, as well as in the case of the data of the time series adjusted with the help of the delta method, it has been noticed that they have been able to bring their values closer to those of the observed time series both in the minimum and in the maximum and the mean (cf. Tab. 5.16).

Minimal Correlation Matrices Conclusion

It was possible to prove that the bias-adjustment procedures had a clear influence on the correlations and teleconnections within the data sets. It has been shown that the minimum correlation matrices of the bias-adjusted data sets have very similar values, analogous to the modeled data, except for QM and QDM while investigating the minimal correlation matrices based on monthly means. Particularly noticeable is that when considering minimum correlation matrices based on monthly means, the modeled data have far too positive minimum correlation coefficients compared to the observed time series. This could not be remedied by the scaling-based techniques, but both quantile and quantile delta mapping could provide a very good approximation to the observed minimum correlation matrix here. This means that the scaling-based methods are much too bound to the individual months so they cannot influence the global correlations due to seasonality.

In the separate consideration of winter and summer months, the applied techniques have produced very similar results, the minimum correlation coefficients are particularly far in the negative range in the summer months, since here a strong opposite behavior exists between the Mediterranean, as well as the North Atlantic in the direction of the Near East region. In the winter months, however, other patterns can be seen, because especially northern Europe always shows an at least partly opposite behavior to North Africa, as well as the coastal area of Greenland.

6. Conclusion

In this thesis, five different methods for minimizing bias between modeled and observed time series of air temperatures in 2m height for the European region are shown and implemented both mathematically and in the programming languages Python and C++. The mathematical foundations were derived for the scaling-based bias adjustment techniques from the articles of Teutschbein and Seibert, 2012 and Beyer et al., 2020. For the quantile and quantile delta mapping, the articles of Tong et al., 2021, and Cannon et al., 2015 were followed.

All presented methods aim at minimizing deviations between observed and modeled climate time series data by determining the bias between past or present simulated and observed data and projecting it to subsequent periods. The approaches are very different. While the delta method focuses on the differences between two modeled time series, the linear and variance scaling methods try to apply the differences in mean or even variance between observed and modeled data on the control period. The distribution-based bias adjustment techniques do not scale the monthly time series, but minimize deviations, by the appropriate assignment of quantiles between the control period and the time series to be adjusted.

In this work, observed and modeled 2m air temperatures of the 1951-1980 control period were always used to minimize the bias between the time series of modeled and observed data of the 1981-2010 period. Care was taken to ensure that all data sets had historical values only to compare the influences of the procedures with existing observed data. Of course, it is also possible to adjust modeled time series of future periods using the methods presented here, but these could not be evaluated in such a comparative way as is the case with the historical data, because of course no observed data sets are available for future periods yet.

It is also possible to bias-adjust the modeled data of the control period by using only data from the control period for the adjustment. This can lead to much better results compared to time series of different periods, since all methods presented here assume that the bias between modeled and observed time series remains consistent over time. Since both the error between the modeled data (representing the same time series, because control period equals scenario period) and the error between the observed and modeled data remains the same, these deviations can be adjusted directly.

After the bias adjustment methods have been elaborated, presented, implemented, and applied, the comparative analysis between the observed, modeled, and bias-adjusted time series of the period 1981-2010 has been performed. Here it has been shown that the methods have not only changed the mean temperatures but furthermore have provided for lower mean bias and root mean square error. Even the variance of the modeled 2m air temperatures of the MPI-ESM1-2-HR could be approximated to the values of the observed time series with almost all procedures for the period 1981-2010. However, this improvement is not always the case, because it could also be shown that the mean error

can also increase after applying the procedures (cf. Tab. 5.3). It has been shown that the modeled data are on average significantly colder than the observed time series and this was already the case in the control period, which has led to the fact that all procedures have always ensured higher mean temperatures, which has significantly reduced the deviations in the summer months, both in terms of variance, as well as the mean value and the root mean square error.

Despite their different approaches, all investigated methods provided similar but not identical results and these were in almost all cases significantly better than the raw model output of the MPI-ESM1-2-HR. This has been noticed not only in the comparison of the mean bias and root mean square error of the temperatures, but also in the investigation of the distributions using quantile-quantile and probability density plots. Here it has been shown that especially the mean tercile after the adjustment shows significantly more similar values to the observed time series. However, it was possible to change the extreme values so that these could be approximated to those of the observed data for the selected time series. In addition, the distribution functions of the whole European region could be approximated to those of the observed data, so that these provided almost identical results, especially in the summer months (cf. Fig. 5.18).

The examination of the minimal correlation maps, as well as the teleconnections they contain, has shown that the grid cells of the modeled data are much more positively correlated with each other than the observed in the data set. These sometimes strong differences could not be eliminated by the scaling-based bias adjustment procedures, but the distribution-based techniques were able to provide clear approximations to the observed matrices. These overly strong seasonality-induced trends in the model are a perfect example, on the one hand, of the differences between the adjustment procedures, but also of the need for bias adjustments. Teleconnections, which characterize strongly negatively correlated grid cells and regions (Wallace and Gutzler, 1981), could be maintained by the adjustment methods for the selected summer and winter months, so that except for the delta method always very similar teleconnections were detected. Contrary to the assumption, which was part of the motivation of this work, it could not be confirmed that the minimal correlation matrices and their teleconnections deteriorate by the application of the presented bias adjustment techniques.

Thus, it can be clearly stated that the applied bias adjustment techniques have had an impact on all investigated aspects of the distributional properties, as well as their correlations, although the limitations of bias adjustments are not to be disregarded, because all scaling-based methods, as well as the distribution-based techniques assume that patterns from past climate data also exist in present and future time periods. For example, large time gaps between the control and the adjustment period are a source of error, since climate itself is also subject to change. This affects the simple quantile mapping for temperatures in particular, because the results of this method are bound to the range of values of the control period, which means that rising temperatures caused by climate change cannot be mapped in the same way as is the case with the quantile delta mapping, which takes into account the change between the modeled data of the control and adaptation periods. In

addition, the presented adjustment techniques are only applied per grid box. However, this is a problem because grid box variability is not the same as small scale variability, since values are approximated and averaged within grid cells, but the actual climate that occurred is much more differentiated for small regions (Tong et al., 2021, Cannon et al., 2015). Also, none of the methods presented take into account the values of neighboring cells, leading to uncertainty as locale patterns are lost or cannot be adequately accounted for.

