



Estimating snow cover decline using the RSLE in Google Earth Engine

A Caucasus case study

M.S. van Esch

Delft University of Technology

Cover image: (Descloîtres, 2002)

Estimating snow cover decline using the RSLE method in Google Earth Engine

A Caucasus Case Study

by

Matthijs Sake van Esch

in partial fulfilment of the requirements for the degree of

Master of Science
in Civil Engineering

at the Delft University of Technology,
to be defended publicly on December 18th, 2019

Supervisor:	Dr. M. Hrachowitz,	TU Delft
Thesis committee:	Dr. S.L.M. Lhermitte,	TU Delft
	Prof. Dr. ir. S.C. Steele-Dunne,	TU Delft

Abstract

Snow in mountainous areas is of great importance for the water supply in many catchments. It is therefore important to be able to measure the daily snow cover accurately. To get data on snow cover, ground station data is not enough and, in many catchments, not available. Therefore, satellite data is used to measure snow cover. In this thesis the MODIS daily snow cover dataset (MOD10A1) is used. Since these satellite images are taken in the visible spectrum, they are obstructed by clouds. In order to create a complete dataset, the Regional Snowline Elevation Method is used which uses the elevation where the snowline starts to interpolate over the missing data. This method is accurate but is computationally demanding.

Using Google Earth Engine, it is attempted to improve the method. It was needed to adapt the method to be able to use the parallel structure of Google Earth Engine. The method developed combines grid cells with daily images and computes the RSLE for each cell, for each day. The results are exported to a CSV file, reducing the downloaded data from 150GB to 41.6 MB for the 0.50° resolution and 168MB for the 0.25° resolution. The computation time however was not improved with this method, mainly due to the large number of cells which had to be computed daily.

The developed code was used on the Caucasus area. This area was chosen because of the variations in elevation and climate. After the data from Google Earth Engine was downloaded, some filters were applied, and trends on yearly snow cover duration were computed using the Mann-Kendall test. It followed that 17% of the trends in both resolutions were significant and, except for one location, were all decreasing trends. The decreasing trends show a decline of snow cover duration of 1-4 days per year. Looking at regional differences it becomes clear the greatest number of trends can be found in the south-west, where large parts of the Pontic mountains and part of the Armenian plateau are located.

The Google Earth Engine code was able to compute the required data however, it took a long time doing so. Therefore, a more sophisticated code has been developed, making use of the ability to reduce the resolution of an image, and computing the value of pixels at the same time. This code runs quicker, but is at this moment unusable, due to problems with the thresholds and export. Being unable to export an image collection is one of the shortcomings of Google Earth Engine. Others include that download tasks that run out after twelve days without raising an error when starting the task, limits to the amounts of bands used, and sensitivity of the computation time to busyness on servers.

Although trends have been found, there are a few things that need to be improved and need further research. These things are: the filters applied to smoothen the dataset, the use of other trend analyses next to the Mann-Kendall test, the trends on different elevations should be combined with area information, and the resolution in combination with the cloud threshold value should be investigated.

The main goal of decreasing the computation time, also includes the analyses of the data in Python. These analyses took six hours to run, which can be improved by using function-based programming, instead of process-based.

In the end the goal to develop a more efficient method has partly been met by decreasing the amount of downloaded data significantly, even though the running time has not been improved. Using the data from the developed method, decreasing trends were found in snow cover duration over the entire Caucasus but mainly in the South-West, which can greatly influence the water supply to a large part of the Caucasus and its surrounding areas.

Preface

Before you lies the end result of my thesis in the form of a report. It contains information on my findings, but also on my struggles of the past year. Learning a new program/programming language and trying to write a code good enough for a thesis has been a blast at times, and at other times made me close to throwing my laptop out of the window. It might seem a bit crazy but that is just how I like it.

This topic was suggested to me by my supervisor Markus Hrachowitz, who encouraged me to step out of my comfort zone into a topic that was challenging. I have always liked snow, and found remote sensing rather interesting, therefore the combination was perfect. The enthusiasm of Markus made it easier and helped me to keep my motivation up, especially when I decided to switch to Google Earth Engine, going more into detail on the code with some frustrating moments as a result.

For helping with Google Earth Engine, giving creative insights and ways to solve problems which to me seemed unsolvable, I want to thank Stef Lhermitte. I also want to thank Susan Steele-Dunne for her sharp comments and input on my presentations and reports.

A special thanks to my friend Brendan Dalmijn for helping me when I was stuck, and to my father Karst Jan van Esch and my friend Chido Charimari for reading through the report and giving helpful comments.

Before you lies the end result of my thesis in the form of a report. I hope that you will read it with a lot of interest.

Thijs van Esch
's-Gravenhage, December 2019

Content

Abstract	i
Preface	iii
List of figures	vii
List of snippets	viii
1 Introduction	1
2 Case study: Caucasus	3
3 Method	5
3.1 Data used	5
3.1.1 MODIS/Terra Snow Cover Daily	5
3.1.2 GMTED 2010	6
3.2 General approach RSLE	7
3.3 RSLE implementation in GEE	8
3.3.1 Background on Google Earth Engine	9
3.3.2 Data structures of Google Earth Engine	9
3.3.3 Details of the Implementation	10
3.4 Data Processing and Trend Analysis	20
3.4.1 Pre-processing	20
3.4.2 Calculating Snow Cover Duration	20
3.4.3 Mann-Kendall test	22
4 Results	23
4.1 Google Earth Engine Result	23
4.2 Analyses results	24
4.2.1 RSLE development	24
4.2.2 Snow Cover Duration	25
4.2.3 Difference between 0.5° and 0.25° resolution	28
4.2.4 Overall trends'	29
4.2.5 Local trends	32
5 Discussion	36
5.1 Effect of declining SCD trends	36
5.2 Analyses Discussion	37
5.2.1 Filtering the dataset	37
5.2.2 Mann-Kendall Test	38

5.2.3	Snow Covered Area	38
5.2.4	Runtime	38
5.3	Discussion on Google Earth Engine	38
5.3.2	General shortcomings	38
5.3.2	GEE code issues	39
5.3.3	Improved code and its issues	40
5.3.4	Continuing this research in GEE.	42
6.	Conclusion	43
	Bibliography	45
	Appendix A, Code Used	48
	Appendix B, Trend Results for all Elevations	49

List of figures

Figure 1, research area chosen to apply the RSLE-method on	3
Figure 2, one image of the MOD10A1 dataset	6
Figure 3, the digital elevation model used in this thesis	7
Figure 4, estimating the RSLE	7
Figure 5, RSLE methodology	8
Figure 6, visualization of how a Reducer reduces an image	10
Figure 7, flow chart of the adapted RSLE for Google Earth Engine	10
Figure 8, result of the grid cell code	12
Figure 9, binary snow or no snow map	13
Figure 10, split layer dem for the 1500-meter elevation	14
Figure 11, equation 3 applied on an image	15
Figure 12, schematic representations of the nested function	17
Figure 13, example of one grid cell	17
Figure 14, NDSI snow cover class	19
Figure 15, RSLE plotted over three years	24
Figure 16, total percentage of time covered in snow	25
Figure 17, boxplot of the yearly SCD (...) 0.25° resolution	26
Figure 18, boxplot of the yearly SCD (...) 0.50° resolution	26
Figure 19, boxplots of the yearly SCD, only sorted by year	27
Figure 20, colour plots of all the mean snow cover duration	27
Figure 21, difference in mean snow cover duration plots	28
Figure 22, stacked histograms (...) SCD trends at different elevations	28
Figure 23, colour plots for the number of different trends at different elevations	29
Figure 24, SCD trends at 1500 meters for the 0.25° resolution	30
Figure 25, SCD trends at 2500 meters for the 0.25° resolution	31
Figure 26, division to estimate local trends	32
Figure 27, boxplots per elevation for different regions	33
Figure 28, Mann-Kendall p-values per region per elevation	34
Figure 29, Significant SCD trends per area	35

List of snippets

Snippet 1, importing the MODIS dataset	11
Snippet 2, creation of a grid	12
Snippet 3, creating a binary snow or no snow map from the MODIS dataset	13
Snippet 4, creating elevation bands	14
Snippet 5, function to implement equation 3	15
Snippet 6, code to find the RSLE for each day for each grid cell	16
Snippet 7, finding the threshold value for each grid cell	18
Snippet 8, combining the RSLE collection with the cloud threshold	19
Snippet 9, function in python to calculate SCD per year per location	21
Snippet 10, a part of one of the CSVs exported from GEE	23
Snippet 11, the improved code for Google Earth Engine	41

1 Introduction

Snow in mountainous areas is of great importance for the water supply in many catchments. Meltwater provides a steady flow in rivers into summer and its change influences water availability for among others: irrigation, drinking water, industry and energy (Barnett, Adam, & Lettenmaier, 2005; Hatfield et al., 2011; Huning & Aghakouchak, 2018; Viviroli, Dürr, Messerli, Meybeck, & Weingartner, 2007). The timing and amount of meltwater also influences morphology and the ecology of a river system (Bunn & Arthington, 2002) and is of great importance for groundwater levels. When little snow has fallen, ground water tables drop, causing ground water deficits which influences the water availability for plants and trees. An example where this happened is the Californian drought between 2012 and 2016 (Bouman, 2018). Besides being important as a source of water, snow cover influences the albedo of an area. A decrease in snow cover can increase the incoming radiation, heating the surface, causing the snow cover to decrease faster. This will shorten snow seasons in mountainous areas (A. Hall & Qu, 2006), affecting the water availability for its users.

It is desirable to improve models to better predict discharge patterns in the future. Therefore, it is needed to understand the behaviour of melt periods and amount of meltwater expected in mountain ranges and thus it is needed to collect data on the amount of snow, snow cover duration and snow water equivalent (SWE). Data on snow can be obtained in several ways, mainly through weather stations and other on the ground measurement stations or via satellite. Although weather stations have the most variety in data, including height of snow and SWE, the data are local. Interpolation would be possible, but only with enough stations. Especially in less developed countries the availability of ground data can be a problem.

Satellite data can be used to obtain information on snow cover over a larger scale but measuring SWE is difficult. Although there are methods developed to use satellite data for measuring SWE (Yueh et al., 2017), there are no long term remote sensing datasets for SWE. Therefore, it is not possible yet to get a good idea of the behaviour of snow water equivalent through a season on a large scale. For now, the focus is on snow cover. A complete daily dataset is desirable so that the length of the snow season at different elevations can be investigated.

Satellite snow cover pictures are taken using wavelengths within the visible spectrum. As a result, the wavelengths do not penetrate clouds. The problem of clouds being misidentified as snow has mostly been tackled in the MODIS data set, which is the remote sensing dataset used in this thesis (see 3.1 Data used). However, since wavelengths do not penetrate clouds, there is no data on cloudy days. Different methods have been developed to eliminate the problem of cloud interference and create a complete dataset. The method used in this study is the Regional Snowline Elevation method (RSLE) which managed to have an average accuracy of 86% when it was applied to the upper Vah basin (Krajčič, Holko, Perdigão, & Parajka, 2014). The RSLE, and similar methods, have been used on different size regions, from catchments to countries to mountain ranges, for instance the Alps, Slovakia, Norway and the San Joaquin Valley in California (Bouman, 2018; Fugger, 2018; Hauksson, 2017; Krajčič, Holko, & Parajka, 2016; Parajka, Pepe, Rampini, Rossi, & Blöschl, 2010).

Although the RSLE is an accurate method to estimate snow cover, it can be computationally demanding for larger areas. If an area fits into one MODIS image (10° by 10°), and daily analysis is needed for 20 years (which is the timespan MODIS is active), the amount of data to be downloaded is in the order of 130 gigabytes. The region chosen in this study (see 2 *Case study: Caucasus*) is middle of four images. For a daily dataset over a time span of 20 years, this would result in more than 500 gigabytes, more than an average laptop can store. A similar code as used by Bouman (2018) on this area takes several days to run even with a decreased resolution. To be able to investigate large areas, and combine the RSLE data with other datasets (SWE, snow depth, precipitation, radiation) the RSLE needs to be improved, resulting in the following research goal:

Develop a method that decreases the computation time of the RSLE and decreases the data volume to be downloaded.

If the developed method produces results, they can be used to calculate snow cover duration, which in its turn can be used to find out if snow cover duration is decreasing. The analysis is done for the Caucasus area, resulting in the following research question:

Can trends be found in yearly snow cover duration in the Caucasus region?

It is expected that decreasing trends can be found in the Caucasus region. To have a bit more insight in that region, in the next section (2 *Case study: Caucasus*) the case study will be discussed. In the section 3 *Method*, the data, RSLE, adaptations and analyses method will be explained, followed by 4 *Results*. The 5 *Discussion* will be used to review the method used, and propose an alternative, after which the answer to the research question can be given, as well as a conclusion on the effectiveness of the developed method in 6. *Conclusion*.

2 Case study: Caucasus

The adapted RSLE method will be tested in the Caucasus area (Figure 1). This area is chosen because of its variability in climate regions. The Caucasus region is a region located between the Black Sea and the Caspian Sea. The area consists of Armenia, Azerbaijan, Georgia and parts of Iran, Russia and Turkey. It includes the Greater and Lesser Caucasus Mountains, part of the Pontic mountains and the Armenian Plateau. Catchments in the area are the basins of the Tigris-Euphrates system, Van lake, Çoruh river, Kura-Araks river system, Rioni river, Terek river, Kuma river, Laba river and several smaller ones. They drain into either the Black Sea or the Caspian Sea, except for Van lake, which is a sink, and the Tigris-Euphrates system, which drains into the Persian Gulf.

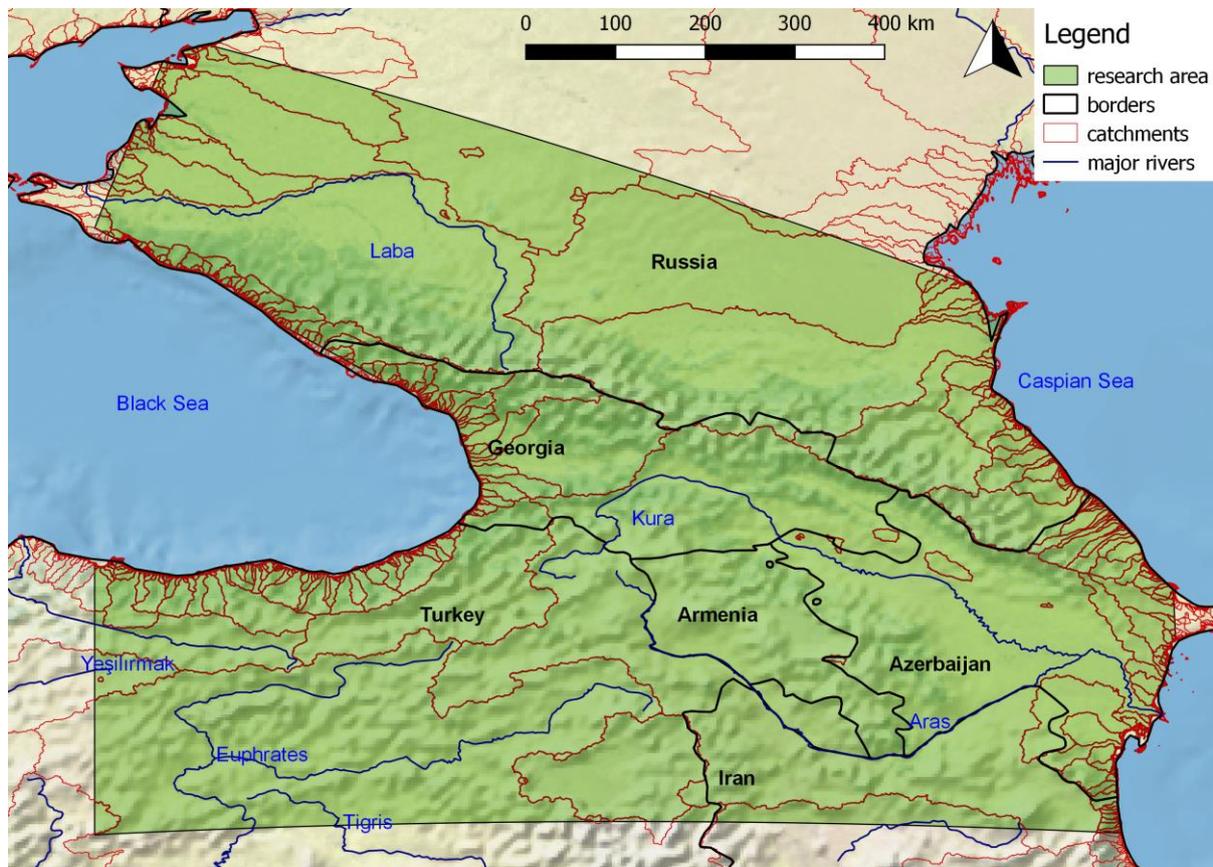


Figure 1, research area chosen to apply the RSLE-method on

The climate ranges from a warm temperate climate in the west, to an steppe climate in the east, with some mountainous areas qualifying as boreal climate, according to the Köppen-Geiger climate classification (Kottek, Grieser, Beck, Rudolf, & Rubel, 2006). The more humid and temperate climate can be found near the Black Sea, and the arid region borders the Caspian Sea.

Water availability studies, studies on snow cover and glacier development are abundant for the Greater/Northern Caucasus. The Greater Caucasus is important for the water supply for Russia and the Southern Caucasus countries (Armenia, Azerbaijan and Georgia). In these mountains, the glacier area has decreased from $1674.9 \pm 70.4 \text{ km}^2$ in 1960 to $1193.2 \pm 54.0 \text{ km}^2$ in 2014 (Tielidze & Wheate, 2018). Average temperature increase has been observed at almost all weather stations,

except for the high mountain station in Terksol (Tashilova, Kesheva, Teunova, & Taubekova, 2016). Yearly precipitation varies from the east to the west with 400-600 mm to 600-800 mm in the western part of the plains. In the mountains the precipitation is 800-1300 mm according to Rets et al. (2018), who carried out a study on runoff trends in the Northern Caucasus, and gave an overview of the different studies concerning precipitation patterns. The predictions on runoff in the near future under the changing climate predict higher runoff during spring, due to earlier and faster melt of glaciers, and a decrease of summer runoff (Hagg, Shahgedanova, Mayer, Lambrecht, & Popovnin, 2010).

In the rest of the Caucasus, studies are less abundant, however there are a few extensive studies in existence, and more local studies as well. A study by Yilmaz & Aalstad (2019) has shown multiple remote sensing sources, one of which is the MODIS dataset, that point to snow pack depletion trends in the Lesser Caucasus mountains, and surrounding areas. The Lesser Caucasus mountains are important because they are the headwaters of the Tigris-Euphrates system, and thus for the water supply of Turkey, Syria, Iraq and Iran. They are also important for the Kura-Araks basin which supplies the South Caucasus countries and has tributaries in Turkey and Iran (FAO, 2016b, 2016a).

Another study was done on the glaciers of the Ararat mountain, which is similar to the behaviour of other ice bodies in the Caucasus and Western Asia (Baldasso et al., 2019). The study concluded that the glaciers have been retreating due to rising temperatures in that area and considering that Mount Ararat is an indicator for the snow pack in the surrounding area, it can be expected that these trends are happening in the rest of the Caucasus as well. Therefore, as the hypothesis stated in the introduction, it is expected that, using the daily MODIS snow dataset, decreasing trends in snow cover will be found.

3 Method

To be able to improve on the RSLE method, a programming platform is needed where one can utilize remote sensing data. One platform in the field of geosciences is Google Earth Engine (GEE). This is an engine where open source data sets are combined with a usefull Application Programming Interface (API). The way the programming works is a bit different which will be explained in 3.3 *RSLE implementation in GEE*. If GEE could decrease computation time and decrease amount of downloaded data, that would make the RSLE method a lot easier to apply to large areas. Before the GEE adaptation can be explained however, some knowledge is needed on data and the RSLE.