The results now serve as a basis for further investigations. The data structures and algorithms implemented in this study, which are available open source and can be used for further development, offer many possibilities for extended research. The next steps could be to work out which method can be applied for a specific research questions, the strengths and weaknesses of individual methods, as well as to find out under which conditions, with which parameters and in which regions or at which times of the year methods for certain climate variables provide particularly good or bad results.

Bibliography

- Beyer, R., Krapp, M., & Manica, A. (2020). An empirical evaluation of bias correction methods for palaeoclimate simulations. *Climate of the Past*, 16(4), 1493–1508. <https://doi.org/10.5194/cp-16-1493-2020>
- Böttinger, M., Meyer, J., Adamidis, P., Anders, I., Behrens, J., Biercamp, J., Bockelmann, H., Buurman, M., Ehbrecht, C., Frauen, C., Fieg, K., Garternicht, U., Heydebreck, D., Kasang, D., Kindermann, S., Lammert, A., Lautenschlager, M., Ludwig, T., Martens, C., ... Florian, Z. (2020). *DKRZ Jahrbuch 2018-2019*. World Data Center for Climate (WDCC) at DKRZ. https://doi.org/10.35095/WDCC/DKRZ_Yearbook_2018-2019
- Cannon, A. J., Sobie, S. R., & Murdock, T. Q. (2015). Bias Correction of GCM Precipitation by Quantile Mapping: How Well Do Methods Preserve Changes in Quantiles and Extremes? *Journal of Climate*, 28(17), 6938–6959. <https://doi.org/10.1175/JCLI-D-14-00754.1>
- Climate Data Operators* [Accessed: June 20, 2022]. (2022). <https://code.mpimet.mpg.de/projects/cdo/>
- CMIP6: The Coupled Model Intercomparison Project* [Accessed: June 20, 2022]. (2020). <https://mpimet.mpg.de/en/science/projects/integrated-activities/translate-to-englisch-cmip6-das-gekoppelte-modellvergleichsprojekt>
- Copernicus Climate Change Service. (2022). *What is Bias Correction?* [Accessed: June 20, 2022]. <https://climate.copernicus.eu/sites/default/files/2021-01/infosheet7.pdf>
- Deutscher Wetterdienst. (2022). *Numerical Weather Prediction Models* [Accessed: June 20, 2022]. https://www.dwd.de/EN/research/weatherforecasting/num_modelling/01_num_weather_prediction_modells/num_weather_prediction_models_node.html
- Edwards, P. N. (2010). *History of Climate Modeling* [Accessed: June 20, 2022]. <https://doi.org/10.1002/wcc.95>
- ERA5-Land hourly data from 1950 to present* [Accessed: June 20, 2022]. (2022). <https://cds.climate.copernicus.eu/cdsapp#!/dataset/reanalysis-era5-land?tab=overview>
- GeoServer 2.20.x User Manual / Installing required NetCDF-4 Native libraries* [Accessed: June 20, 2022]. (2022). <https://docs.geoserver.org/stable/en/user/extensions/netcdf-out/nc4.html>
- Gutjahr, O., Putrasahan, D., Lohmann, K., Jungclaus, J. H., von Storch, J.-S., Brüggemann, N., Haak, H., & Stössel, A. (2019). Max Planck Institute Earth System Model (MPI-ESM1.2) for the High-Resolution Model Intercomparison

- Project (HighResMIP). *Geoscientific Model Development*, 12(7), 3241–3281. <https://doi.org/10.5194/gmd-12-3241-2019>
- Hybrid cubic spherical Bernstein-Bézier patch interpolation* [Accessed: June 20, 2022]. (2022). https://dkrz-sw.gitlab-pages.dkrz.de/yac/d8/de5/interp_method_hcsbb.html
- Jahrhunderthochwasser 2021 in Deutschland* [Accessed: June 20, 2022]. (2021). <https://www.bpb.de/kurz-knapp/hintergrund-aktuell/337277/jahrhunderthochwasser-2021-in-deutschland>
- Jungclaus, J., Bittner, M., Wieners, K.-H., Wachsmann, F., Schupfner, M., Legutke, S., Giorgetta, M., Reick, C., Gayler, V., Haak, H., de Vrese, P., Raddatz, T., Esch, M., Mauritsen, T., von Storch, J.-S., Behrens, J., Brovkin, V., Claussen, M., Crueger, T., ... Roeckner, E. (2019). *MPI-M MPIESM1.2-HR model output prepared for CMIP6 CMIP*. <https://doi.org/10.22033/ESGF/CMIP6.741>
- Katzfuss, M., Stroud, J. R., & Wikle, C. K. (2016). Understanding the Ensemble Kalman Filter. *The American Statistician*, 70(4), 350–357. <https://doi.org/10.1080/00031305.2016.1141709>
- Kotamarthi, R., Hayhoe, K., Mearns, L. O., Wuebbles, D., Jacobs, J., & Jurado, J. (2021). Global Climate Models. In *Downscaling techniques for high-resolution climate projections: From global change to local impacts* (pp. 19–39). Cambridge University Press. <https://doi.org/10.1017/9781108601269.003>
- Maraun, D. (2012). Nonstationarities of Regional Climate Model Biases in European Seasonal Mean Temperature and Precipitation Sums. *Geophysical Research Letters*, 39, 6706–. <https://doi.org/10.1029/2012GL051210>
- Mauritsen, T., Bader, J., Becker, T., Behrens, J., Bittner, M., Brokopf, R., Brovkin, V., Claussen, M., Crueger, T., Esch, M., Fast, I., Fiedler, S., Fläschner, D., Gayler, V., Giorgetta, M., Goll, D., Haak, H., Hagemann, S., Hedemann, C., & Roeckner, E. (2019). Developments in the MPI-M Earth System Model version 1.2 (MPI-ESM 1.2) and its response to increasing CO_2 . *Journal of Advances in Modeling Earth Systems*, 11. <https://doi.org/10.1029/2018MS001400>
- McSweeney, R., & Hausfather, Z. (2018). *Q&A: How do climate models work?* [Accessed: June, 14 2022]. <https://www.carbonbrief.org/qa-how-do-climate-models-work/>
- NetCDF-C Documentation. (2022). *The NetCDF Data Model* [Accessed: June 20, 2022]. https://docs.unidata.ucar.edu/netcdf-c/current/netcdf_data_model.html
- NOAA-CIRES-DOE Twentieth Century Reanalysis (V3): Summary [Accessed: June 20, 2022]. (2022). https://psl.noaa.gov/data/gridded/data.20thC_ReanV3.html
- Rast, S. (2022a). *Modelling MPI-ESM ECHAM* [Accessed: June 20, 2022]. <https://mpimet.mpg.de/en/science/models/mpi-esm/echam>
- Rast, S. (2022b). *Modelling MPI-ESM MPIOM* [Accessed: Mai, 17. 2022]. <https://mpimet.mpg.de/en/science/models/mpi-esm/mpiom>
- Schulzweida, U. (2021). *CDO User Guide: REMAPNN - Nearest neighbor remapping* [Accessed: June 20, 2022]. <https://code.mpimet.mpg.de/projects/cdo/embedded/index.html#x1-6720002.12.5>