3.1 Data used

In order to use the RSLE a snow dataset is needed as well as a Digital Elevation Model (DEM). The sources used for these datasets are:

3.1.1 MODIS/Terra Snow Cover Daily L3 Global 500m Grid, Version 6, MOD10A1 (D. K. Hall & Riggs, 2016)

The Moderate-resolution Imaging Spectroradiometer or MODIS, is an instrument in the Terra and Aqua satellites. In this thesis, the dataset from Terra is used. The satellite manages to make daily images of nearly the entire earth, with spectral bands ranging from 0.405 to 14.385 μm (NASA, n.d.).

The MOD10A1 product has different datasets on the NDSI (Normalized-Difference Snow Index) and snow albedo. In this research, the *NDSI snow cover* and the *NDSI snow cover class* datasets were used in GEE. The Normalized-Difference Snow Index (NDSI) is calculated from MODIS using equation 1, resulting in a fraction of snow cover, which is given in the dataset as a percentage of snow cover in a pixel (Figure 2).

$$\text{NDSI} = \frac{\text{band4} - \text{band6}}{\text{band4} + \text{band6}} \quad (1)$$

$$\begin{aligned} \text{with band 4} &= 0.555\mu\text{m} \\ \text{band 6} &= 1.640\mu\text{m} \end{aligned}$$

The NDSI can be calculated this way because snow is highly reflective in the visible spectrum (band 4, 0.555 μm) and has a low reflectiveness in the shortwave infrared (band 6, 1.640 μm). Clouds do not have the same reflectiveness in the shortwave infrared, which means that they can be distinguished from snow.

If the daily images from MOD10A1 would be downloaded directly from NASA, they would be images of 10° by 10°, which is, depending on longitude and latitude, approximately 1200 km². In order to capture the complete Caucasus, four images would have to be downloaded. However, in GEE the MODIS images are combined into one dataset, so the user does not have to combine the separate MODIS images, but can import the data as a world covering dataset. For more information on the dataset,

the reader is referred to the MODIS Snow Products Collection 6 User Guide (Dorothy K Hall & Riggs, 2015).

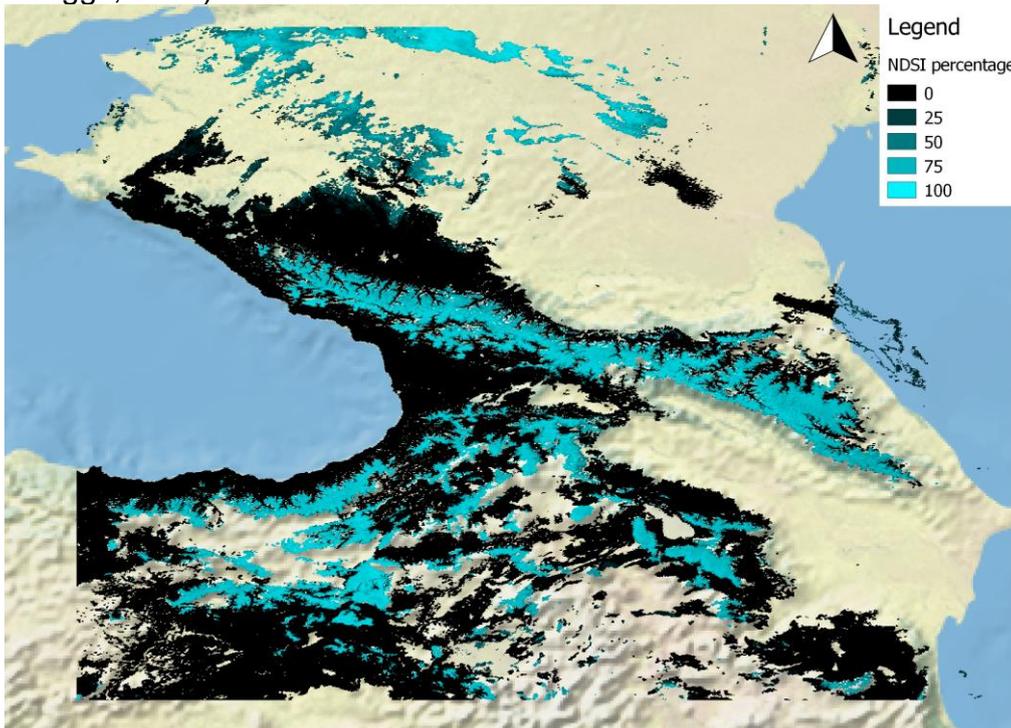


Figure 2, one image of the MOD10A1 dataset, measuring the NDSI percentage on places not obstructed by clouds

3.1.2 GMTED 2010: Global Multi-resolution Terrain Elevation Data 2010 (Danielson & Gesh, 2011)

GMTED2010 is a Digital Elevation Model (DEM) developed by the U.S. Geological Survey (USGS) and the National Geospatial-Intelligence Agency (NGA). The 7.5 arc-second resolution, which has approximately 250 m² pixels, is used (Figure 3).

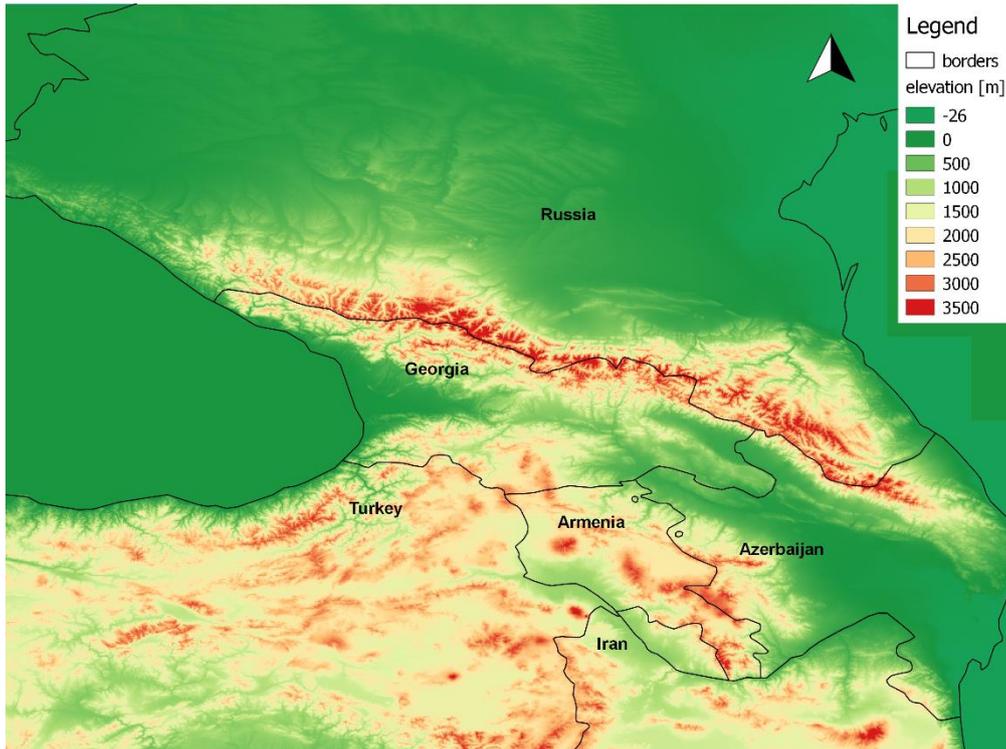


Figure 3, the digital elevation model used in this thesis

3.2 General approach RSLE

In the 1 Introduction it was stated that the RSLE is a method to create data on cloud obstructed days. It is adapted from the SNOWL method (Parajka et al., 2010). The SNOWL method uses the elevation where the snow begins, the snow line, for interpolation. This can be transformed into snow cover (SC), or snow cover duration (SCD) on which trend analyses can be carried out. Krajčí et al. (2014) used an adapted method that estimates the snow line in different subregions to create a more accurate dataset. The RSLE is defined as the elevation where the number of snow-covered pixels (P_s) below summed up with the land pixels (P_l) above, which is called the index of scatter (IS), is minimal (Figure 4). Looking at equation 2 this means that the RSLE is equal to the z for which IS is minimal.

$$IS(z) = \sum (P_s(z) + P_l(z)) \quad (2)$$

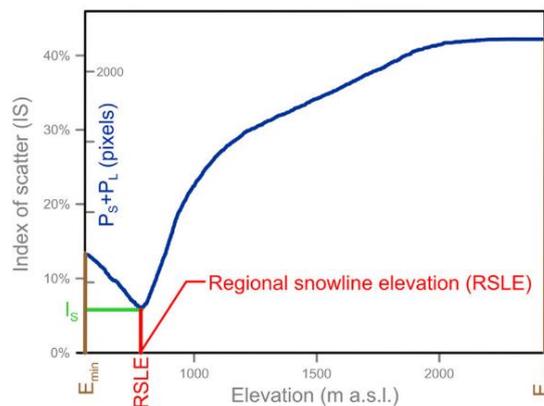


Figure 4, estimating the RSLE, visualizing the math behind the approach (equation 2) (Krajčí et al., 2014)

To be able to apply this to a large area, the area needs to be divided in different cells or tiles. For each cell steps can be followed to find the RSLE, which can be seen in Figure 5.

First, the cloud cover percentage needs to be estimated. Only if this passes the threshold (ξ_c), a RSLE will be calculated for that cell. The next step is the estimation of the amount of snow pixels, again, with a limiting threshold which needs to be passed. Next step is to calculate the sum for the minimum elevation, then add 1 meter and calculate the sum for the next elevation and so on. The RSLE is the elevation where the sum is minimal. Krajčič et al. (2014) used 1 meter steps, however Bouman (2018) used 10 meter steps and Fugger (2018) used 100 meters. Since the area used in this study is comparable in size to Fugger (2018), 100 meter steps are used as well. It was assumed that a smaller step size would cause high running times during the development of the code.

Once the RSLE is calculated it can be used to translate to SCD or other relevant snow cover parameters. Calculating the SCD in a certain area on a certain elevation is done by calculating the number of days that the RSLE is lower than that elevation.

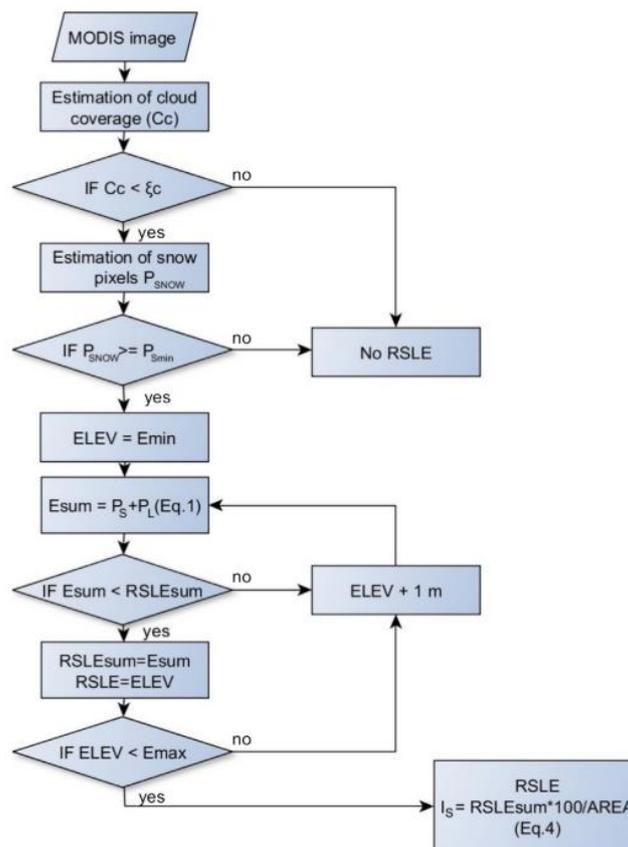


Figure 5, RSLE methodology as used by Krajčič et al. (2014)

3.3 RSLE implementation in GEE

To be able to apply this method for large areas and long timeseries, two problems must be overcome. First is the large amount of data and second, is computation time. In the ideal situation a researcher would only have a small downloadable file with the information needed, for instance the trends in SCD, ready to go. For this thesis the

focus was on finding the RSLE using GEE. Once this was accomplished the resulting dataset was analysed using the programming language Python.

3.3.1 Background on Google Earth Engine

GEE is a powerful scientific tool because it combines satellite data with a useful API (Application Programming Interface) to do analyses and calculations (Gorelick et al., 2017). The data provided by GEE is open source, which is also available from other sources but would take a long time to download. For instance, downloading all the MODIS images needed for this study separately would take days, and result in around 500 GB of data. By doing most of the analysis in GEE, one would only have to download images, text files, or other required data, reducing amount of data considerably. GEE therefore seems like the perfect tool to implement the RSLE in, even though the programming works a bit different.

Programming done by researchers is mostly procedural which incorporates a lot of for, if and else loops. It is a robust way of programming, but the programs generally take a long time to run. GEE can do many calculations faster because it utilizes multiple servers for calculations. For instance, if the daily average amount of snow in images would be calculated over a timespan of a month, each day can be calculated on a different server. To be able to do this the program needs to be constructed in a way that it can utilise multiple servers at the same time.

The parallel structure means that calculations should not be based on previous steps, and thus loops should be avoided. Instead we use the principles of function-based programming which maps created functions over a data set. This means that a function is applied to each element of a collection at the same time, after which the full collection is returned as output. To be able to use this for the RSLE, the method needs to be adapted in order to remove the iterations.

Before demonstrating how these changes work, a short introduction is needed into different data structures in GEE and satellite data in general. For a full introduction, visit the tutorial of GEE (Google Developers, n.d.).

3.3.2 Data structures of Google Earth Engine

The most important GEE data structures are images and features. Features consist of a geometry and one or more properties. Images are standard raster data sets, consisting of several bands with data and additional information. Both features and images can be stored in collections, for instance twenty years of daily images of MODIS are called an image collection. Several different predefined, or self-constructed functions can be used on images, features and collections.

Reducers are special functions which take a multiband image, feature or collection as an input and return a reduced version of the input. For instance, reducing an image with a *mean* reducer results in an image with the mean of all the bands combined into a one-band image. Reducing an image collection with a mean, results in an image with the mean of every image in the collection (see Figure 6). Reducing an image with three bands will thus result in one image with one band. Reducing an image collection consisting of images with three bands, will result in one image with three bands. To get a mean value of all the bands for every day separately in an Image Collection, the reducer must be incorporated into a function and mapped over the collection, instead

of being applied directly. Mapping a reducer means it is applied to every image separately, giving an output of a collection, with reduced images, instead of a reduced collection. Using reducers and other functions, the RSLE is adapted to remove most iterations.

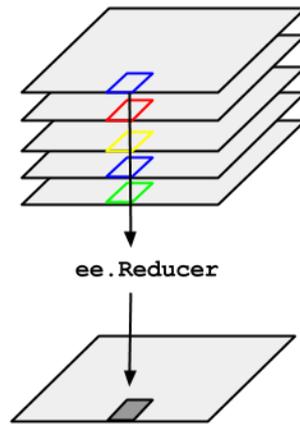


Figure 6, visualization of how a Reducer reduces an image, by taking pixelwise the best value of different bands and combining them into one image (Google Developers, n.d.)

3.3.3 Details of the Implementation

In this section a detailed explanation is given for each of the steps in Figure 7. The Link to the code can be found in Appendix A, Code Used

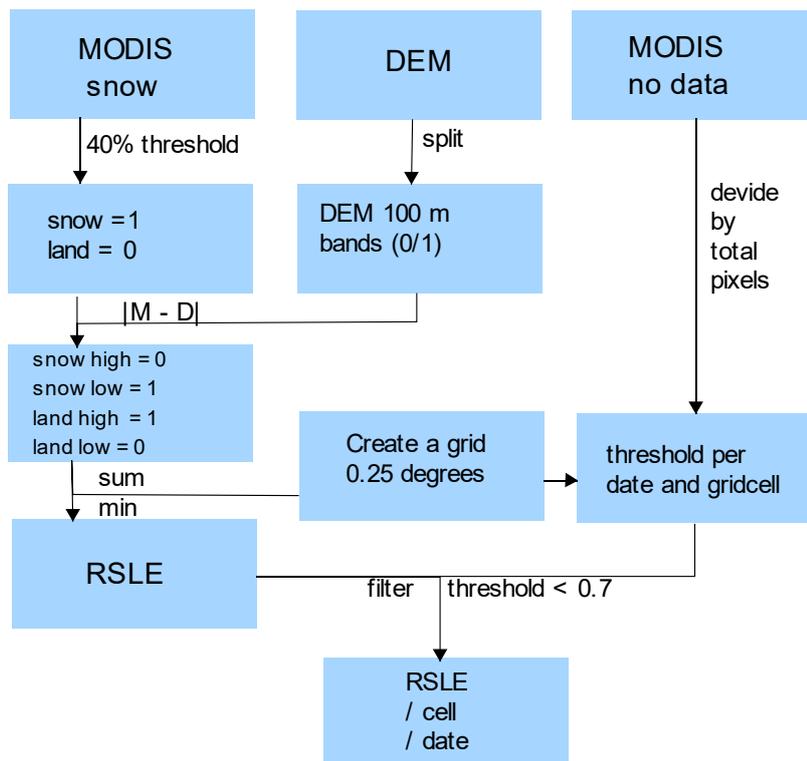


Figure 7, flow chart of the adapted RSLE for Google Earth Engine

3.3.3.1 Importing data and pre-processing

The very first step is to import the necessary data and to select the research area. The research area can be uploaded in different formats or can be drawn on the map. In this

case, it was drawn in a different piece of code and uploaded in this code as a table called *Caucasus*.

The upload and selection of data can be achieved by searching for the necessary collection and selecting import, or by directly writing it in the code. The DEM was entered in the code using the import and the *MODIS NDSI_Snow_Cover* dataset (Figure 2) was entered using a piece of code (Snippet 1), in which the data was filtered to the area (*Caucasus*) and the desired timespan.

```
//snow cover information import
var MOD = ee.ImageCollection('MODIS/006/MOD10A1')
  .select('NDSI_Snow_Cover')
  .filterBounds(Caucasus).filterDate('2019-02-01','2019-05-31');
```

Snippet 1, importing the MODIS dataset and filtering to the research area and desired date

To start pre-processing the data a grid was created using loops, since this is not a demanding calculation and does not need to be split over several servers. The result is a collection of features, in which each feature is a grid cell (Snippet 2, Figure 8). It was chosen to use two resolutions, 0.25° by 0.25° (30 by 40 km²) and 0.50° by 0.50°. The 0.25° resolution is a bit larger than the cell size used by Fugger (2018), who used 50 pixels (25 x 30 km²). For GEE it is easier to use degrees instead of number of pixels. The 0.50° resolution is used to see if a larger cell size makes a big difference for the trend analyses, since a larger resolution decreases the computation time.

```
// 1) Create bounding box
var lon_start = 37.5;
var lon_end = 49.5;
var lat_start = 38.0;
var lat_end = 47.0;

// 2) Decide cells
var lon_edge = (lon_end-lon_start)/48;
var lat_edge = (lat_end-lat_start)/36;

// 3) Create the grid
var polys = [];
var cell_id = [0,0];
for (var lon = lon_start; lon < lon_end; lon += lon_edge) {
  var x1 = lon;
  var x2 = lon + lon_edge;
  for (var lat = lat_start; lat < lat_end; lat += lat_edge) {
    cell_id = [lon,lat];
    var y1 = lat;
    var y2 = lat + lat_edge;
    polys.push(ee.Feature(ee.Geometry.Rectangle(x1, y1, x2, y2), {label:
cell_id})));
  }
}
var grid = ee.FeatureCollection(polys);
var grid = grid.filterBounds(Caucasus);
```

Snippet 2, creation of a grid, in this case for the 0.25° resolution

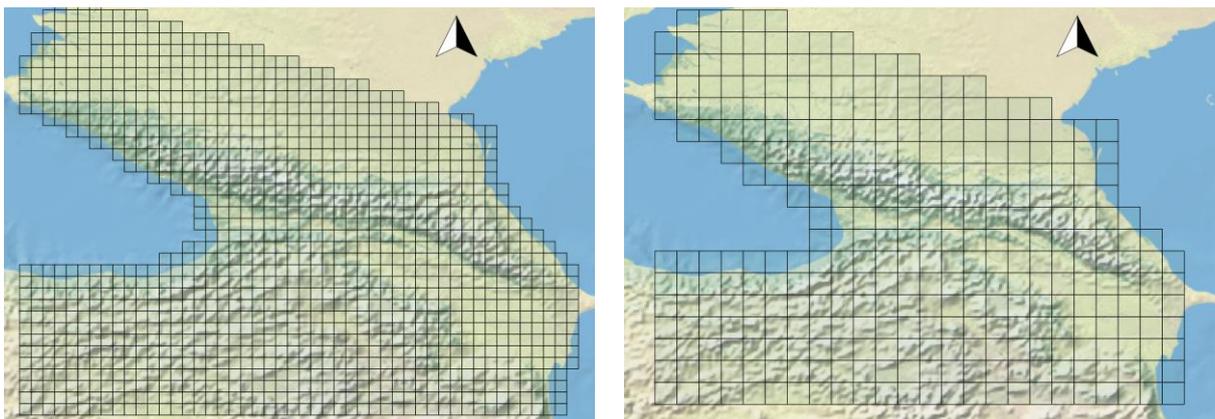


Figure 8, result of the grid cell code, with at the left the 0.25° resolution and at the right the 0.50° resolution.