- Slivinski, L. C., Compo, G. P., Sardeshmukh, P. D., Whitaker, J. S., McColl, C., Allan, R. J., Brohan, P., Yin, X., Smith, C. A., Spencer, L. J., Vose, R. S., Rohrer, M., Conroy, R. P., Schuster, D. C., Kennedy, J. J., Ashcroft, L., Brönnimann, S., Brunet, M., Camuffo, D., ... Wyszyński, P. (2021). An Evaluation of the Performance of the Twentieth Century Reanalysis Version 3. *Journal of Climate*, *34*(4), 1417–1438. <https://doi.org/10.1175/JCLI-D-20-0505.1>
- Stocker, T. (2011). *Introduction to Climate Modelling*. Springer. <https://doi.org/10.1007/978-3-642-00773-6>
- Teutschbein, C., & Seibert, J. (2012). Bias correction of regional climate model simulations for hydrological climate-change impact studies: Review and evaluation of different methods. *Journal of Hydrology*, *s 456–457*, 12–29. <https://doi.org/10.1016/j.jhydrol.2012.05.052>
- The AWI Earth System Model (AWI-ESM)* [Accessed: June 20, 2022]. (2022). <https://fesom.de/models/awi-esm/>
- The Paris Agreement* [Accessed: June 20, 2022]. (2015). https://unfccc.int/sites/default/files/resource/parisagreement_publication.pdf
- The Twentieth Century Reanalysis Project* [Accessed: Mai, 20. 2022]. (2022). https://psl.noaa.gov/data/20thC_Rean/
- Thejll, P., & Gleisner, H. (2015). Earth’s climate response to a changing Sun; Chapter: Reanalysis data. <https://doi.org/10.1051/978-2-7598-1733-7>
- Tong, Y., Gao, X., Han, Z., Xu, Y., Xu, Y., & Giorgi, F. (2021). Bias correction of temperature and precipitation over China for RCM simulations using the QM and QDM methods. *Climate Dynamics*, *57*. <https://doi.org/10.1007/s00382-020-05447-4>
- Vuuren, D., Edmonds, J., Kainuma, M., Riahi, K., Thomson, A., Hibbard, K., Hurtt, G., Kram, T., Krey, V., Lamarque, J.-F., Masui, T., Meinshausen, M., Nakicenovic, N., Smith, S., & Rose, S. (2011). The representative concentration pathways: an overview. *Climatic Change*, *109*, 5–31. <https://doi.org/10.1007/s10584-011-0148-z>
- Wallace, J. M., & Gutzler, D. S. (1981). Teleconnections in the Geopotential Height Field during the Northern Hemisphere Winter. *Monthly Weather Review*, *109*(4), 784–812. [https://doi.org/10.1175/1520-0493\(1981\)109<0784:TITGHF>2.0.CO;2](https://doi.org/10.1175/1520-0493(1981)109<0784:TITGHF>2.0.CO;2)
- WCRP Coupled Model Intercomparison Project (CMIP)* [Accessed: June 20, 2022]. (2022). <https://www.wcrp-climate.org/wgcm-cmip>
- Why Africa’s heatwaves are a forgotten impact of climate change* [Accessed: June 20, 2022]. (2020). <https://www.carbonbrief.org/guest-post-why-africas-heatwaves-are-a-forgotten-impact-of-climate-change/>

Appendix

A. Figures

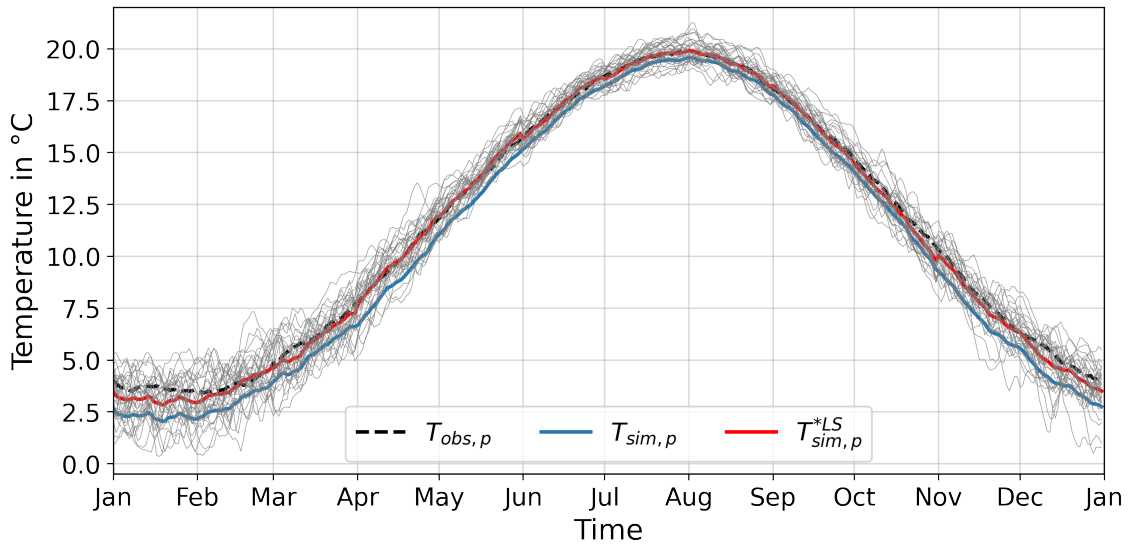


Figure A.1.: Mean 2m air temperatures per day of the year in Europe between 1981 and 2010; grey lines represent the daily mean 2m air temperatures of each individual year in LS-adjusted data