The snow map was turned from a snow percentage map in to a binary map, where a pixel value of 1 represents a pixel with more than 40% snow (Snippet 3, Figure 9). Different thresholds are used in literature, however, 40% is used often and is deemed adequate for spatial resolutions of 500 m and up (Härer, Bernhardt, Siebers, & Schulz, 2018). Although the definition is somewhat arbitrary, the threshold ensures that the detected snow in a pixel is significant and not a measuring error.

```
//add first 40% snow cover threshold for pixels in  
var MOD = MOD.map(function(img){ return img.gte(40).clip(grid)});
```

Snippet 3, creating a binary snow or no snow map from the MODIS dataset

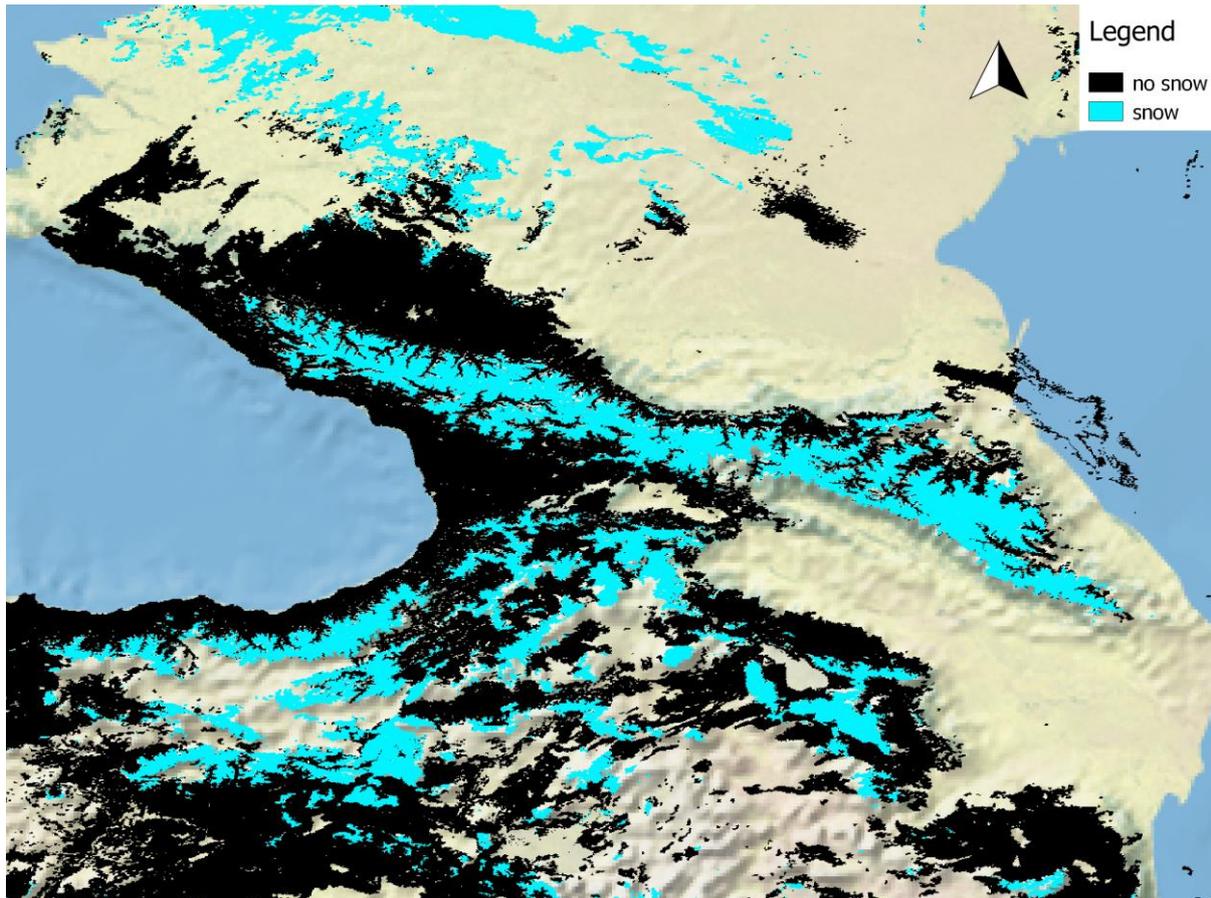


Figure 9, binary snow or no snow map, by assigning all NDSI values higher than 40 a 1, which means snow, and all values lower than 40 a 0, which means no snow

The last step of the pre-processing is to prepare the elevation map. Since the iterative loops cannot be used to find the snow line at all elevations, the snow line function needs to be mapped over a collection containing all the elevation steps. This is created by splitting the elevation map into several bands (Snippet 4, Figure 10). The bands start at 0 m and end at 3500 m, since most peaks above 3500 m are small, and there are only a few. Each band has pixels with 0s and 1s, which respectively represent elevations below or above the step of that band. For example, for the band of the 200-meter step, all pixels below 200 m are defined as 0 and everything above is defined as a 1.

```
//Height is the DEM
var elev100 = Height.gt(100).clip(grid).rename('0100');
var elev200 = Height.gt(200).clip(grid).rename('0200');
      :
      :
var elev3400 = Height.gt(3400).clip(grid).rename('3400');
var elev3500 = Height.gt(3500).clip(grid).rename('3500');

var elev = ee.Image.cat([elev100,elev200, (...),elev3300,elev3400,elev3500]);
```

Snippet 4, creating elevation bands. *gt* means greater than, giving a pixel the value 1 if it is greater than the defined value, and a 0 if it is not

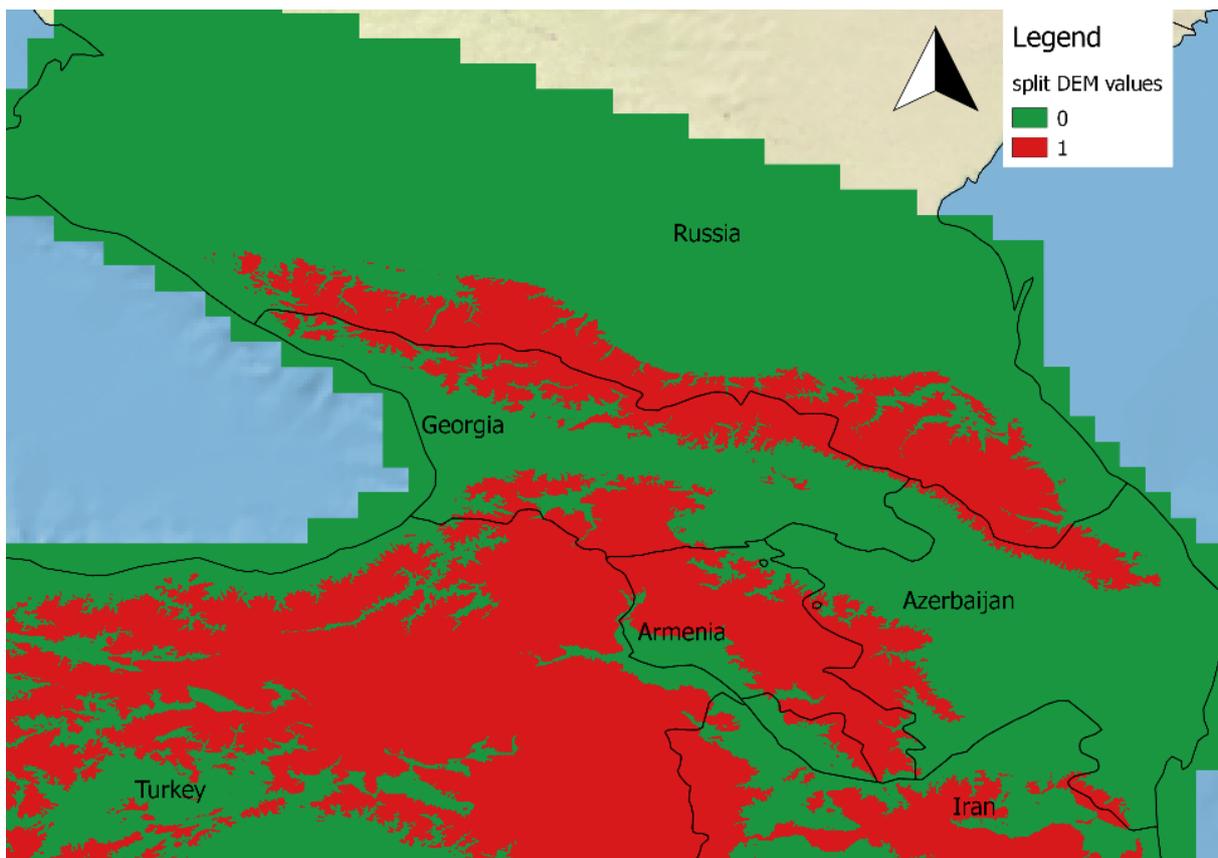


Figure 10, split layer dem for the 1500-meter elevation. Red means above 1500 meters, green means below 1500 meters.

3.3.3.2 Calculating the RSLE

In order to convert the snow and elevation map into maps which can be used for the RSLE analyses (equation 2), a function is made in which the elevation map is subtracted from the snow map for each snow pixel, and the absolute value is taken (equation 3, Snippet 5, Figure 11). The result of this function is an image containing 0s and 1s. Applying this to our example, this would result in a 200m band with 0s, meaning snow above 200m or land below 200m, and 1s, meaning snow below 200m and land above 200m. summing up to get the Index of Scatter and finding the elevation for which this is minimal, would thus result in the RSLE.

$$|S - E(z)| = X(z) \quad (3)$$

$$S = \begin{cases} 0 & \text{if no snow} \\ 1 & \text{if snow} \end{cases}$$

$$E(z) = \begin{cases} 0 & \text{if below } z \\ 1 & \text{if above } z \end{cases}$$

$$X(z) = \begin{cases} 0 & \text{snow above } z \text{ or no snow below } z \\ 1 & \text{snow below } z \text{ or no snow above } z \end{cases}$$

$z = \text{elevation step}$

```
var x = function(img) {return img.subtract(elev).abs()};
var B = MOD.map(x);
```

Snippet 5, function to implement equation 3

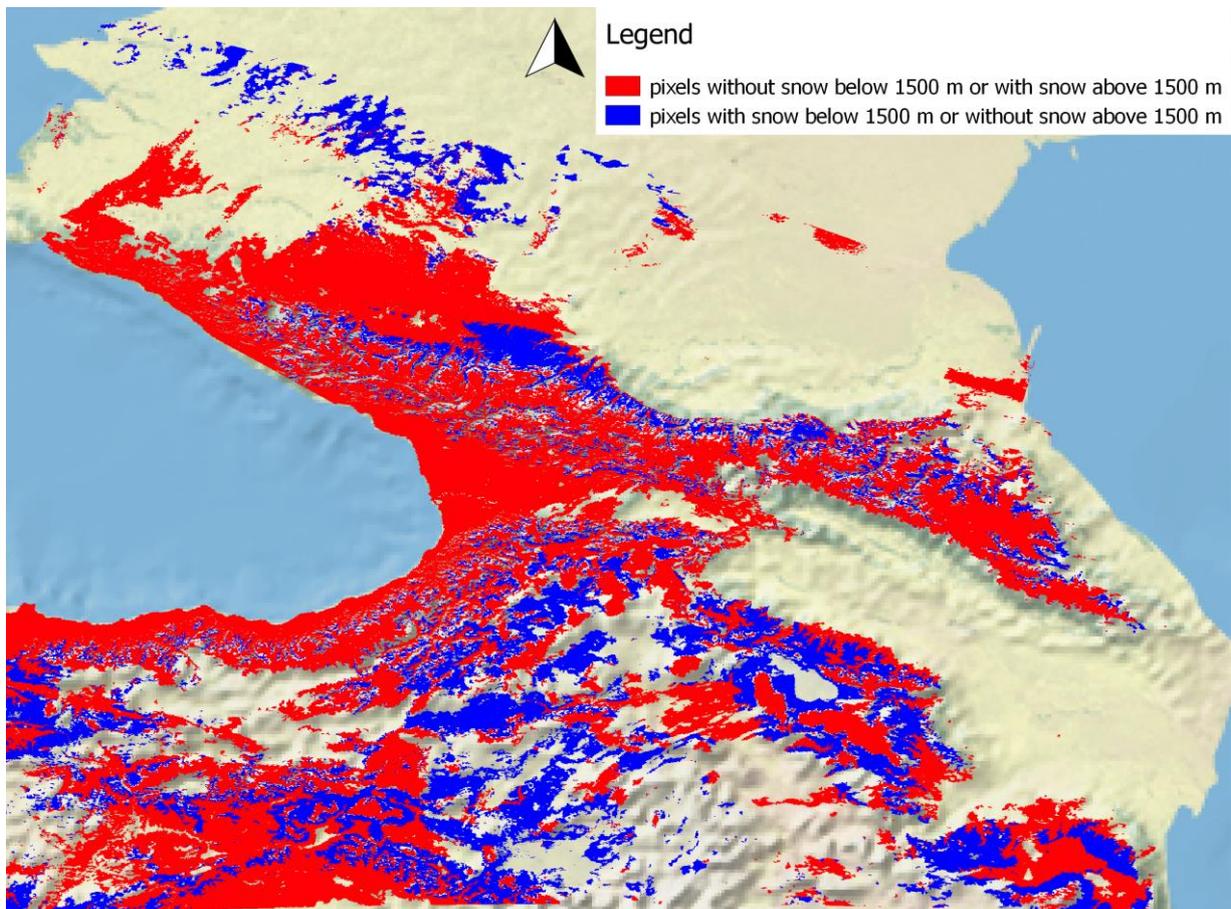


Figure 11, equation 3 applied on an image. This is the 1500-meter band, red having value 0 (no snow below 1500 meters, snow above 1500 meters), and blue having value 1 (land above 1500 meters and snow below 1500 meters).

The dataset now consists of an image collection with daily images. An image contains 34 elevation bands. Each band has pixels with either values 0 or 1. If all the 1s are summed up in a region the result will be the same as equation 2, giving the index of scatter (IS), which leads to the RSLE if the minimum IS is found.

This summation needs to be done for each grid cell and the minimum band must be selected for that grid cell (Figure 13) for each day. A function is written within a function (nested) to be able to combine the images and features (Snippet 6, Figure 12). However, this has consequences for the running time, since the first function relies now on the second function and cannot be completely computed in parallel. The function makes a sum of all the 1s from equation 3 and puts those values in a feature. A reducer is applied to this feature to find the minimum band, corresponding to the RSLE. This is done for each feature, for each day of the total dataset. The result is a feature collection, with for each feature the location and the RSLE.

```

var unweighted = function(img) {
  var uni = img.reduceRegions({
    reducer: ee.Reducer.sum().unweighted(),
    collection: grid2,
    scale: 500}).map(function(feats) {
    var id = ee.String(feats.get('system:index')).cat('-');
    var date = ee.String(img.get('system:index'));
    var bandNames = img.bandNames().map(function(name) {return
ee.Number.parse(name)});
    var values = ee.Feature(feats).toDictionary(img.bandNames().values());
    var array = ee.Array.cat([values, bandNames], 1);
    var min = array.reduce(ee.Reducer.min(2), [0, 1]);
    // get min and max values and cast to numbers
    var minh = ee.Number(feats.get('min'));
    var maxh = ee.Number(feats.get('max'));
    // set a first value for key
    var key = min.get([0,1]).toInt();
    // maximum elevation check
    key = ee.Algorithms.If({condition: min.get([0,1]).toInt().gt(maxh),
      trueCase: maxh,
      falseCase: key
    });
    // minimum elevation check
    key = ee.Algorithms.If({condition: minh.gt(min.get([0,1]).toInt()),
      trueCase: minh,
      falseCase: key
    });
    return
ee.Feature(feats.setMulti(ee.Dictionary.fromLists(['Elevation','id'], [key,
id])));
  });
  return uni;
};

```

Snippet 6, code to find the RSLE for each day for each grid cell

Estimating snow cover decline using the RSLE method in Google Earth Engine

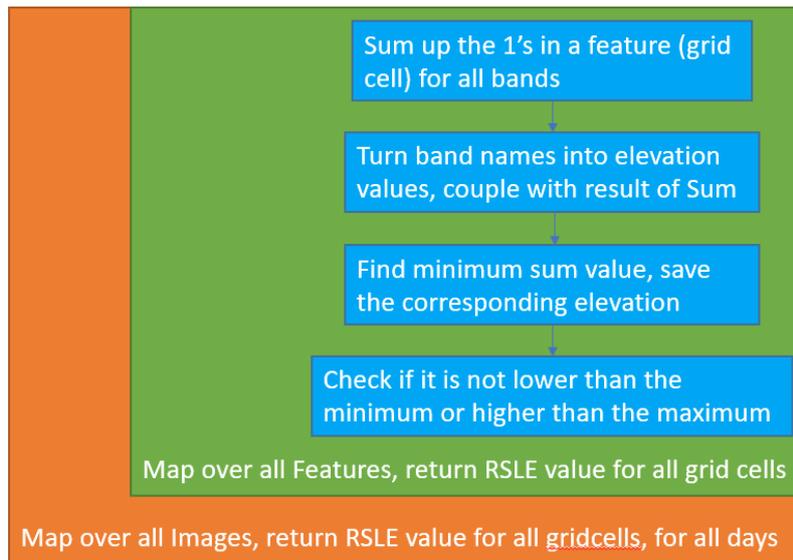


Figure 12, schematic representations of the nested function used to calculate the RSLE

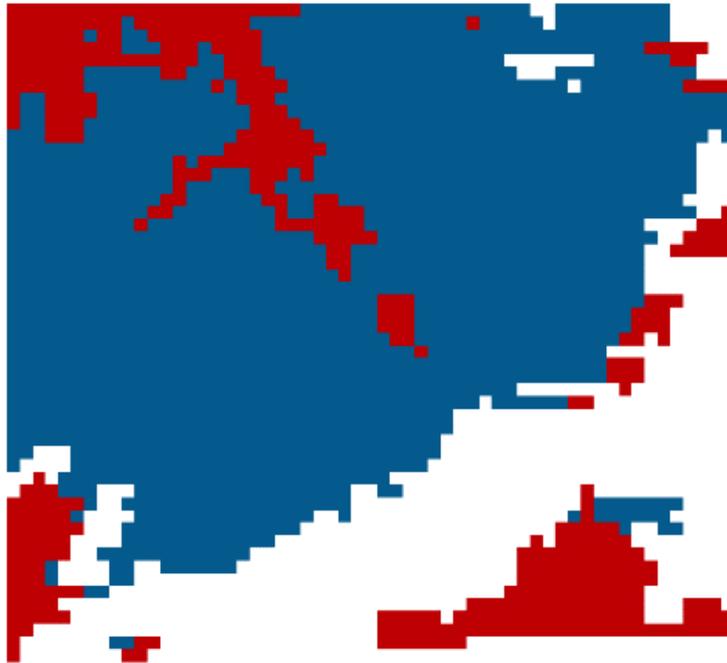


Figure 13, example of one grid cell which is approximately 30x40 km. To find the RSLE, the band must be found where the blue area is the smallest. The white is missing data.