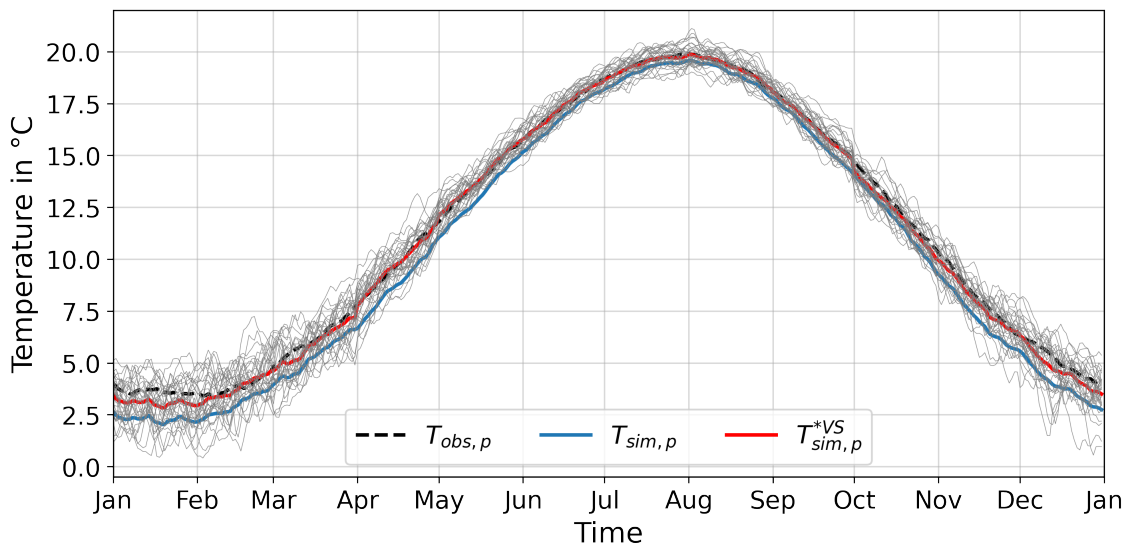


Figure A.2.: Mean 2m air temperatures per day of the year in Europe between 1981 and 2010; grey lines represent the daily mean 2m air temperatures of each individual year in VS-adjusted data

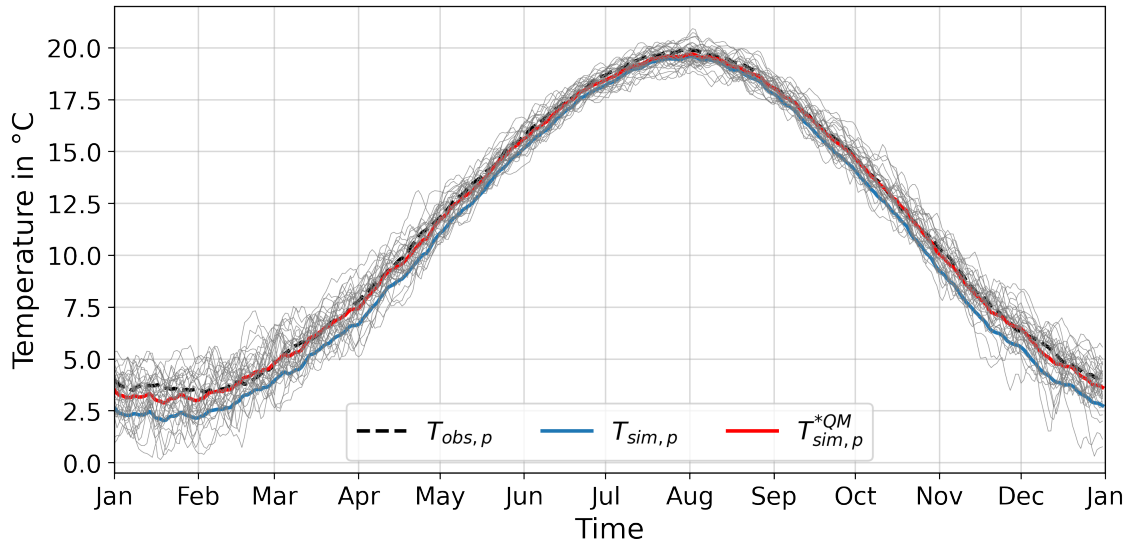


Figure A.3.: Mean 2m air temperatures per day of the year in Europe between 1981 and 2010; grey lines represent the daily mean 2m air temperatures of each individual year in QM-adjusted data

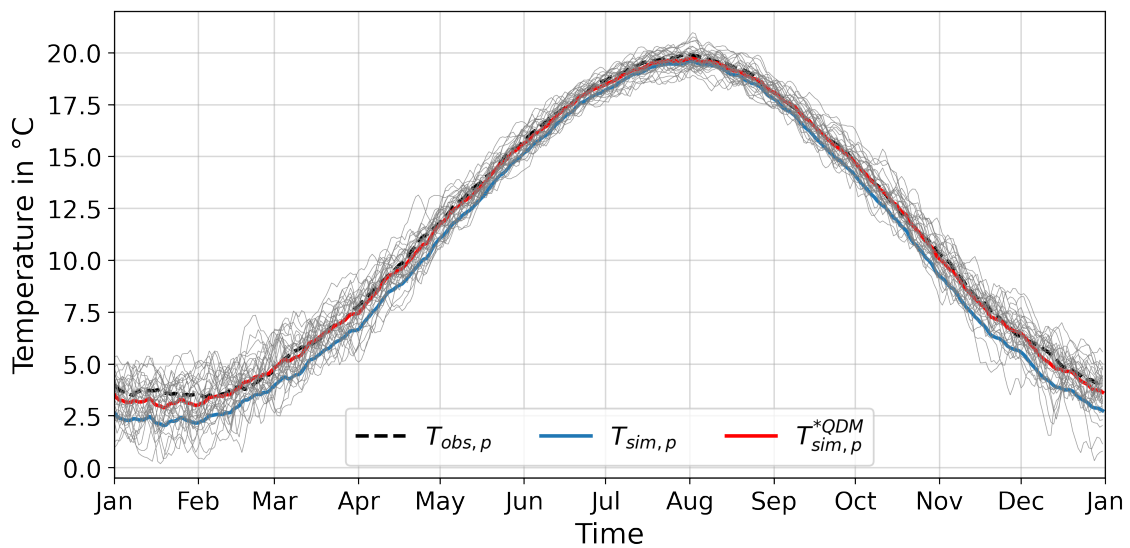


Figure A.4.: Mean 2m air temperatures per day of the year in Europe between 1981 and 2010; grey lines represent the daily mean 2m air temperatures of each individual year in QDM-adjusted data

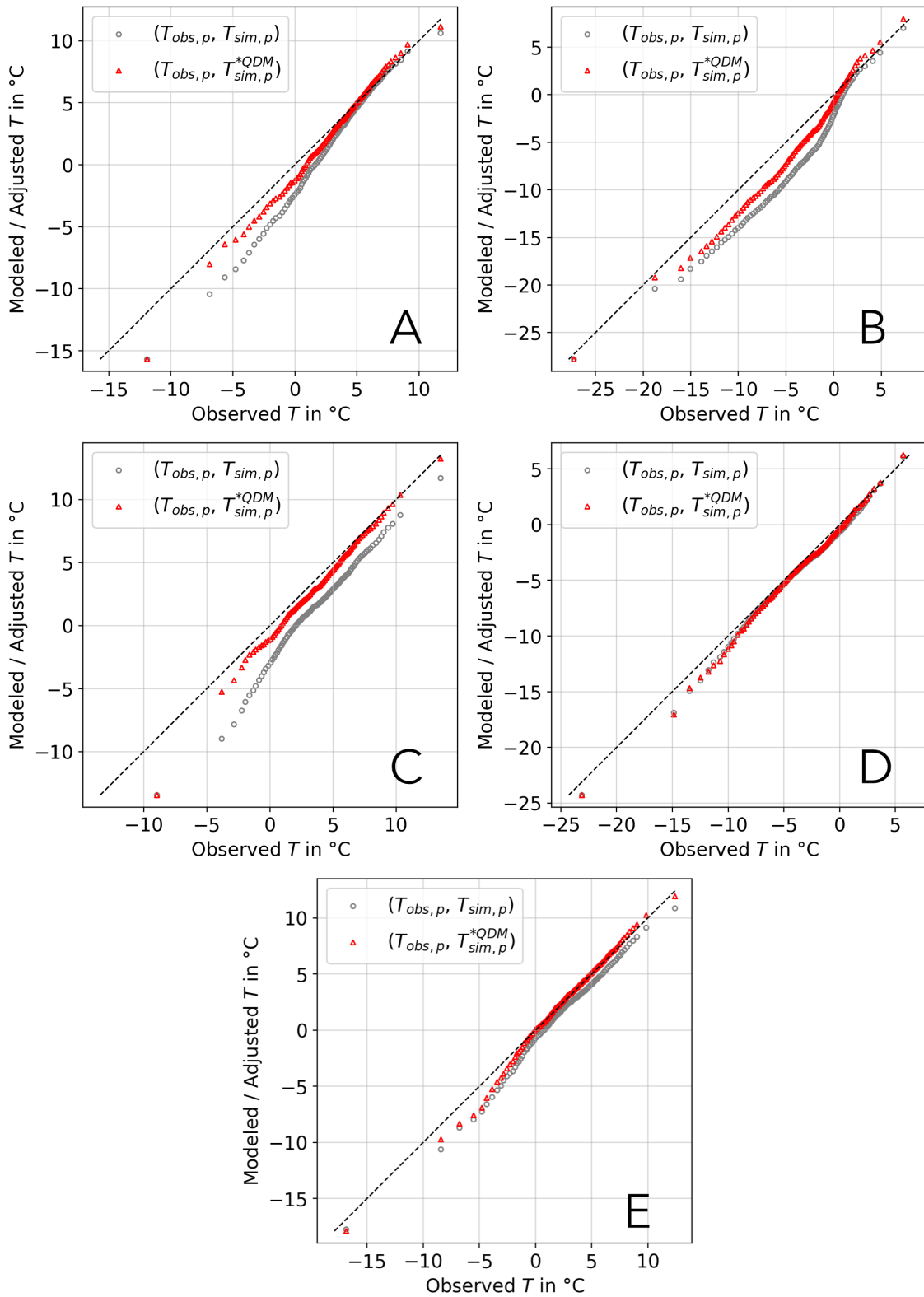


Figure A.5.: Q-Q plots of daily mean 2m air temperatures in winter months (DJF) between 1981 and 2010 in (A) Bremerhaven, (B) Oslo, (C) Milan, (D) Zurich and (E) Lyon