3.3.3.3 Exporting the RSLE

Before the RSLE dataset can be exported, the cloud cover threshold (ξ_c) of 0.7 must be met. This threshold entails that a RSLE can only be calculated if the cloud cover in that area is less than 70% (Krajčí et al., 2014). As an extra security, the threshold in this thesis is extended for all the missing data pixels. All the missing data pixels in the *NDSI snow cover class* (Figure 14) band were given a number 1. A function was created to sum up these 1s and divide that sum by the total amount of pixels to get the threshold for each grid cell (Snippet 7).

The grid cell number is only saved if the threshold (<0.7) is met. This collection was then joined with to the RSLE collection, with the condition that the location of the threshold collection was in the RSLE collection (Snippet 8). The final feature collection is exported as a Comma Separated Values file (CSV), which is a text file, and can be used for further analyses (Snippet 10). Each row in the file is represented by a feature. In the columns the following parameters are stored:

- * id, which consists of a date and an id number
- * minimum elevation (added as a check)
- * maximum elevation
- * RSLE
- * Label (which consists of the longitude and latitude)

```
//step 1: remap no data (cloud, water, no data, etc) to 1
var threshold = Cloud.map(function(img) {
  var NoSnow =
img.clip(grid).remap([200,201,211,237,239,250,254],[1,1,1,1,1,1,1]);
  return ee.Image(NoSnow);
});
//step2: create total amount of pixel band
var totalA =
B.select(0).filterMetadata('system:index','equals','2019_05_02').map(
  function(img){return img.remap([0,1],[1,1])});
var total = totalA.merge(threshold.filterDate('2019-05-02')).mosaic();
var threshold2 = threshold.map(function(img){return
img.addBands(total.rename('total'))});
//step 3: sum up total amount band, and no data band
var threshold3 = threshold2.map(function(img) {
  var red = img.reduceRegions({
    reducer:ee.Reducer.sum().unweighted(),
    collection:grid, scale:500}).map(function(feet) {
    var id = ee.String(feet.get('system:index')).cat('-
').cat(ee.String(img.get('system:index')));
    return ee.Feature(feet.set('id',id))});
  return red }).flatten();
//step 4: make the threshold value by dividing no data by total.
var threshold4 = threshold3.map(function(ft) {
  var id = ee.String(ft.get('id'));
  var r = ee.Number(ft.get('remapped'));
  var t = ee.Number(ft.get('total'));
  return ee.Feature(ft.setMulti({x:
r.divide(t),remapped:null,total:null,id: id}));
});
var threshold5 = threshold4.filter(ee.Filter.lessThanOrEqualTo('x',0.7));
```

Snippet 7, finding the threshold value for each grid cell by dividing a map with all the no data pixels set to 1 with a map where all the pixels are set to 1

Estimating snow cover decline using the RSLE method in Google Earth Engine

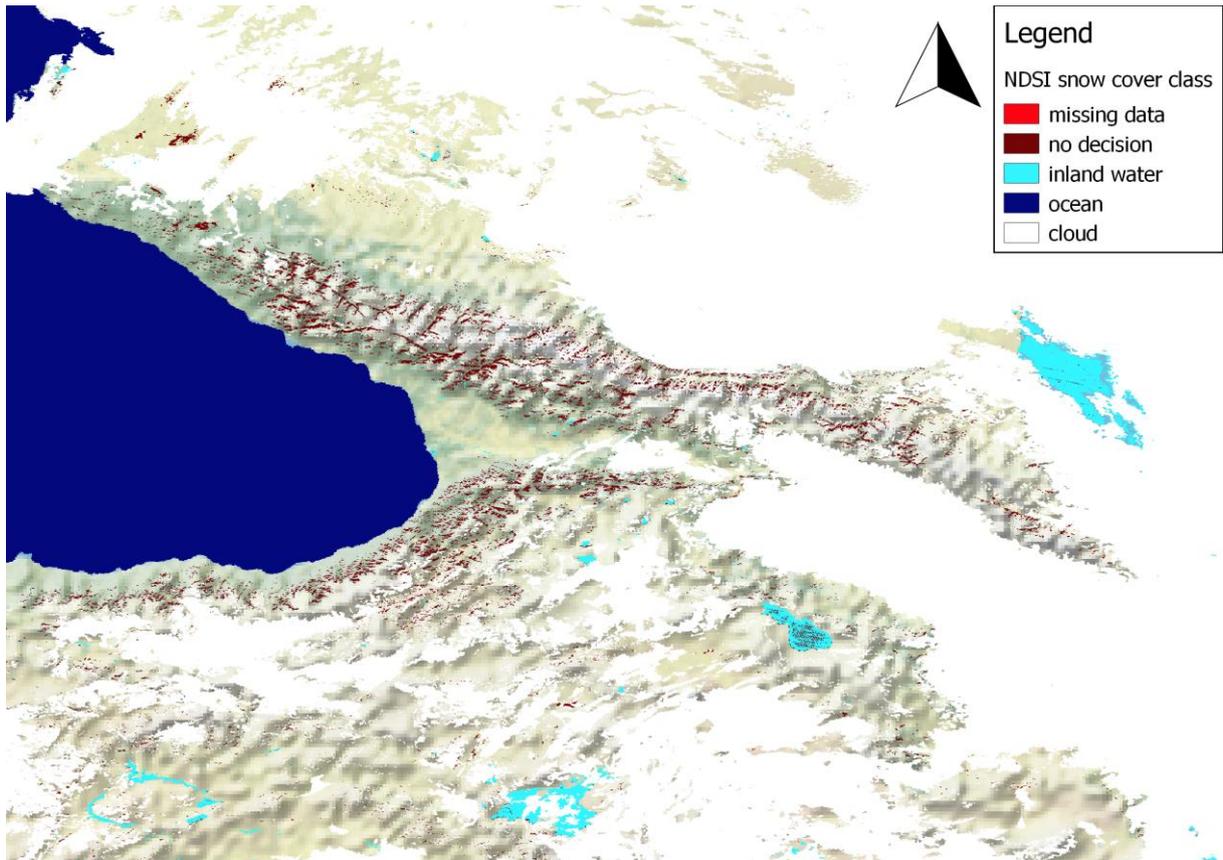


Figure 14, NDSI snow cover class, all the different datapoints that are marked as no data.

```
var joinfilter = ee.Filter.equals({leftField:'id',rightField:'id'});  
var join = ee.Join.simple().apply({  
  primary:RSLE,  
  secondary:threshold5,  
  condition: joinfilter  
});
```

Snippet 8, combining the RSLE collection with the cloud threshold. A join function only saves the primary data which are also present in the secondary collection

3.4 Data Processing and Trend Analysis

To know if there is a trend in yearly SCD in the Caucasus, a trend analysis method must be carried out. Unfortunately, the data downloaded from GEE is incomplete and needs to be processed first. Links to these codes can be found in Appendix A, Code Used.

3.4.1 Pre-processing

To be able to do a trend analysis on the CSV downloaded from GEE, it needs to be pre-processed. First thing is to make sure that each variable is in its own column without any additional signs. The date is taken out of the id column and saved in a separate column and the label is split into a longitude and latitude column.

Besides the RSLE collection, the grid collection with minimum and maximum elevation is downloaded as well, which is an extra check. It has the grid cell ids with the corresponding latitude, longitude, maximum and minimum elevation. For this collection, the same pre-processing steps apply.

After making sure that the collection has no other signs apart from numbers, the collection is filtered on the following conditions:

- * Minimum elevation must be above 0 m. Some of the research area lies below 0 m in the elevation model. It was expected that these negative values could cause problems due to their ‘-’ sign. To prevent this these locations were discarded.
- * The difference between minimum and maximum elevation must be greater than 250 m. This is done to make sure that at least two elevation classes are present. If only one class is present, it might be marked as constantly covered in snow, even though that cannot be the case in lower elevations.
- * The RSLE difference between two days may not be larger than 80% of the local elevation range. There were some misidentified RSLE values, due to a filter in the GEE code. These values resulted in a fully snow-covered grid cell in mid-summer, which was not physically possible. To remove these spikes, but keep the spiky character of the RSLE, this filter was implemented instead of for instance a rolling mean.

After the filters the RSLE collection was scanned for missing dates, and the missing dates were interpolated per location with a maximum of 8 days.

3.4.2 Calculating Snow Cover Duration

The SCD is calculated for each grid cell, for each elevation step, for each hydrological year (October 1st to September 30th). This was done with several loops (Snippet 9). Each elevation step, for instance 200 meter, is compared with the RSLE at a certain date. If the elevation step is larger than the RSLE then it is counted as a snow-covered day, if it is below the maximum elevation in that area. All snow-covered days in a hydrological year are summed up and the result is snow covered days per hydrological year. The elevation steps which are below the minimum elevation or above the maximum elevation in that grid cell are discarded for that grid cell.

Estimating snow cover decline using the RSLE method in Google Earth Engine

```

def analysesA(df,elevclass,start_date,end_date):
    names = [str(i) for i in elevclass]
    names.insert(0,'id')
    names.insert(0,'year')
    df_grid = df.loc[(df['date'] == '2001-10-29')]#pick a date for which
all id's are represented
    id_list = df_grid['id'].to_numpy()#list with all id numbers
    dfreturn = pd.DataFrame(columns=names) #empty dataframe
    temp2 = [] #to append on
    z=0 #for percentage counter
    perc=0 #for percentage counter
    for i in range(len(start_date)): #loop to create yearly data
        df2 = df.loc[(df['date'] >= start_date[i]) & (df['date'] <=
end_date[i])]
        dtlist = pd.date_range(start_date[i],end_date[i])
        empty3=[]
        for j in range(len(id_list)): #loop over every id
            df_id = df2.loc[(df2['id']==id_list[j])]
            temp1=[]
            empty2=[]
            for h in range(len(df_id['date'])):#loop through daily data
                df_date = df_id.loc[(df_id['date']== dtlist[h])]
                empty1 = np.zeros(len(elevclass))
                for k in range(len(elevclass)): #loop elevation classes
                    x = 0
                    rsle = df_date.iloc[0,4]
                    maxi = df_date.iloc[0,3]
                    mini = df_date.iloc[0,2]
                    if elevclass[k]<mini: # no wrong minima recorded
                        x = np.nan
                    elif elevclass[k] > maxi: #same for maxima
                        x = np.nan
                    else:
                        if elevclass[k]>rsle: #snow
                            x=1
                        else: #elevation is lower than RSLE, no snow
                            x=0
                    empty1[k]=x #fill empty list with value
                temp1.append((empty1))
            #create 1 list for all elevation classes
            empty2 = np.vstack(temp1) #stack elevation values, per day
            ar = np.sum(empty2,axis=0) #sum them up
            ar = np.insert(ar,0,id_list[j]) #add id column
            ar = np.insert(ar,0,i+1) #add year number (1 = 2001-2002)
            ar = ar.tolist() #make the np.array into a list, because
appending is easier
            empty3.append(ar) #append all lists to empty 3
            perc = round((j+1)/len(id_list)*100/(len(start_date))+z,2)
#percentage counter to so progress
            print(perc,'%')
            z=perc #part of percentage counter
            temp2 = np.vstack(empty3) #create a np array by vstacking all
lists in empty 3
            df_a = pd.DataFrame(temp2,columns=names) #make a dataframe from
np array
            dfreturn = dfreturn.append(df_a) #append for each year the
dataframes
            print('yearloop',i)#loop numbers for checking
    return dfreturn

```

Snippet 9,function in python to calculate SCD per year per location. It incorporates four nested for-loops (marked in cyan) and in the last for-loop a nested if-else loop.

3.4.3 Mann-Kendall test

The trend analysis method used is the Mann-Kendall test (Kendall, 1975; Mann, 1945). It is chosen because the test is used in many papers on snow cover trends, and other climate related fields. The test is non-parametric which means that the data does not have to be normally distributed. Seasonality can affect the test, but since the yearly value for the SCD is tested, it does not have a seasonal pattern. The test is implemented via the *pymannkendall* package of python. A trend is said to be significant if the p-value is smaller than 0.05. The p-value is a value linked to the probability density function of the observed trends. A small value signifies that the trend is not likely to occur at random and is thus an actual significant trend.

4 Results

4.1 Google Earth Engine Result

The goal to develop a more efficient code was only partly fulfilled using GEE. Although a great data reduction was achieved, the computation took rather long. This is mainly due to the nested function (Snippet 6), but also due to the threshold comparison. When the entire timespan (18 years) would be computed at once, the code would run and run, and time out on day 12. The time out is a safety measure from GEE to prevent large tasks to occupy servers for a long time. If a large task is executed a smaller number of servers will be assigned to that task. The time needed to do calculations does not increase linearly for larger tasks. Instead a large dataset may run for three or four times as long, compared to two half size datasets. Therefore, the collection was split up. The computations were done for 0.5° by 0.5° (80x60 km²), and 0.25° by 0.25° (40x30 km²). The 0.5 collection was run for each year at a time and the 0.25 collection was run for 1/3 of a year (though 3 export tasks ran at the same time). Running a year of the 0.5 resolution took 5 hours, resulting in a total of 85 hours. A year of the 0.25 resolution (three exports at the same time) took 10 hours (170 hours in total)

The developed method however is still seen as more efficient since a great data reduction has been achieved. The 0.5 resolution has a total volume of 41.6 MB after the download and the 0.25 has a total volume of 168 MB. This is a reduction in the order of a factor 1000. The downloaded CSVs are easier to process than satellite images since CSV's only require basic programming skills. The total runtime of the SCD calculation and the trend test calculations is about 6 hours.

The used method in GEE is not the most efficient method, but it produced results. A faster code has been developed, however since the download was not in the form of a CSV but in the form of an Image Collection, which cannot be downloaded, this code was not taken into consideration for the analyses. Even though it is not used for the analyses, it still is a more efficient method. More on this can be read in 5.3.3 *Improved code and its issues*.

The downloaded CSV can be seen in Snippet 10. It consists of an id number, which consists of the location id, followed by the date (yyyy_mm_dd), followed by the minimum and maximum elevations, then the RSLE, and finally the label, which is the longitude and latitude.

```
id,min,max,RSLE,coordinates
2-2011_10_01,944,2159,2000,"[37.5, 38.5]"
3-2011_10_01,1180,1966,1966,"[37.5, 38.75]"
27-2011_10_01,12,525,525,"[37.5, 44.75]"
28-2011_10_01,-3,287,287,"[37.5, 45]"
36-2011_10_01,936,2677,2400,"[37.75, 38]"
39-2011_10_01,991,2223,2200,"[37.75, 38.75]"
40-2011_10_01,1279,2682,2600,"[37.75, 39]"
63-2011_10_01,1,640,640,"[37.75, 44.75]"
64-2011_10_01,-1,207,200,"[37.75, 45]"
```

Snippet 10, a part of one of the CSVs exported from GEE

4.2 Analyses results

4.2.1 RSLE development

Examples of the progression of the RSLE are plotted for one grid cell of the 0.5° resolution, and the four corresponding 0.25° resolution grid cells at the same location (Figure 15). These locations are in the Georgian part of the Greater Caucasus. The most obvious feature is that the snowlines have a lot of spikes. Even though some spikes are filtered not all spikes are removed. Since the analyses are done over large cells, the snowline can move several 100 meters in one night and move back during the day. This happens when there is a lot of snowfall during the night, causing a light dusting of snow over most of the area, which causes the snowline to drop. If during the day the temperature reaches above 0 temperatures with a strong sunshine, all that snow may melt again. Due to this behaviour the choice was made not to smoothen out the signal with for instance a rolling mean.

Looking at Figure 15, it becomes clear that the 0.5 resolution behaves a lot differently then the four 0.25 snowlines. Two of the 0.25° resolution RSLE lines have many spikes (green and blue), and the others a bit less (red and yellow). The 0.50° resolution however is spike in winter but does not show a lot of spikes in summer. meaning that tin the higher elevations, a larger resolution does not capture local rapid snowmelt.

Most peaks are larger than one elevation step, and thus more than 100 meters. Therefore, it is assumed that these spikes are not a result of the large elevation steps. The interpolation might even have caused a local smoothening of the snowline. Due to the form of the snowline it was chosen to leave start and end date of snow seasons out of this analysis. Instead just snow cover duration and its trends were calculated at different elevations.