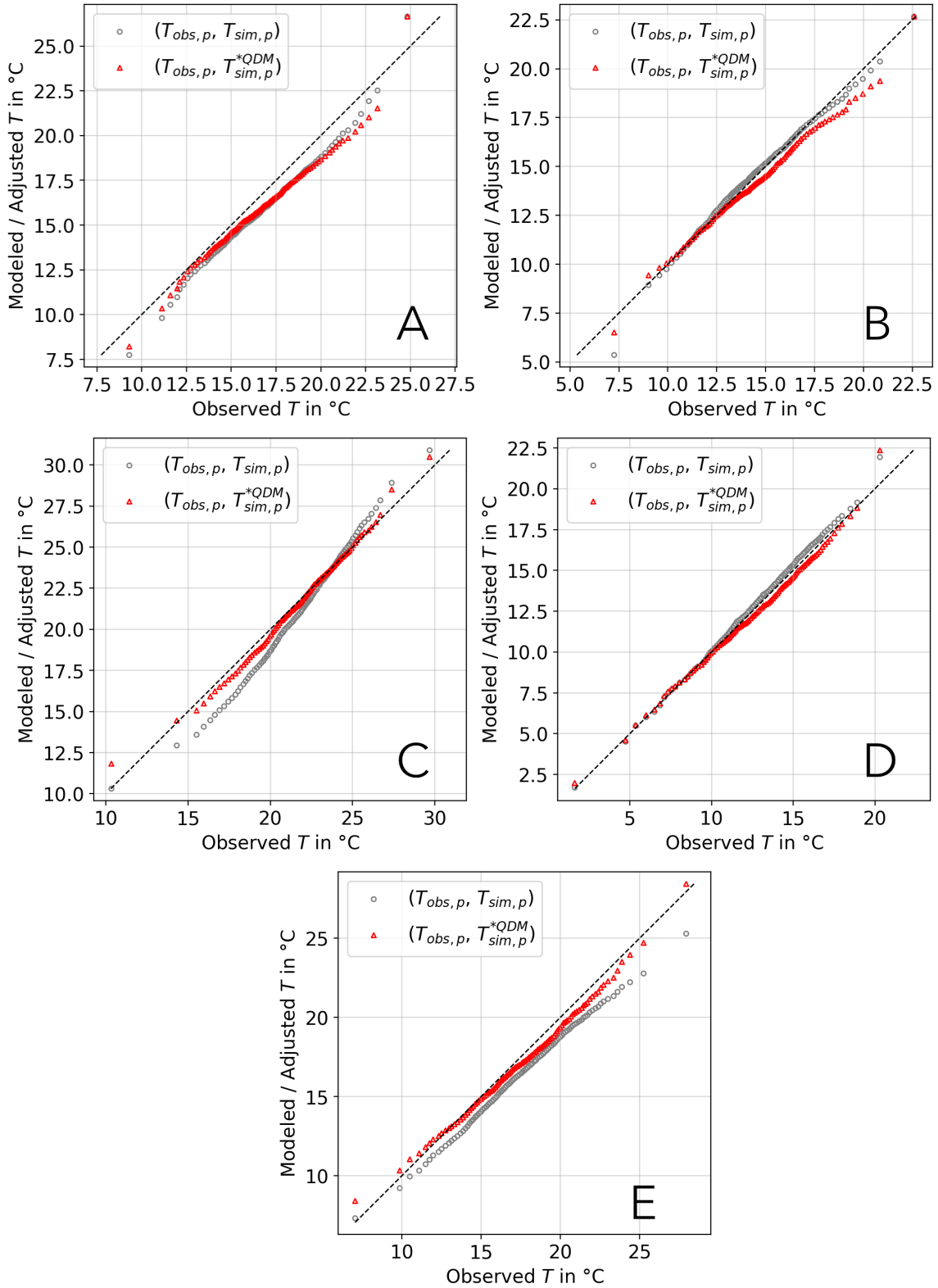


Figure A.6.: Q-Q plots of daily mean 2m air temperatures in summer months (JJA) between 1981 and 2010 in (A) Bremerhaven, (B) Oslo, (C) Milan, (D) Zurich and (E) Lyon

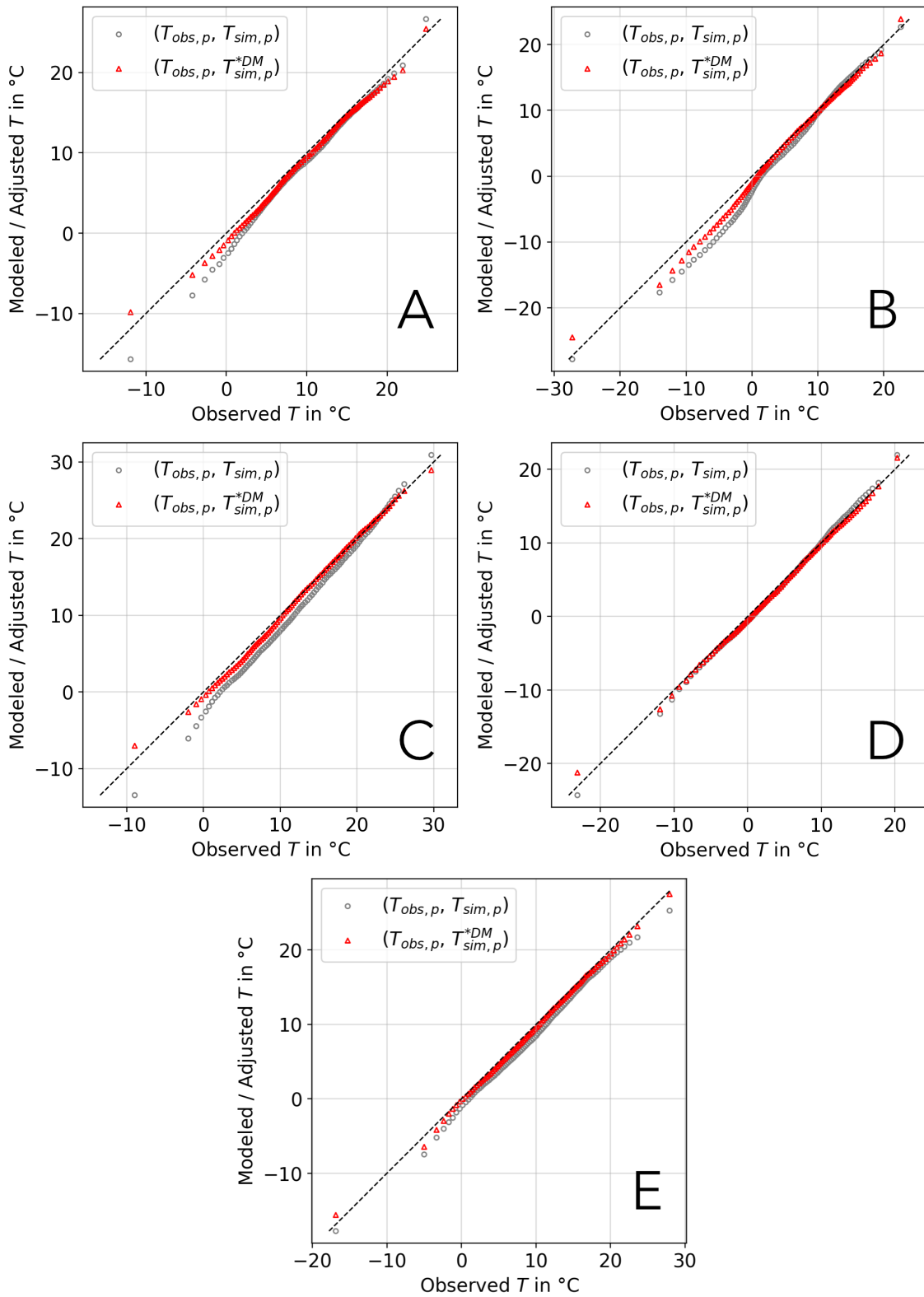


Figure A.7.: Q-Q plots of daily mean 2m air temperatures between 1981 and 2010 in (A) Bremerhaven, (B) Oslo, (C) Milan, (D) Zurich and (E) Lyon

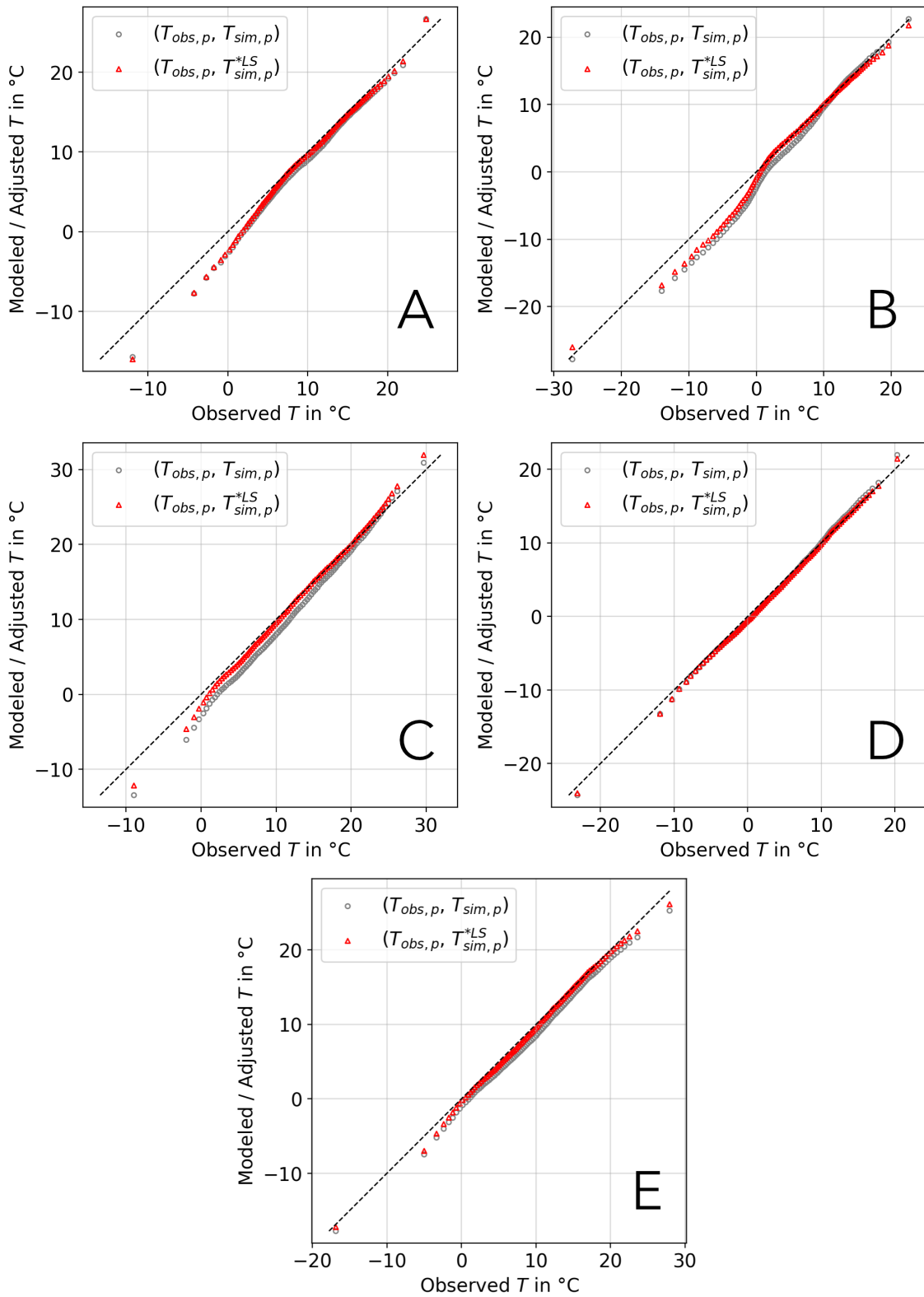


Figure A.8.: Q-Q plots of daily mean 2m air temperatures between 1981 and 2010 in (A) Bremerhaven, (B) Oslo, (C) Milan, (D) Zurich and (E) Lyon