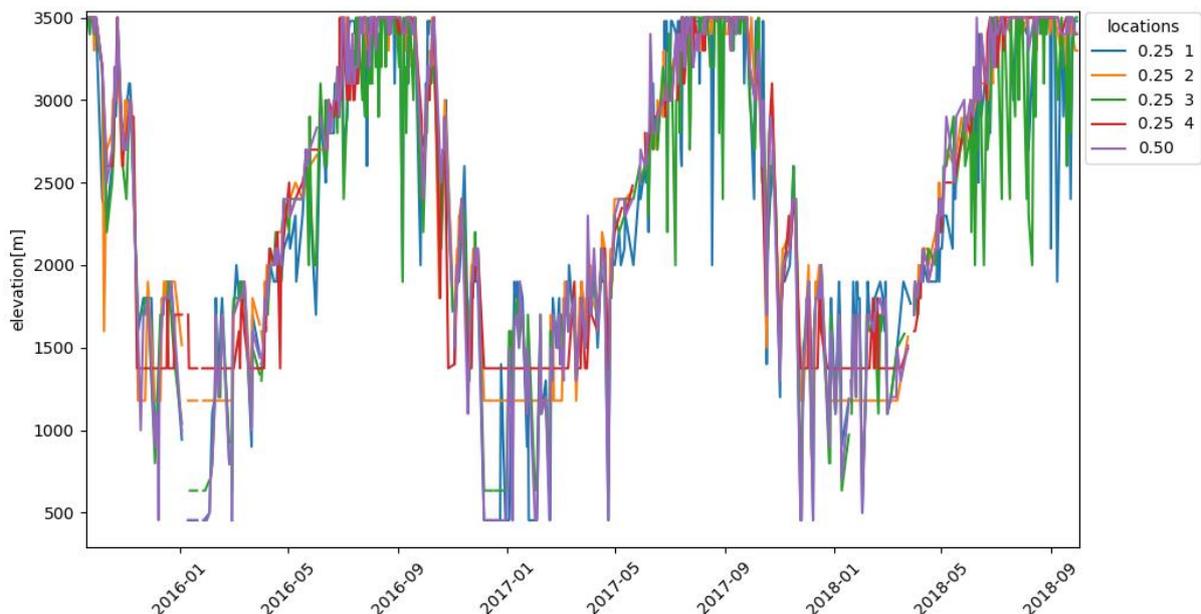


Figure 15, RSLE plotted over three years for 1 grid cell of the 0.50° resolution, and the four 0.25° resolution grid cells that are in the same location

4.2.2 Snow Cover Duration

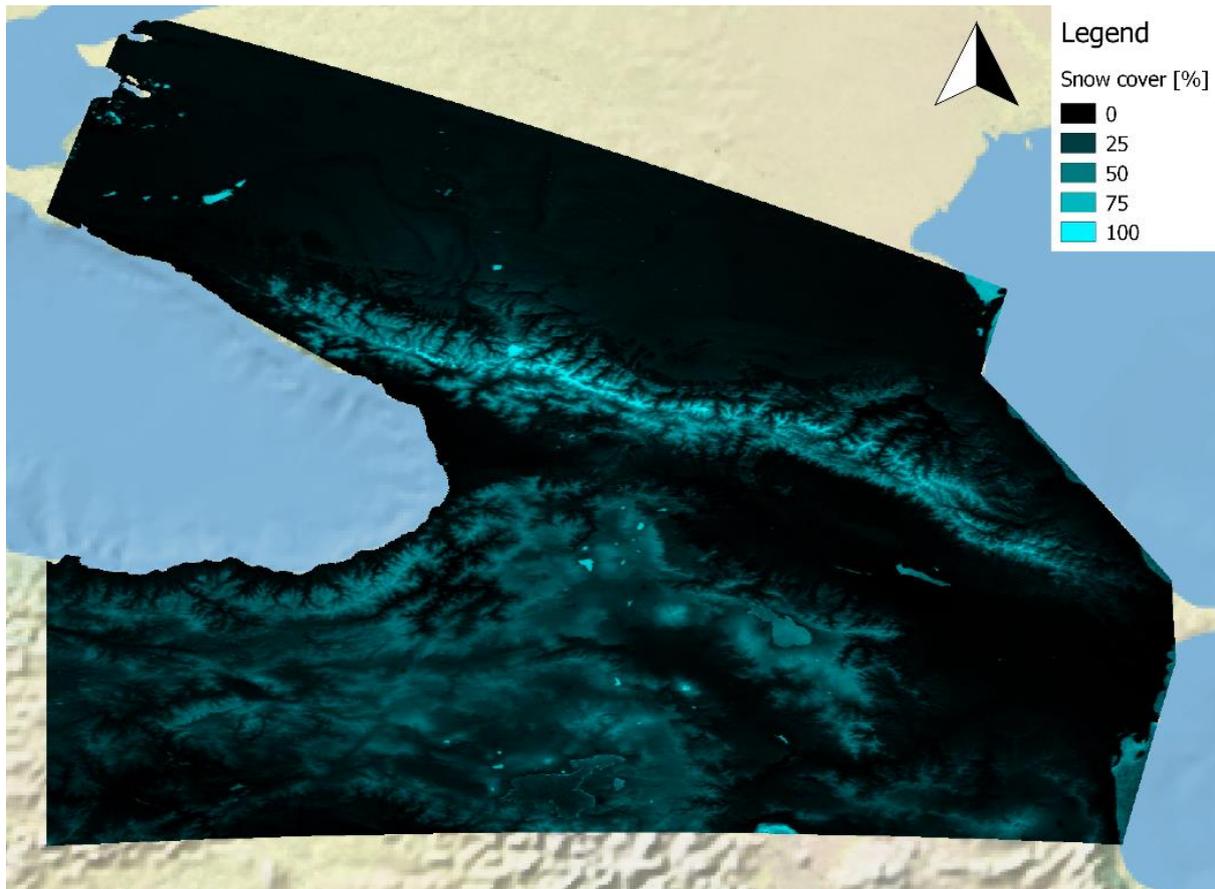


Figure 16, total percentage of time covered in snow from 1-10-2001 until 30-9-2018

For SCD it is possible to look at the percentage of snow cover fallen over the last 20 years, at the pixel resolution, which is 500 by 500 meter (Figure 16). However, because all cloudy days are not taken into account, a lot of data is missing. It is however easy to see where some glaciers are located and where permanent snow cover is present.

As said in the method, in order to calculate trends, the SCD is calculated per hydrological year. In the following plots these years are assigned a number. This number is dependent on the year of the start date. For instance, the hydrological year which starts on the 1st of October 2001 and ends on the 30th of September 2002 is referred to as year 1. To analyse the overall differences between elevations, all the SCD values for all years and all locations have been accumulated into one boxplot (Figure 17, Figure 18). As expected, the snowline increases with increasing elevation. For the 0.25 resolution, there are datapoints where the yearly SCD is zero days up to 2800 meters.

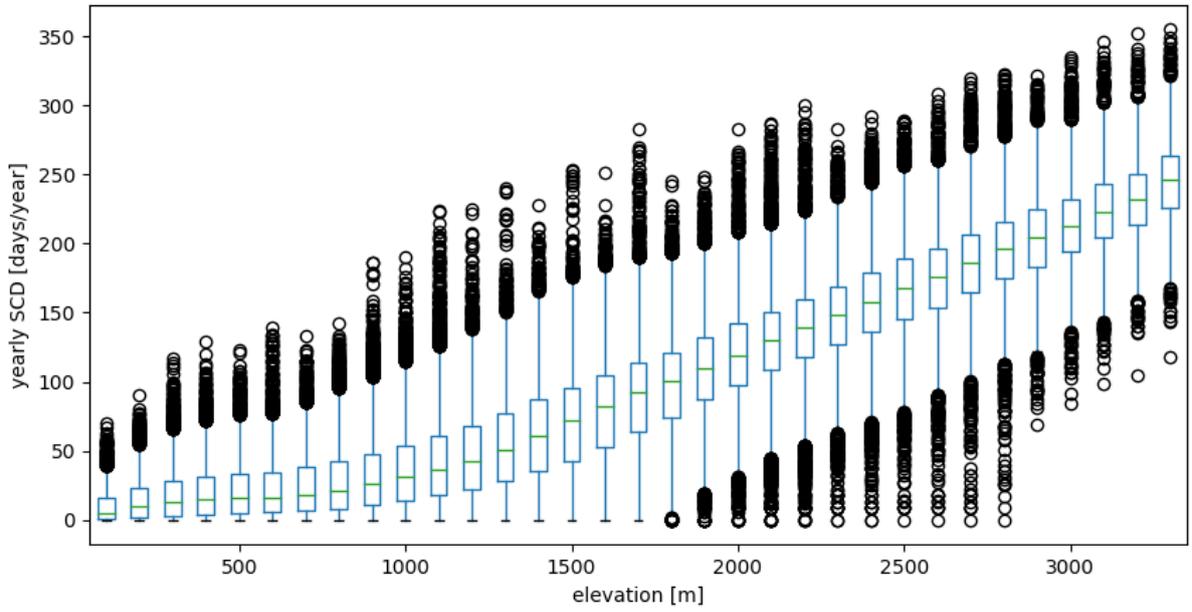


Figure 17, boxplot of the yearly SCD, only sorted by elevation for the 0.25° resolution

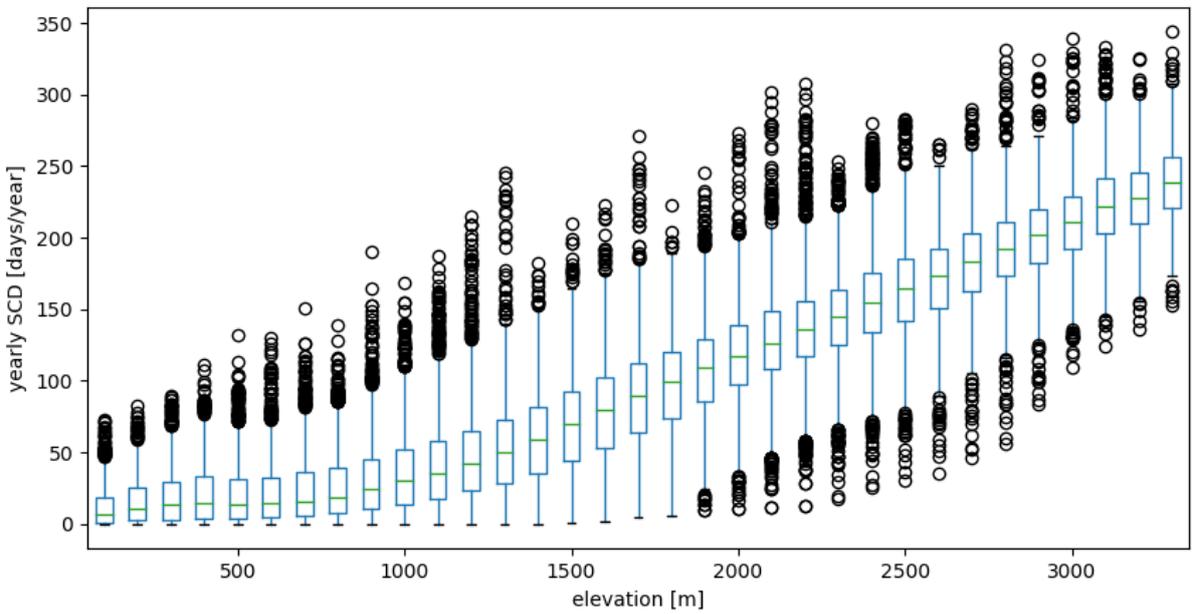


Figure 18, boxplot of the yearly SCD, only sorted by elevation for the 0.50° resolution

To see differences in years, all the SCD values for all locations and all elevations have been accumulated into a different boxplot (Figure 19). The median of all the data is higher before 2009 and lower after 2009, with the clear exception of 2011, when most SCD values are clearly higher. The years 2009 and 2017 have the lowest SCD. The 95 percentile is the lowest in 2002, 2010, 2012 and 2013.

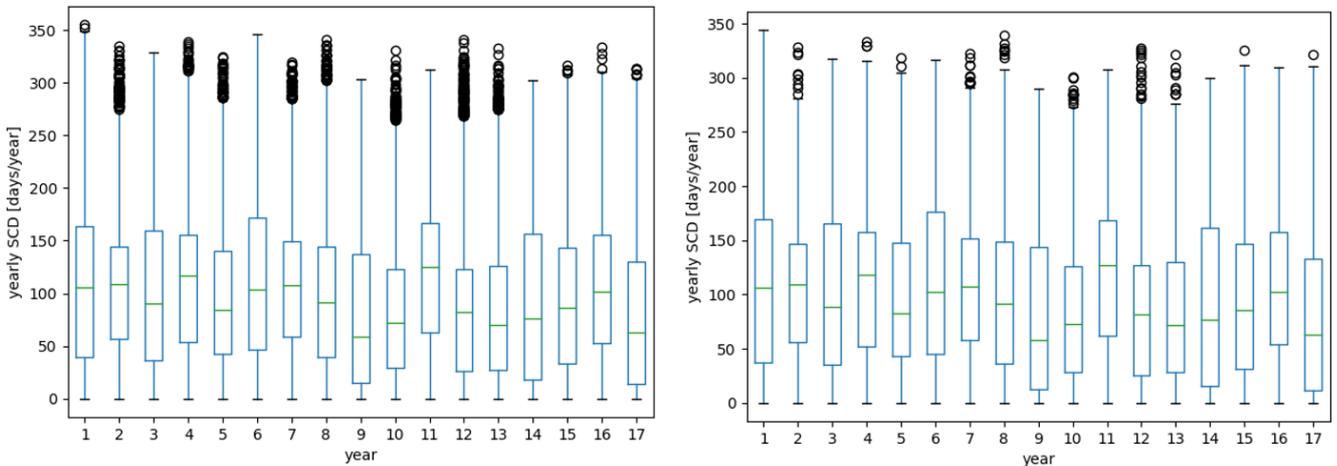


Figure 19, boxplots of the yearly SCD, only sorted by year. Left is the 0.25° resolution and right the 0.50° resolution

To get a better idea what is going on in these years a colour plot has been made for the mean SCD on different elevations in different years (Figure 20). The years 2009 and 2017 stand out since they have lower SCD values up to 2000 meters, after which they become comparable to the rest. Except for elevations above 3000 meters, 2011 has the highest SCD values over all elevations. The low maxima (excluding outliers) in 2002, 2010, 2012 and 2013 is clearly due to lower SCD values in the higher altitudes.

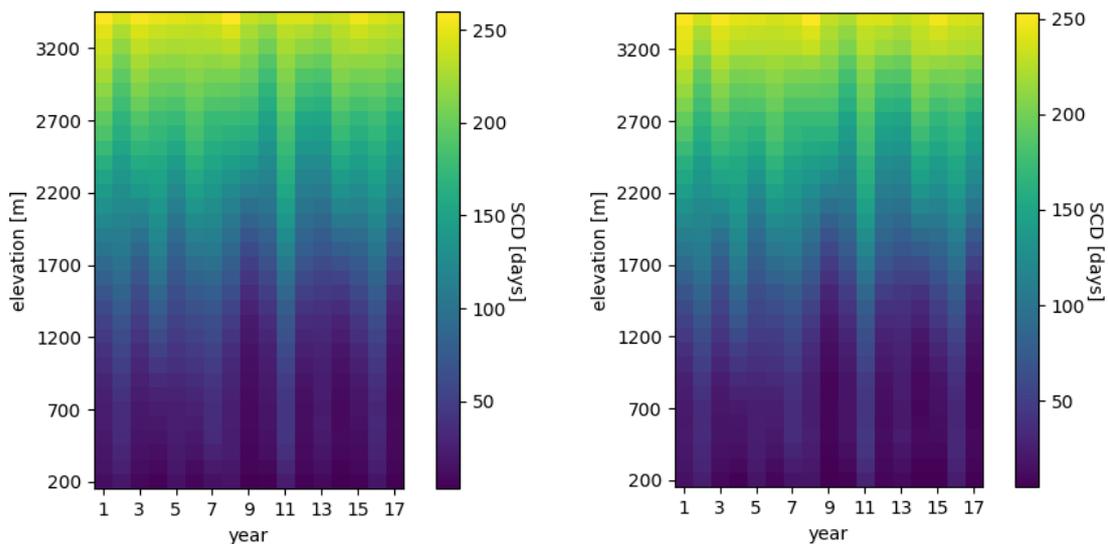


Figure 20, colour plots of all the mean snow cover duration at different elevation and different years. Left is the 0.25° resolution, right is the 0.50° resolution

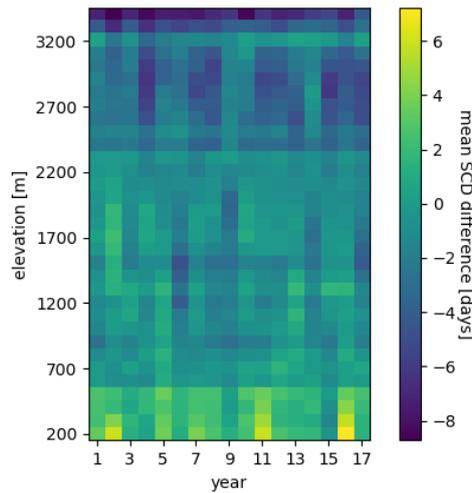


Figure 21, difference in mean snow cover duration plots. The 0.25° resolution was subtracted from the 0.50° resolution.

4.2.3 Difference between 0.5° and 0.25° resolution

When looking at the images in 4.2.2 it seems that there is not much difference between 0.5° and 0.25° resolution datasets. The colour plots especially show almost no difference in SCD results. However, when subtracting the 0.25° from the 0.50° image (Figure 21), it becomes clear that there is a difference. In the lower altitudes the 0.50° dataset has a higher mean SCD, whereas the higher altitudes have a lower SCD.

Looking at the boxplots per elevation (Figure 17, Figure 18) the only major difference is the few locations and years where there is no snow. In the 0.25° resolutions these points can be seen up to 2800 meters, however in the 0.5° resolution the minimum SCD is already larger than 0 at 1600 meters. This is expected because the 0.5° resolution filters out the topographical differences more. The chance that a 0.5° cell consists entirely of a dry area is much lower than a 0.25° cell.

Looking at the yearly boxplots (Figure 19), the only major difference is the upper limit of year 6 (2006-2007). In the 0.25° boxplot the upper limit is 20-30 days higher. Even for the total amounts of trends (Figure 22), both the 0.50° and the 0.25° have both around 17% of statistically significant decreasing trends, though the shape of the histograms are a bit different. The 0.25° resolution is deemed better, keeping in mind that the RSLE of the 0.50° resolution seemed to smoothen the snowline in the summer, and that it shows more detail and local trends.

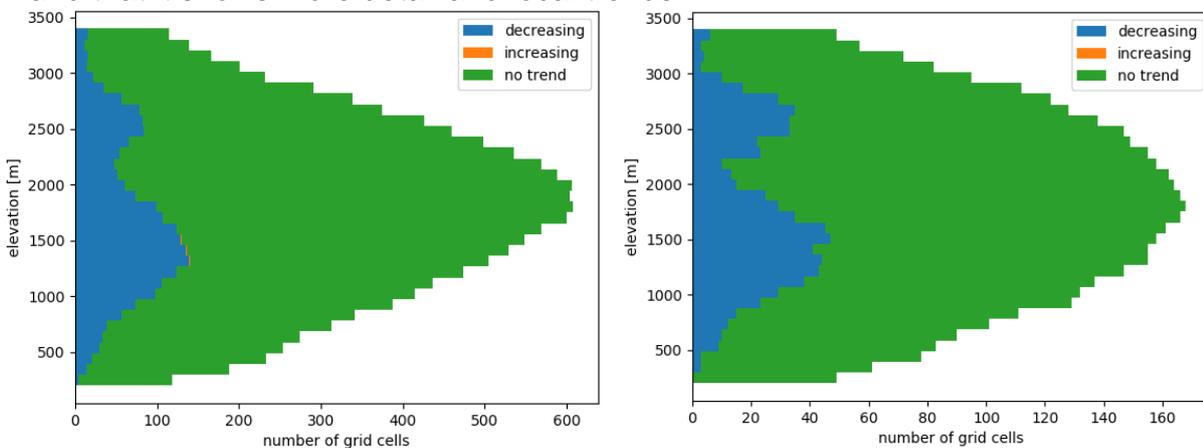


Figure 22, stacked histograms for the number of grid cells that have, or do not have a SCD trends at different elevations. Left is 0.25° and right is 0.50° resolution.

4.2.4 Overall trends'

The results of the Mann-Kendall test show if trends are significant and if they are increasing or decreasing. The trends themselves are expressed in a slope value which is change in SCD/year. The significance is expressed in a p-value. P-values lower than 0.05 are considered good.

Looking at Figure 22 one can see that not many trends are significant. Both the 0.5° and 0.25° resolutions have around 17% of decreasing trends, and the rest is not significant. The 0.25° resolution has three increasing trends (which is 0.02% of the total), all in the same location at 1300, 1400 and 1500 meters. The greatest number of significant trends are observed around 1500 and 2500 meters which can also be seen in Figure 23. For trend analysis on the map, only the 0.25° resolution has been considered since Figure 23 shows that, although not much difference could be observed in the overall results, the 0.25° resolution is more detailed and therefore only used to portray the trends at different locations and elevations. Two examples of this can be seen in Figure 24 and Figure 25, which portray the results of the 1500- and 2500-meter elevations. For all other elevations, see *Appendix B, Trend Results for all Elevations*.

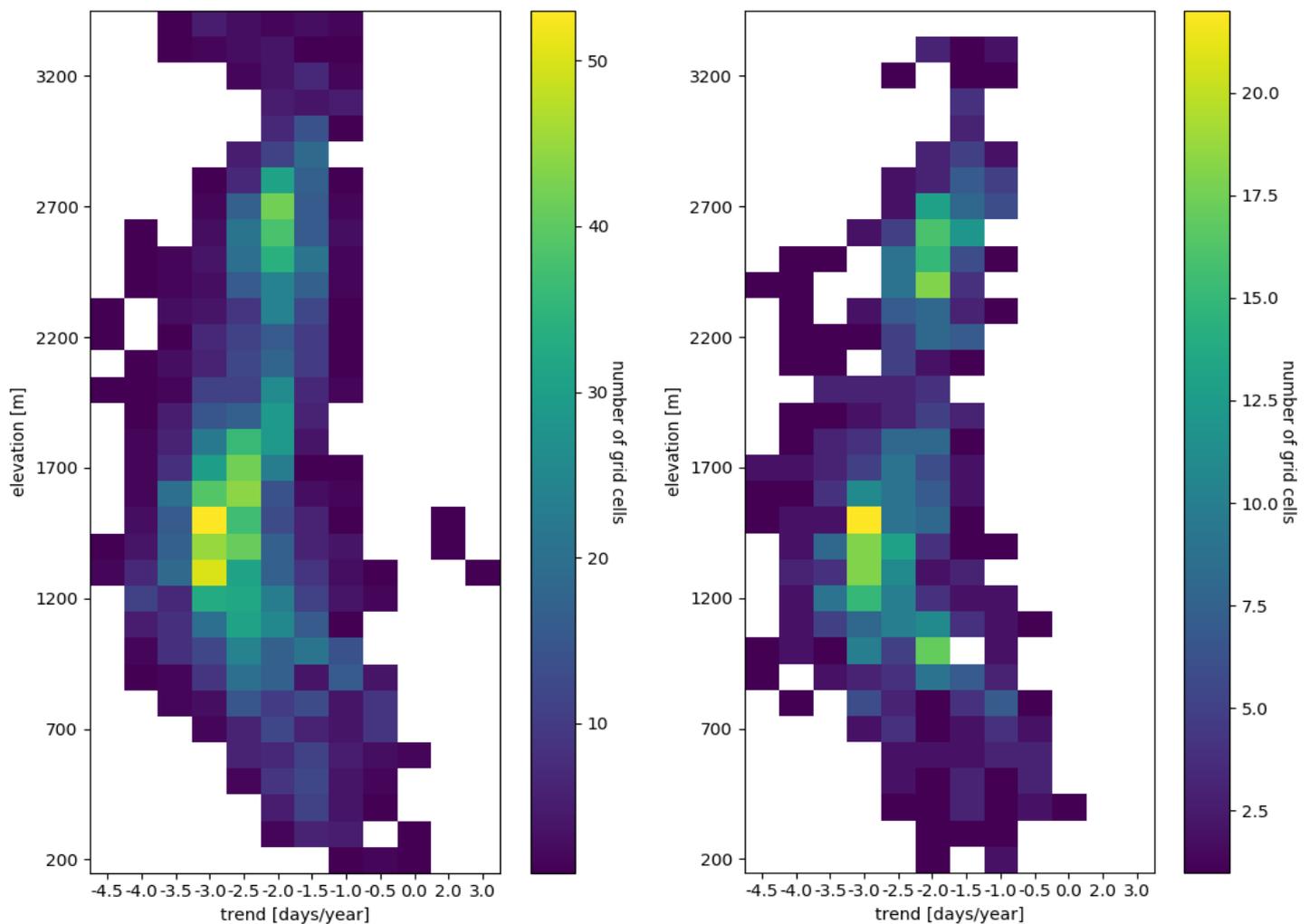


Figure 23, colour plots for the number of different trends at different elevations. The left figure are the trends for the 0.25° resolution and the right figure for the 0.50° resolution

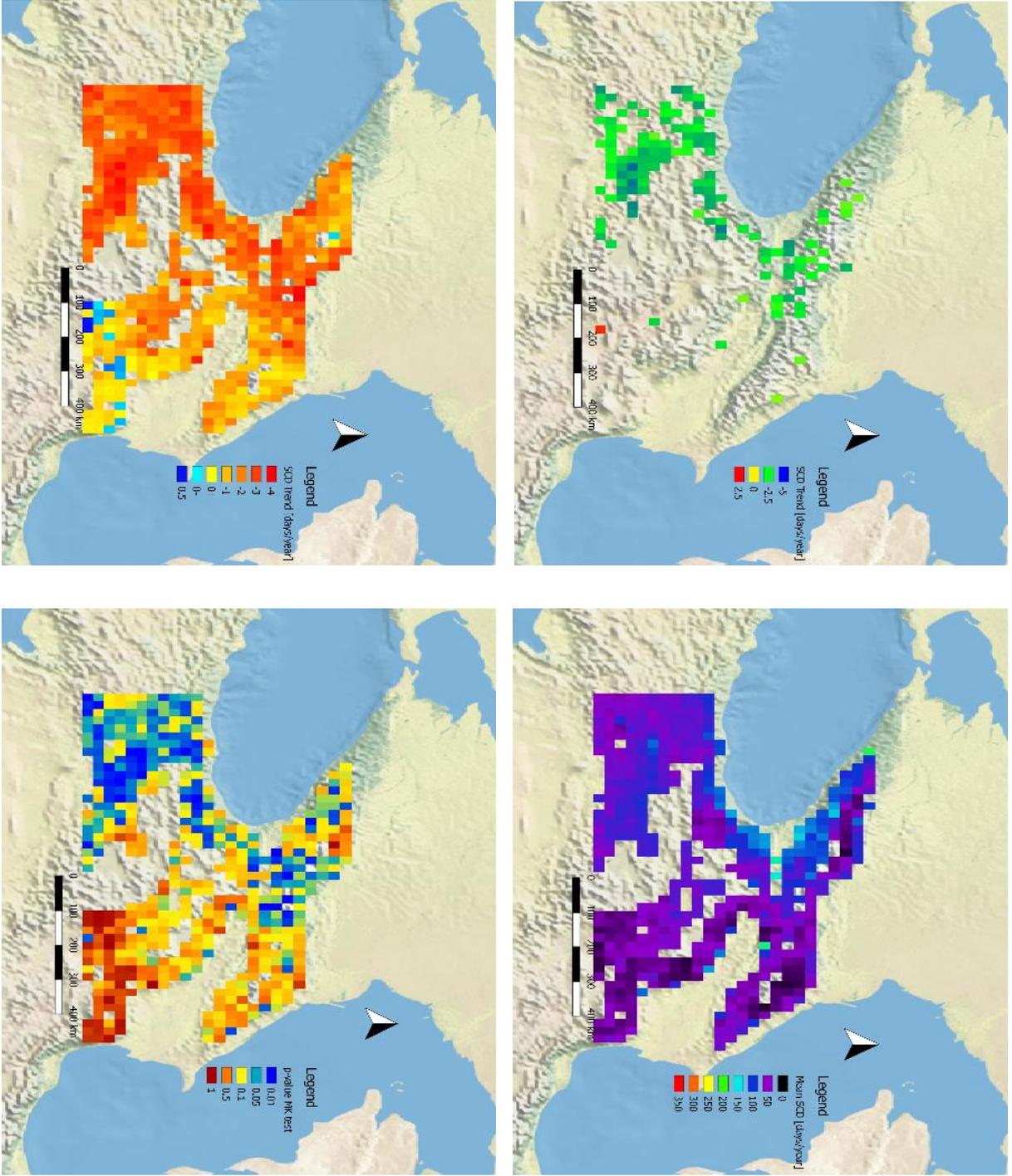


Figure 24. SCD trends at 1500 meters for the 0.25° resolution

Top-Left: significant trends.

Top-Right: mean SCD.

Bottom-Left: All trends, including not significant trends.

Bottom-Right: p-values

Estimating snow cover decline using the RSLE method in Google Earth Engine

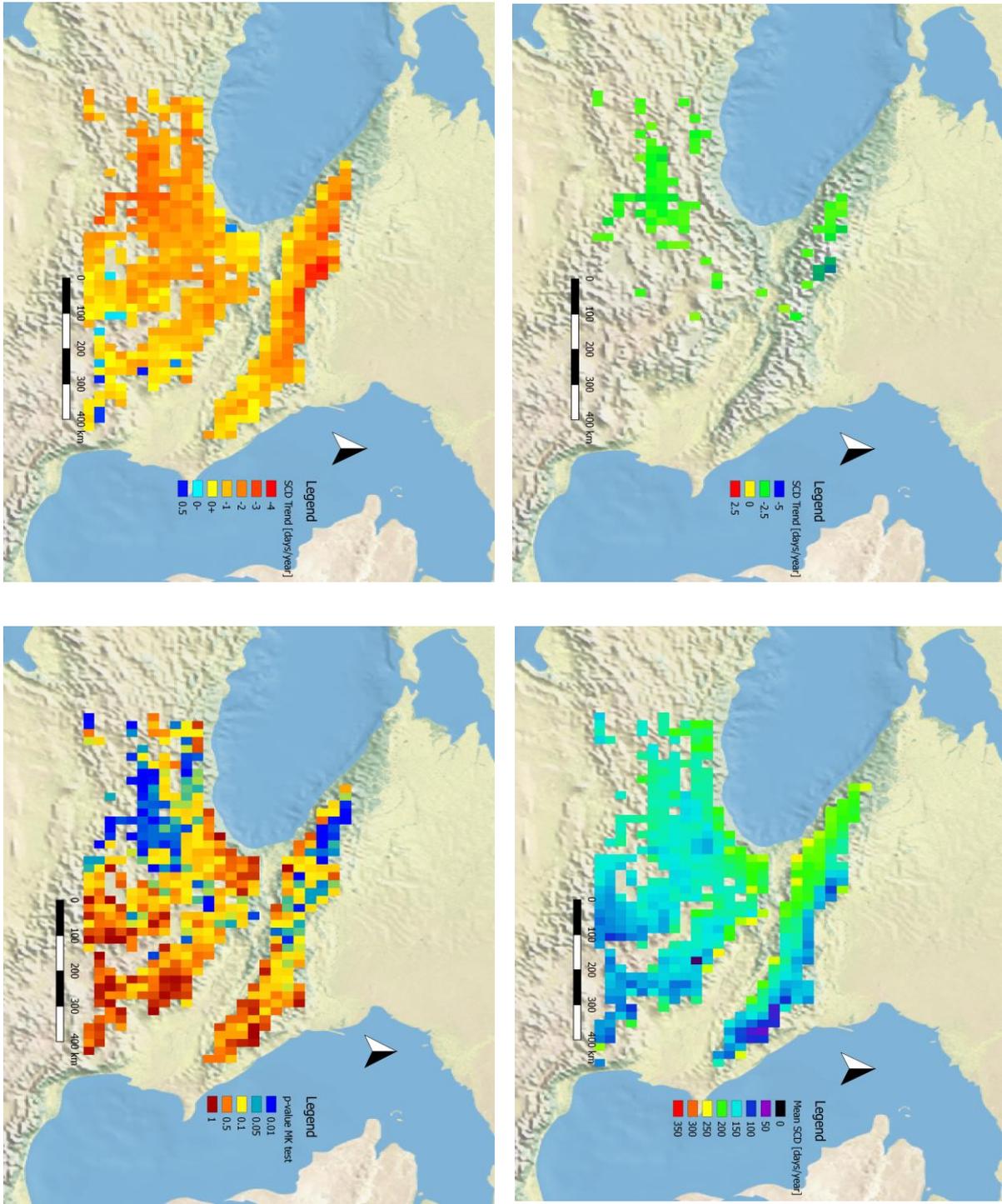


Figure 25, SCD trends at 2500 meters for the 0.25° resolution

Top-Left: significant trends.

Top-Right: mean SCD.

Bottom-Left: All trends, including not significant trends.

Bottom-Right: p-values

4.2.5 Local trends

From the results in the previous section and *Appendix B, Trend Results for all Elevations*, it seems that the south west has more trends than other areas. To check if this is true the data was split up over four areas, as can be seen in Figure 26.

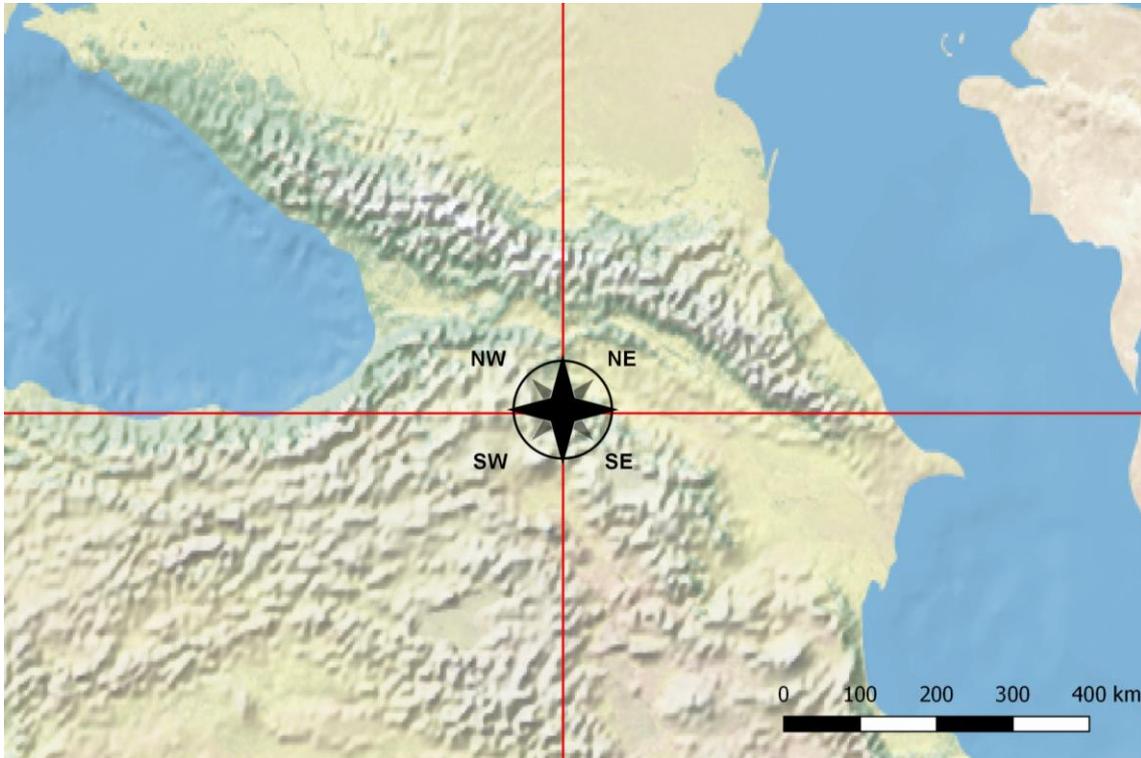


Figure 26, division to estimate local trends

When looking at the SCD boxplots for the four different areas (Figure 27), it becomes clear that the east has lower SCD values than the west, which is expected since the East has a dryer climate. The north has higher maximum SCD values, which is due to higher mountains in the Greater Caucasus.

Looking at the significance of the trends in the different regions (Figure 28), it is very clear that the south west has the most trends, especially up to 1700 meters. The elevations above 3000 meters however, do not have more significant trends. All the other areas do not have such low p-values at the low elevations.

All the significant trends plotted together (Figure 29) show that the east has little trends compared to the west. The characteristics of the trends are roughly the same in all the regions. The most decreasing trends of -4 days of SCD/year or less are found in the mid-high elevations.

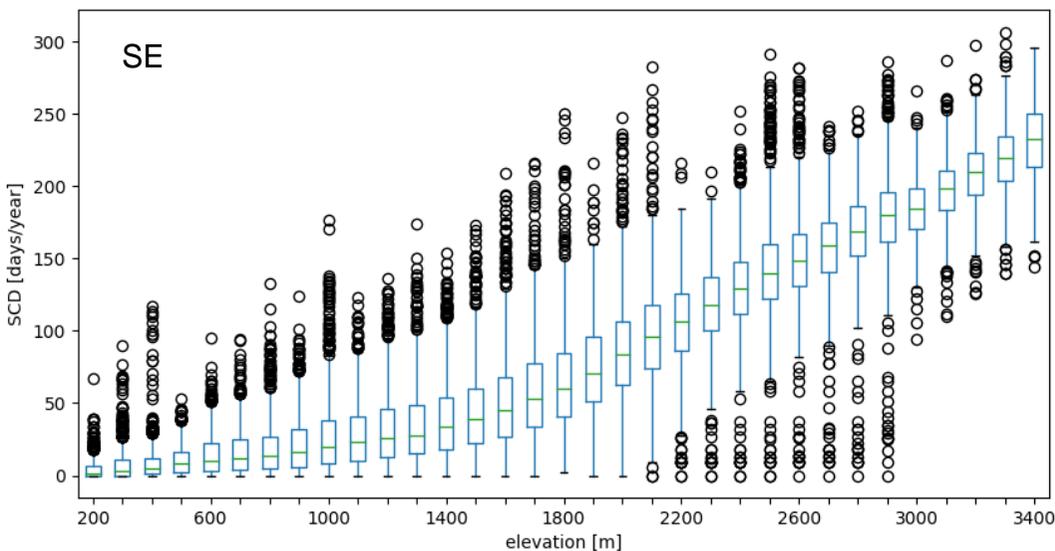
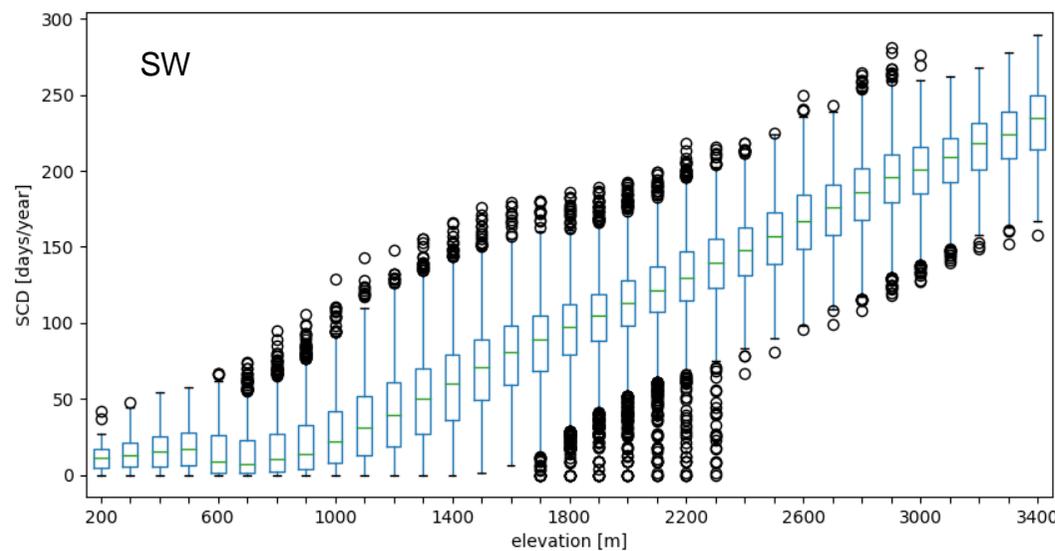
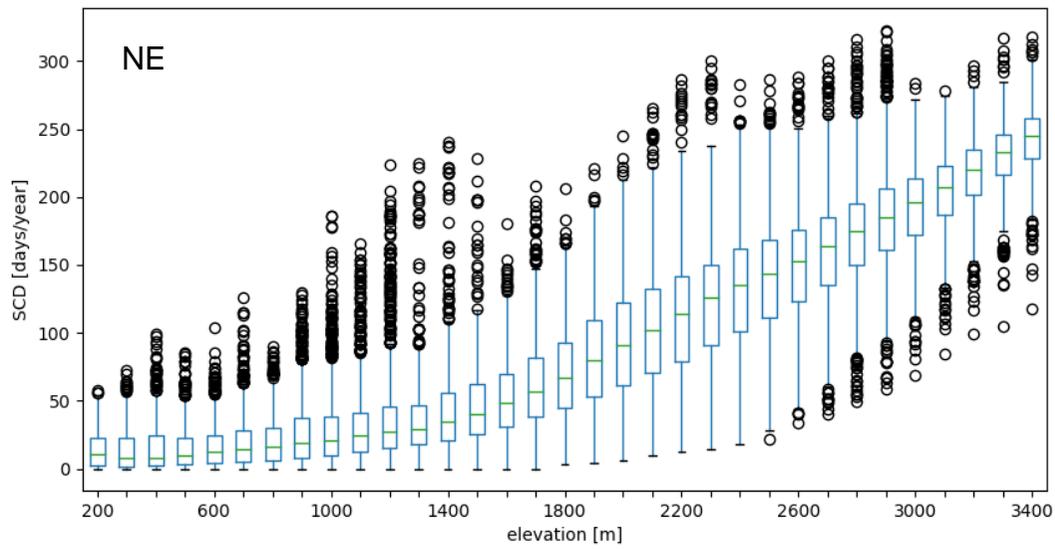
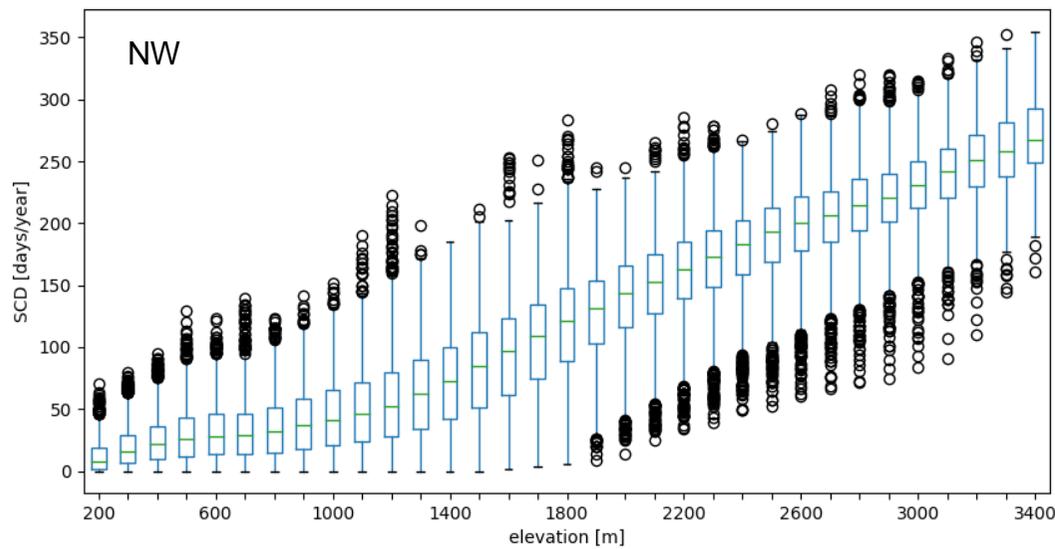