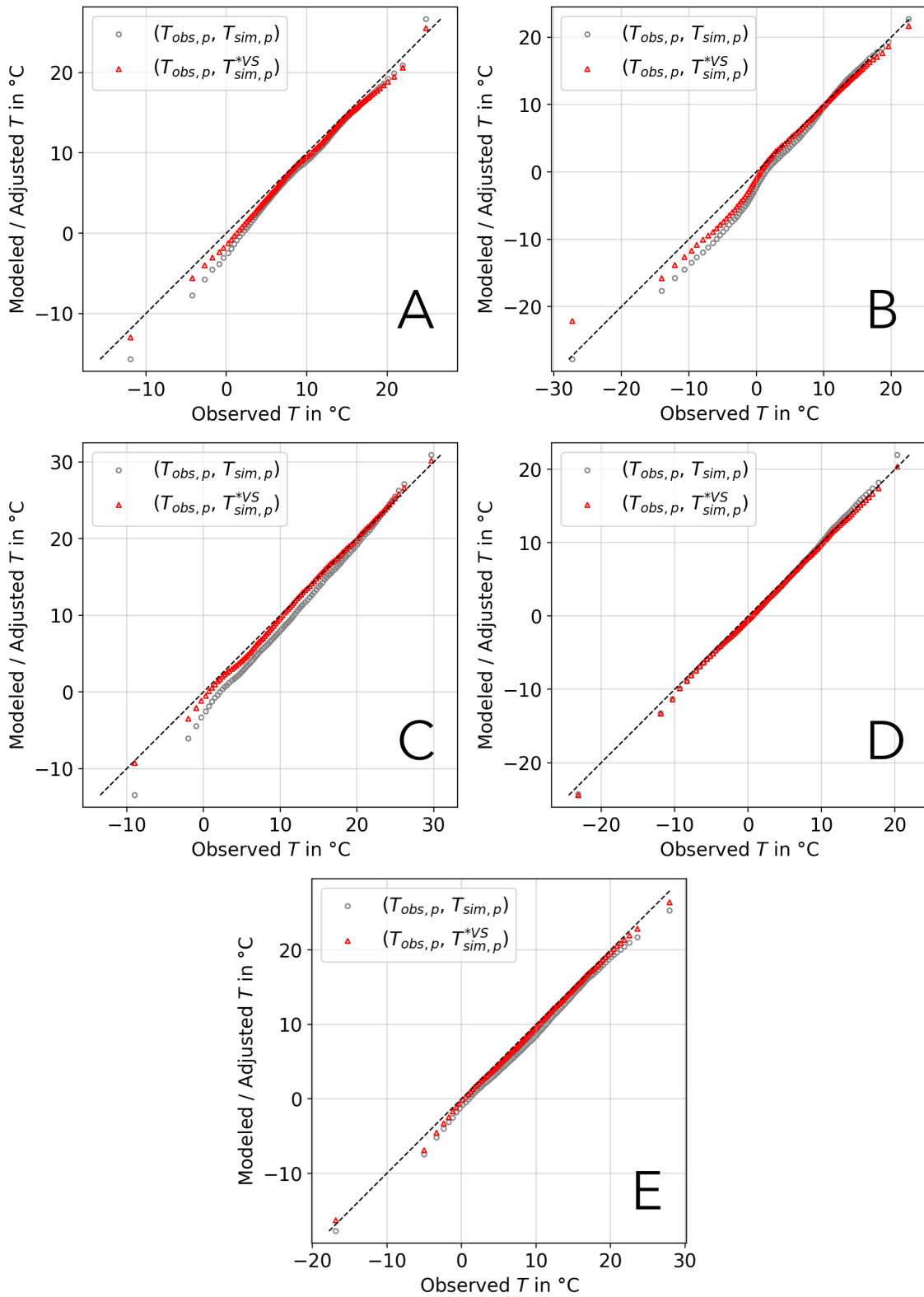


Figure A.9.: Q-Q plots of daily mean 2m air temperatures between 1981 and 2010 in (A) Bremerhaven, (B) Oslo, (C) Milan, (D) Zurich and (E) Lyon

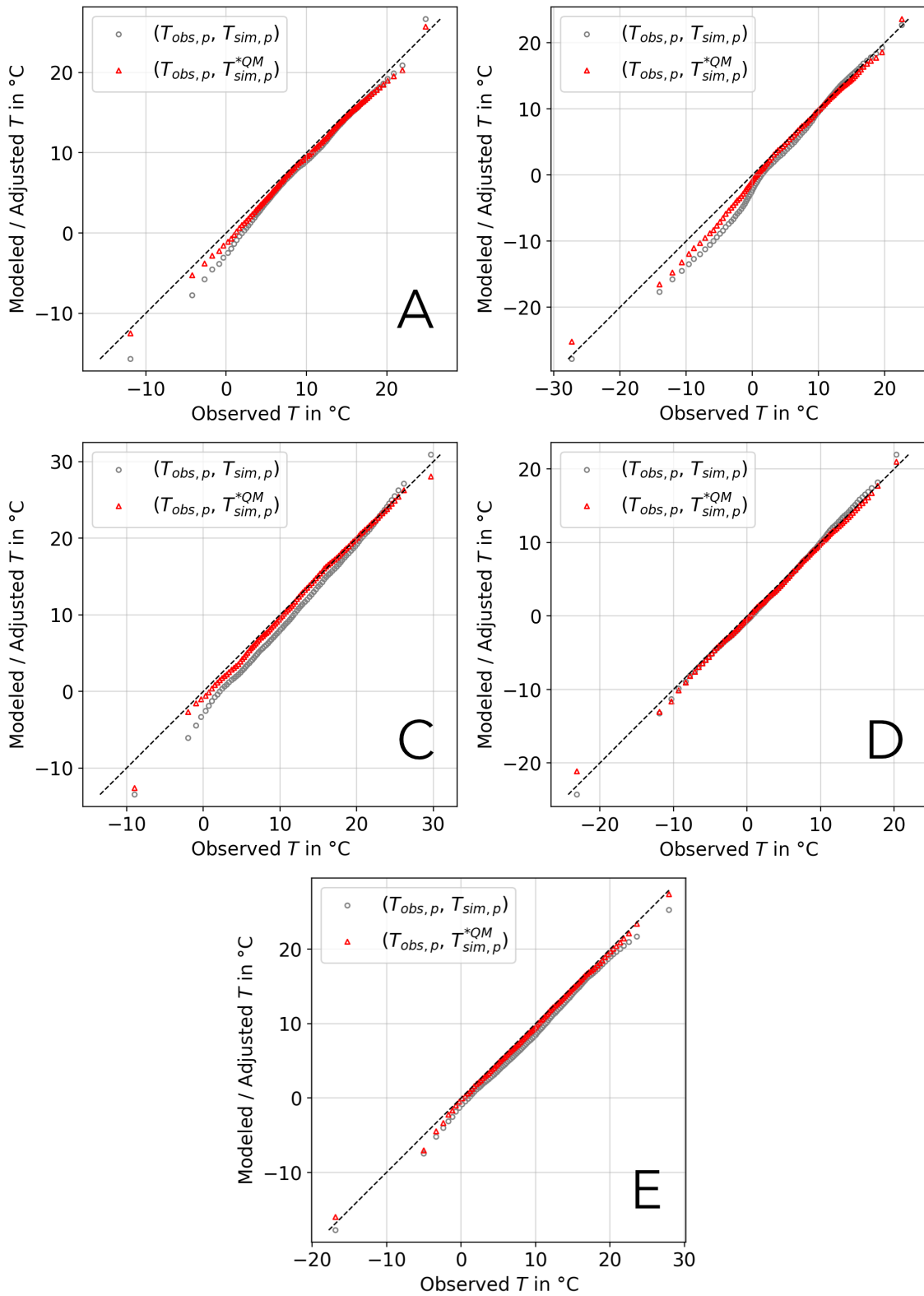


Figure A.10.: Q-Q plots of daily mean 2m air temperatures between 1981 and 2010 in (A) Bremerhaven, (B) Oslo, (C) Milan, (D) Zurich and (E) Lyon