Figure 27, boxplots of snow cover duration for different elevations for the different regions

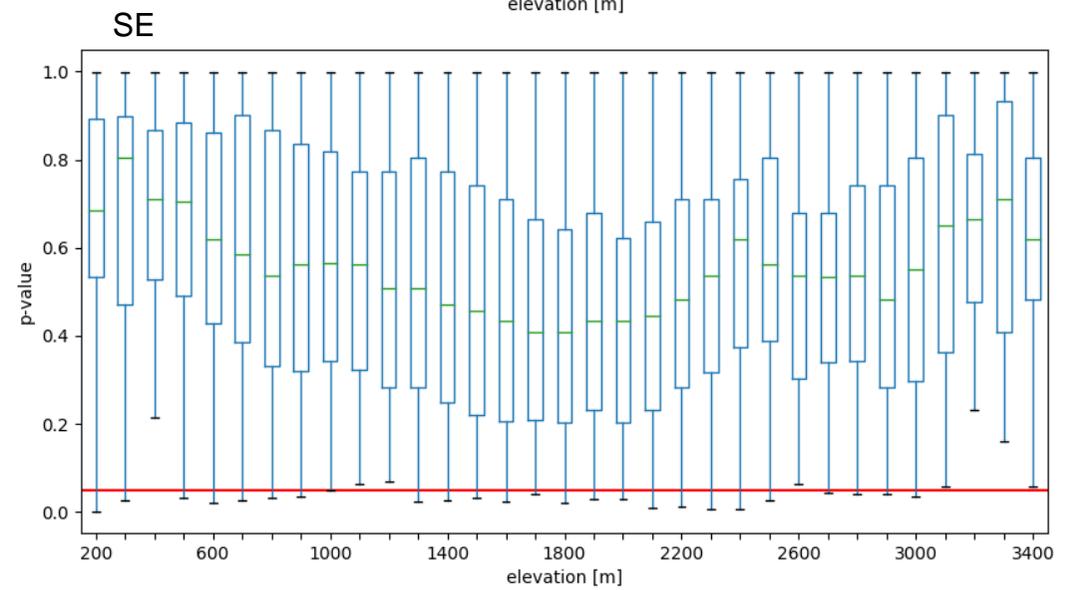
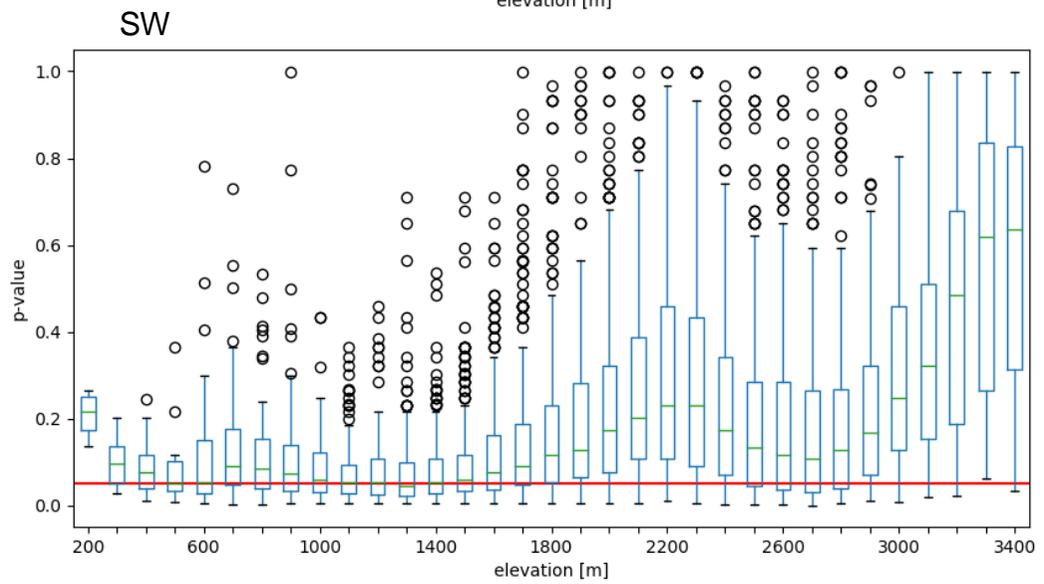
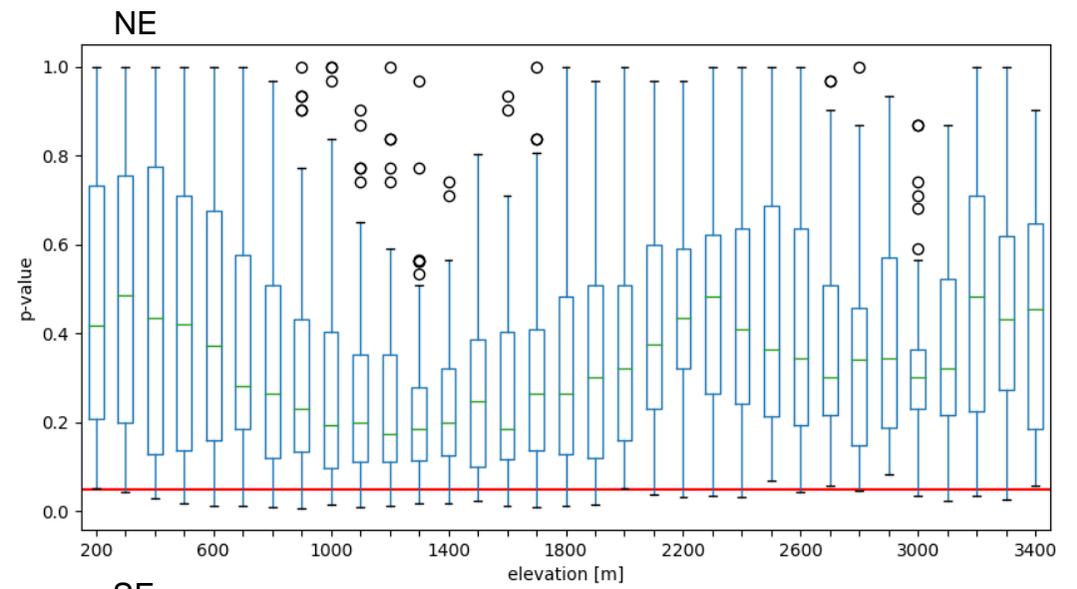
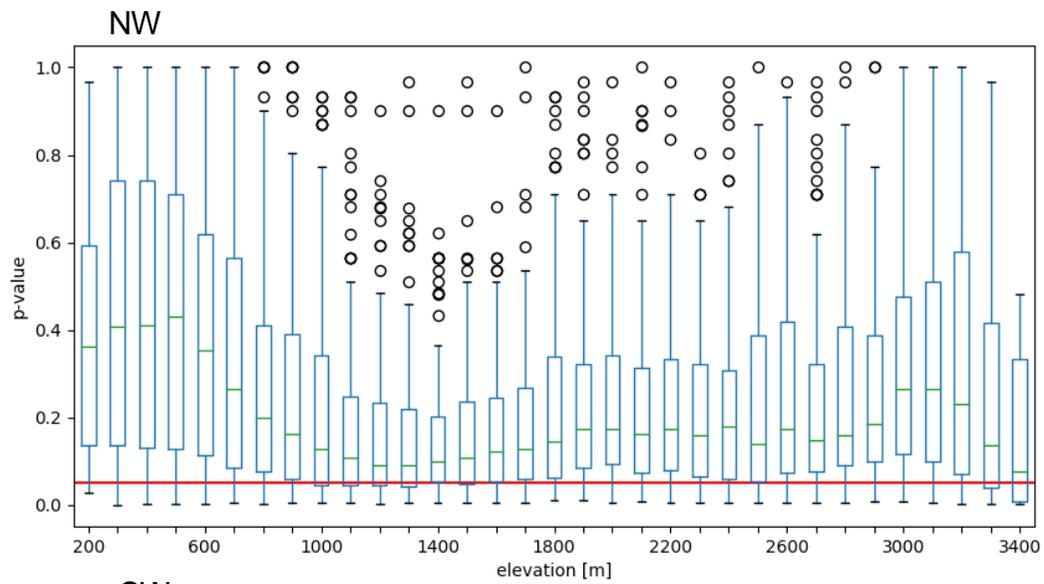


Figure 28, Mann-Kendall p-values per region per elevation. The red line is plotted on 0.05. All trends below that line are deemed significant.

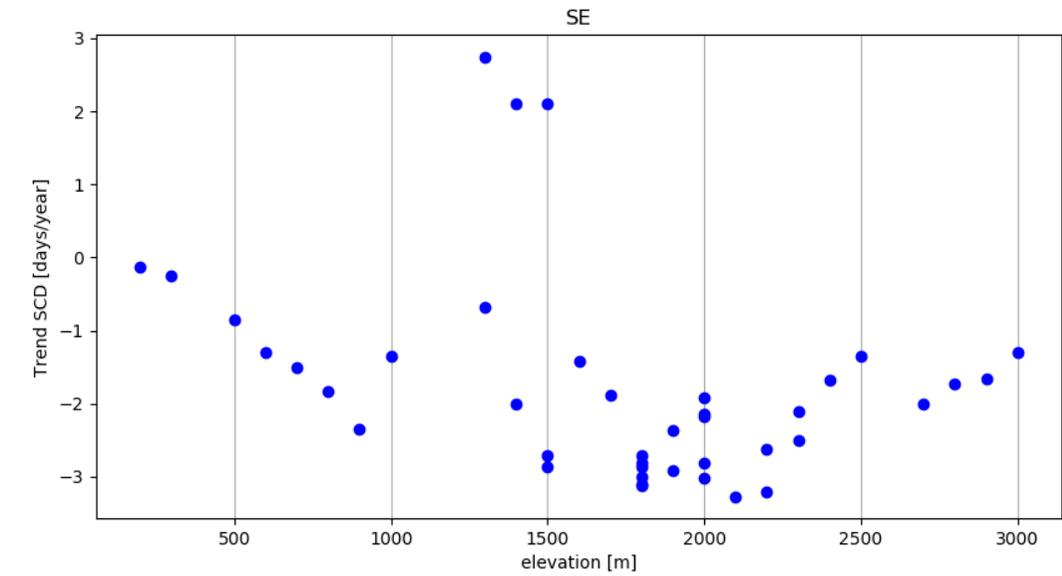
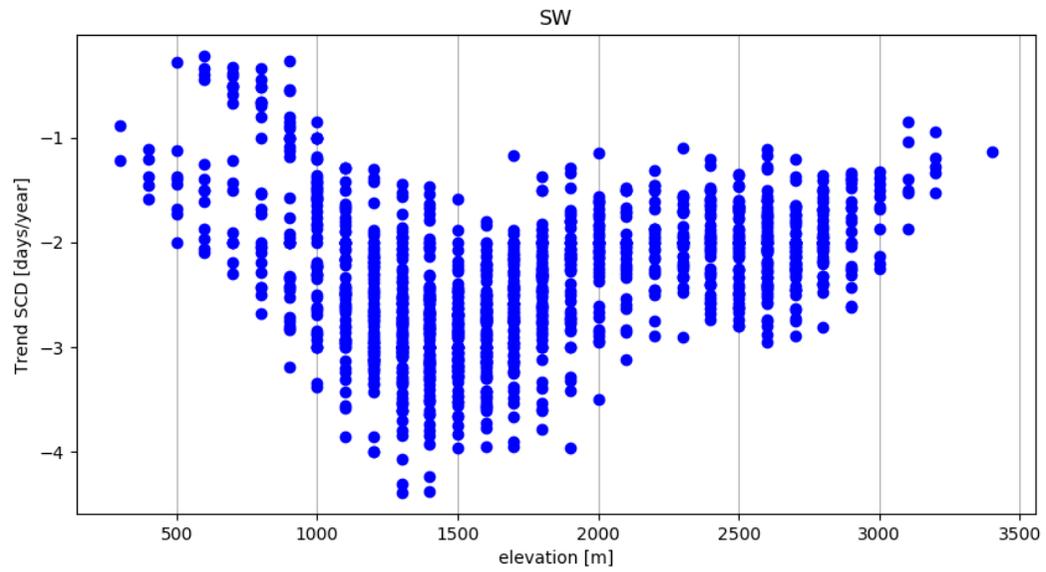
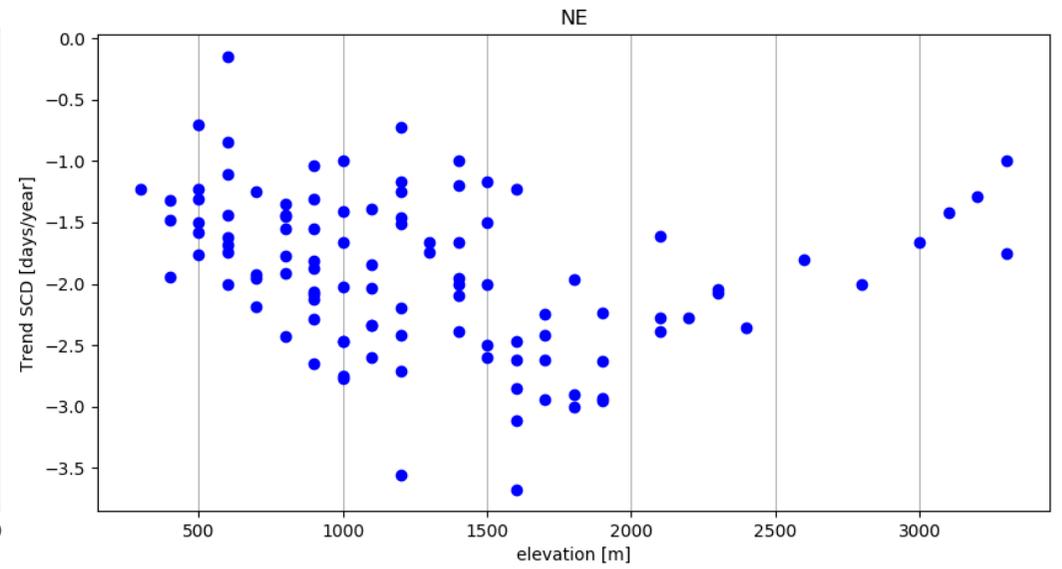
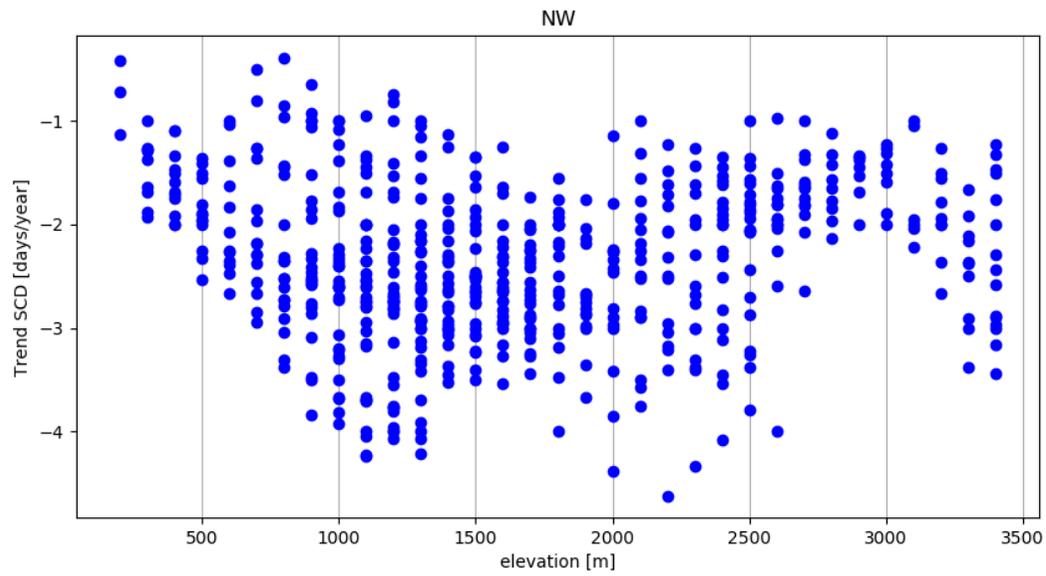


Figure 29, significant SCD trends per area, thus the trends that are below the red line in figure 28.

5 Discussion

5.1 Effect of declining SCD trends

The declining trends in snow cover duration will have large effects on the water supply for the entire region. The Euphrates-Tigris basin is highly dependent on snow cover, with about 90% of the flow of the Euphrates, and about 40% of the Tigris originating from eastern Turkey (Özdoğan, 2011). These mountains, in combination with the Zagros mountains have been called a water tower, meaning that these mountainous areas are almost entirely responsible for the discharge (Viviroli et al., 2007).

A big part of these headwaters, excluding the Zagros mountains, lies in the south-west research area in Figure 26, where a small part of the Kura-Aras basin starts as well. The mean SCD decline over the entire south west research area is 2.3 days/year. Over 18 years, this comes to a bit more than a mean decline of 41 days in total. A decline in SCD can be caused by either a rise in temperature, a decrease in precipitation or a combination of both.

Yucel, Güventürk, & Lutfi (2015) showed that annual precipitation measured at on-ground stations in the east of Turkey showed no significant trends, but average temperature increased significantly. Although an increase in winter precipitation was measured, a depletion of the snowpack was measured as well. The increase in temperature causes precipitation to fall as rain rather than as snow. The temperature increase also might change the intensity of rain, causing floods to be more intense. It was noted that precipitation and snow cover measurements are measured locally and therefore one must be careful to draw conclusions for the entire area.

Using the findings in the results of this thesis, the depletion of the snowpack is observed over the entire area, and all elevations (although most significant trends are below 1500 m). Yucel, Güventürk, & Lutfi (2015) observed an earlier start of the melt season nowadays as compared to the 90's of about 1-2 months. This is similar to the 41 days of decline mentioned before, which would imply that the start of the accumulation season does not change. A more detailed study would be needed to be able to conclude this.

Yılmaz & Aalstad (2019) showed a decline in snow cover as well, although their trends are considerably lower than the ones found in this thesis. This might be due to the use of the daily MODIS in this thesis as compared to the 8-day average used in Yılmaz & Aalstad (2019). Besides snow cover duration measured with MODIS, Yılmaz & Aalstad (2019) conducted research using an array of other remote sensing sources, including measurements of precipitation, temperature, snow water equivalent. A grim outlook is sketched for the Euphrates and Tigris basin, as well as for the Kura-Araks basin, where the seasonal snowpack might be disappearing the coming decades. For future work it would be interesting to see the difference in trends computed with the 8-day MODIS average in this region or the daily MODIS improved with the RSLE.

For the north part of the Caucasus, no detailed studies exist that give an overall view of snow cover patterns. This can be explained by tensions in the region between countries and, from a hydrological point of view, the existence of a lot of smaller streams and rivers, which makes hydrology studies more local as compared to the Euphrates and Tigris basin. In the Russian part of the Caucasus, studies have been

conducted to glaciers, however not of snow cover patterns. It has been suggested that the extend of the current hydrometeorological data is not enough, and that remote sensing solutions are needed (Lur'e & Panov, 2011).

With a decreasing snowpack in the northern Caucasus, river flows change. Higher winter flows, an earlier and faster spring melt, a larger proportion of rain floods are among the things that can be expected (Rets et al., 2018).

In the near future, further decline of the snow season must be a point of focus for water management in this area. The decline of discharge and change in its patterns creates the need for close cooperation which is difficult in a region where not all countries have or maintain diplomatic relations. This does not mean that cooperation on a smaller scale is not possible, it might even be a way to reduce tension (Altingoz & Ali, 2019).

5.2 Analyses Discussion

Although the analyses show solid results, some remarks must be made. The way the dataset was filtered is not based on solid research and might influence the number of local trends. The runtime of the analysis was still rather large and although SCD has been calculated at different altitudes and locations, it is not linked to the actual area of that elevation.

5.2.1 Filtering the dataset

A first filter which was applied was in Google Earth Engine is the cloud cover threshold, which only allows one to calculate the RSLE if the cloud cover is no more then 70%. This 70% is based on a sensitivity analyses done by Krajčí et al. (2014). However, this threshold is not tested on different cell sizes and was only tested in combination with a minimum snow cover threshold of 5%, which was not implemented in this thesis. Although the cell size sensitivity has already been researched for the alps (Fugger, 2018), this does not have to be the same for all regions. Since cloud cover patterns differ per region, it would be good to have a general method added to the RSLE to establish the cell size and thresholds. A clearer understanding of the effect of the different thresholds, in combination with tile size and the region it is used on, is needed to make sure that cloud and snow thresholds are effective.

The dataset has been filtered on three conditions, as mentioned in the method. The first one was to prevent miscalculations due to negative numbers. It was implemented as a precaution for the DEM, of which is was unsure if the measured elevation were all above the sea level. This means a large part near the Caspian Sea has been excluded from the research. The same accounts for the second filter, the minimum of 250 meters difference to make sure at least two elevation steps are present. It was an arbitrary filter, just to make sure that the analysis would start without issues. If the elevation step height is decreased this filter should be removed as well.

The filter that removed all peaks larger than 0.8 times the maximum elevation difference is just to make sure that the largest error peaks are removed. Peaks that are not errors might be removed and peaks that occur in winter, which might be errors, are not removed. Although this factor is not based on literature, it was implemented after plotting the RSLE, and observing the error spikes. For future work, the statistical viability of different factors or the use of a rolling mean should be tested.

5.2.2 Mann-Kendall Test

The Mann-Kendall test is a test used often for climate research. However, the datasets it is used for are often larger than the dataset used in this thesis. In this thesis, only 17% of trends were significant, which is probably due to the short datasets (17 datapoints of SCD, for 18 years). Besides showing less trends for short datasets, the trends shown might be a misrepresentation of trends due to an unforeseen change. For example, the trend observed in the last 20 years may not be a trend which has been happening over the last 50 years.

Besides this, it has been shown that, although the Mann-Kendall test should be non-parametric, the test shows different significant results for different distributions. A normal distributed dataset might detect less trends for the same significance value as a skewed dataset (Sheng Yue, Paul Pilon, & George Cavadias, 2002). Caution should therefore be taken when linking these trends to processes as for instance the changing climate.

5.2.3 Snow Covered Area

The current code does not consider the actual area of the elevation steps. To get a more accurate analysis, the trends need to be combined with a DEM in order to make a guess on the area of snow affected. In the method used in this thesis, the area was one of the first datasets lost in the reduction process, and therefore it was difficult to link it back in the end. Smaller elevation steps or using the elevation steps depending on a probability density function will give a better picture of the decline in snow cover area.

5.2.4 Runtime

The runtime was around six hours for the analyses. This is mainly due to the loops that calculate the yearly SCD. Using some of the principles needed for GEE might greatly improve the runtime. Functional programming can also be applied in Python and will reduce running time. Even if GEE is not used for further analyses of the RSLE, restructuring the method to be able to use functional programming is recommended for large datasets. Brute force methods are used often in hydrological programming, but it is these methods that take a lot of computational resources and, due to loops, can only run on one CPU core at a time. Being able to utilize the full capacity of the computer's CPU increases the number of computations that can be done on average computers.

5.3 Discussion on Google Earth Engine

It was interesting to discover the possibilities of GEE. It is a powerful tool which seems to be used increasingly. However, in working with GEE some issues arose which made the code more difficult to run.

5.3.2 General shortcomings

When adapting the RSLE to fit GEE, the limits of GEE were often met. When one wants to print a collection consisting of more than 5000 items, GEE gives an error. Therefore, one would have to reduce the number of items, or print only the first item of a collection. When a result is not printed, but only exported, there are no limits to the number of items that can be in the collection, which seems to be positive, but only in part. It means that large collections, such as the one used in this report can be calculated, and export

tasks can be run. In the meantime, the browser in which is programmed can be shut down and the program will continue to run, which is a useful feature.

However, since there is no maximum to the export, only a maximum runtime, one could start a calculation with a very large dataset, and have the task run out of time after twelve days. It is not possible to know whether a task will run out of time when starting. To know one would have to wait twelve days to find out. If a maximum had been set for the export, or for the amount of calculations needed, the error could be raised immediately, which saves a lot of time. The lack of such a maximum also means that malicious individuals could send many huge tasks to the servers, slowing down and refusing tasks for all users. This has happened during the development of this code, which meant that no tasks could be carried out for a few days. Since GEE is running on server space of the Google servers, busy days at Google mean slow calculation times for GEE, and an exact number for the calculation time is therefore hard to estimate.

Programming in GEE can still be difficult to grasp when starting to use it. If calculations need to be done which are similar to the examples and manual of GEE (Google Developers, n.d.), it is easy to do. When calculations become more difficult, and errors arise, it becomes increasingly tedious to debug. Especially when handling large datasets and difficult calculations and manipulating these datasets in ways that are not conventional, it becomes a difficult process to solve issues. In the case of this research, it happened when trying to solve the summation and minimization per grid cell to find the RSLE. A feature collection had to be combined with an image collection, which was only possible in a nested function. Before this part worked an error was raised which was over 100 pages long. Eventually all the problems were solved, but the code does not reduce the running time as expected.

5.3.2 GEE code issues

As mentioned in the method, the code is not an efficient code. It works, however the method chosen is demanding. Even though multiple servers can be used in GEE, the current code has so many computations to run that it takes a long time. For the 0.25° resolution, there are about 1200 grid cells in a feature collection, which must be combined with around 120 days of data (1/3 of a year) which are images consisting of 34 bands. In the nested function (Snippet 6) the sum of the snow pixels must be calculated for each band and then minimized to find the minimum band in that grid cell on that day. This calculation must be done approximately 150,000 times. The cloud threshold must be calculated for all the days and cells as well, which are again 150,000 calculations, after which each cell on each day must be compared to the RSLE set to filter out the cloudy data. This is expected to be the cause for the slow running code.

Apart from the parts of the code that decrease the running time, there are two more issues. The first has already been solved but still should be mentioned. There is a maximum number of bands allowed for images. The DEM in the GEE code is split into 34 bands. At one moment the maximum bands in GEE was set to 30 after which the code did not run anymore. It has been increased again so it is no longer a problem, but if the elevation steps must be smaller, for instance 10 meters, it may present an issue.

The second issue is with the code itself. The minimum and maximum check, which is in the nested function, does not always seem to work. Since there are always 34 DEM bands ranging from 100 to 3500 meters, even when those elevations are not present, there had to be a mechanism to prevent snowlines being at elevations which are not present. This was done by setting the RSLE to the minimum elevation if the calculated RSLE was lower than that, and to the maximum elevation if the calculated RSLE was higher than that maximum. The problem with this threshold is the spikey character of the RSLE, which sometimes resulted in a RSLE on the elevation band just under the maximum. It is suspected that these outliers are because of clouds on the highest elevations. In combination with an error in the analysis code, the consequence of the threshold caused the highest elevation band before the maximum to show 365 days of SCD in some years. It is seen as an error since it did not just happen for the high altitudes, but also for low altitudes which do not normally have permanent snow cover.

5.3.3 Improved code and its issues

To improve the GEE code an alternative code has been developed which cancels the need for a grid. As a result, the feature collection consisting of 1200 features, and the nested function are no longer needed, causing major improvements in calculation time. The code uses the GEE function *ReduceResolution* to calculate the snow cover in areas. *ReduceResolution* is a way to reduce the resolution of an image to make it more usable. When *ReduceResolution* is used, the method of reduction must be specified, which is the property used in order to create the RSLE dataset. Because this code is more effective, it was combined with a DEM file which does not have 100 m steps but is varies for each pixel based on a probability density function. The DEM file now has percentage bands, meaning that a reduced pixel in the 4% band has the value of the elevation where 4% of the land is lower than that elevation (D2 in Snippet 11).

Estimating snow cover decline using the RSLE method in Google Earth Engine

```
var C_A = ee.FeatureCollection("users/afstuderenthijis/C_A"),
    D1 = ee.Image("USGS/GMTED2010");

//The reprojection variables
var crs = 'EPSG:3857';
var scale = 30000;

//import Datasets
var M2 = ee.ImageCollection('MODIS/006/MOD10A1')
  .select('NDSI_Snow_Cover').filterDate('2001-04-04','2003-09-30')
  .map(function(img) {return img.gte(40)});

//percentile list
var start = 2;
var end = 98;
var step = 2;
var percent = ee.List.sequence(start,end,step);

//elevation bands based on pdf
var D2 = D1.reduceResolution({reducer:ee.Reducer.percentile(percent),
maxPixels:15000,bestEffort:false}).reproject({crs:crs,scale:scale});

//create snow/no snow map (1 for land above percentile, and snow below)
var D3 = D1.gt(D2).clip(C_A);
var x = function(img) {return img.subtract(D3).abs()};
var S1 = M2.map(x);

// sum up all pixels which need to be minimized
var S2 = S1.map(function(img) {var rr = img.reduceResolution({
  reducer:ee.Reducer.sum().unweighted(),maxPixels:15000,
  bestEffort:false}).reproject({crs:crs,scale:scale}).multiply(-1);
  return ee.Image(rr)});

//create the quality mosaic for all dates
var qualitymodel = function(snow) {
  var BN = D2.bandNames();
  var flipD = function(ls) {return D2.select(ee.String(ls))}; //transform
  image with 19 bands to 19 singleband images
  var flipS = function(ls) {return snow.select(ee.String(ls))};
  var LD2 = ee.ImageCollection(BN.map(flipD)).map(function(img) {return
  img.rename('elevation')}); // combine singleband images into
  imagecollection
  var LS2 = ee.ImageCollection(BN.map(flipS)).map(function(img) {return
  img.rename('snow')});
  var joinfilter =
  ee.Filter.equals({leftField:'system:index',rightField:'system:index'});
  //make 1 imagecollection
  var SD = ee.ImageCollection(ee.Join.inner().apply(LD2,LS2,joinfilter));
  var SD2 = SD.map(function(ft) {return ee.Image.cat(ft.get('primary'),
  ft.get('secondary'))});
  var QM = SD2.qualityMosaic({qualityBand:('snow')}); //make qualitymosaic
  return ee.Image(QM.select('elevation'));
}
var final = S2.map(qualitymodel);
```

Snippet 11, the improved code for Google Earth Engine. It is much faster, but the cloud threshold is missing, and the export does not work. It is based on a reduction of the resolution, from the data resolution to the grid resolution where the final pixel values represent the RSLE.

After creating a MODIS dataset with 1s and 0s as in the first method and combining it with the split DEM (D3), the image is reduced to a lower resolution by summing up the data inside that pixel. For instance, a pixel of 30 by 30 km consists of several 500x500m pixels. The sum of these pixels is the dataset which needs to be minimized to find RSLE. That sum was previously saved in a collection, but in this method becomes the new pixel value. An image now consists of several bands for each elevation, with its pixels representing the sum of snow pixels below that elevation and land above it (S2 in Snippet 11). To be able to calculate the RSLE the minimum pixel value from every band must be linked to the elevation of that band.

To do this calculation, a function is developed which is mapped over the reduced collection. This function takes a band of the reduced DEM (D2) and combines it with the respective band in S2, corresponding to that elevation. It makes an image from those two bands. This combination is done for all the bands, resulting in 2 images with 49 bands being turned into 49 images with 2 bands. These are then combined into an image collection which is needed to do a quality mosaic. A quality mosaic takes of all the images the pixel with the best value in a quality band and creates 1 image from the entire collection. The quality band is the band that needs to be minimized (previously in collection S2), but the band that is saved is the elevation band. The result is an image with all the RSLE values in each pixel. Since this is all written in a function which is mapped over an entire collection, the result is a collection with images that have the RSLE as pixel values. This method is significantly more complex, but significantly faster.

There are two issues with this method, causing it to not yet have usable results. The cloud threshold should still be developed, and the export is not useful. Since GEE cannot export an image collection, and the result of this code is an image collection, the download can only be completed by downloading every single image separately. Normally, this would be possible using a batch downloader or something similar, but in GEE the images are exported to Google Drive, and each task must be started manually. Unfortunately, one cannot create a function that executes all the exports at once since only 30 exports can be in the tasks list and thus run at the same time, which is done by GEE to prevent malicious individuals from occupying all the server space. Due to these issues the effective code is not used for further analyses. It should be however used to continue researching if the RSLE can be made more effective in GEE.

5.3.4 Continuing this research in GEE.

To be able to finish the effective code in GEE two things need to happen. First, the cloud threshold must be added.

Second, the analysis must be done already partly in GEE, creating a CSV with SCD per year or other data that can easily be exported. It might be useful to use the Python link developed for GEE. It is a bit more difficult to install but creates a direct link between the computer and GEE.

6. Conclusion

The Regional Snow Line Elevation method is a good method to improve datasets on snow cover. The method however must be improved to be able to handle very large datasets. A more efficient method will make it easier to combine the dataset with water content, snow thickness, radiation, precipitation and other datasets to get a complete view of the behaviour of snow in mountainous areas under a changing climate. Therefore, in the introduction the following research goal was stated:

Develop a method that decreases the computation time of the RSLE and decreases the data volume to be downloaded.

To test if the developed method worked, the resulted data was investigated with the following research question:

Can trends be found in yearly snow cover duration in the Caucasus region?

A method has been developed in Google Earth Engine that works. That said, it only decreases the data volume, not necessarily the computation time. Although a more effective approach has been developed in GEE, this approach has not yet been finalized and is not able to export the data in an effective way. However, with further development, it will be able to use the RSLE effectively in GEE. Google Earth Engine is a good platform for research making use of remote sensing data, even though it has its complications, and is a good addition to traditional programming in hydrology.

After analysing the downloaded data, the snow cover duration could be calculated for different elevations in different areas. Trends of snow cover duration are observed over the entire Caucasus but are concentrated in the South East. The trends were most abundant at 1500 and 2500 meters. Except for one location in the Armenian plateau, all significant trends show a decline of snow cover duration with up to four days a year, which is a total decline of more than two months of the snow season.

The number of significant trends, that is, trends calculated using the Mann-Kendall test with a p-value smaller than 0.05, is around 17% for both the 0.25° and 0.50° resolutions. The differences between the 0.25° and 0.50° resolutions were small however, the 0.25° resolution showed more detail.

The declining trends of snow cover have a big impact on the local hydrology. The melt season might start earlier, influencing ecology in the area. The summer drought will also start at an earlier date and last longer, while winter flow increases due to precipitation falling as rain instead of snow. The overall discharge of the Euphrates, Tigris, Kura, Aras and other rivers in the area, is expected to decrease, which might cause water shortages in a region already divided by political tension.

Following the findings of this thesis, the first step would be to calculate the trends for smaller grid cells, combine them with snow covered area and estimate the effect on discharge. Snow cover trends in the Caucasus can greatly influence the water supply, not only for the Caucasus, but for all the surrounding countries. For countries to prepare for water shortages, and perhaps start diplomatic talks to solve issues, detailed information is needed on expected water resources in the future under different climate scenarios.

Bibliography

- Altingoz, M., & Ali, S. H. (2019). Environmental Cooperation in Conflict Zones: Riparian Infrastructure at the Armenian–Turkish Border. *Journal of Environment and Development*, 28(3), 309–335. <https://doi.org/10.1177/1070496519859680>
- Baldasso, V., Soncini, A., Azzoni, R. S., Diolaiuti, G., Smiraglia, C., & Bocchiola, D. (2019). Recent evolution of glaciers in Western Asia in response to global warming : the case study of Mount Ararat , Turkey. *Theoretical and Applied Climatology*, 137, 45–59. Retrieved from <https://doi.org/10.1007/s00704-018-2581-7>
- Barnett, T. P., Adam, J. C., & Lettenmaier, D. P. (2005). Potential impacts of a warming climate on water availability in snow-dominated regions. *Nature*, 438, 303–309. <https://doi.org/10.1038/nature04141>
- Bouman, C. (2018). *Root-zone storage and snow cover effects, A catchment study on the dynamic behaviour of the root-zone moisture capacity related to changing snow cover patterns*. Delft University of Technology.
- Bunn, S. E., & Arthington, A. H. (2002). Basic principles and ecological consequences of altered flow regimes for aquatic biodiversity. *Environmental Management*, 30(4), 492–507. <https://doi.org/10.1007/s00267-002-2737-0>
- Danielson, J. J., & Gesh, D. B. (2011). *Global multi-resolution terrain elevation data 2010 (GMTED2010): U.S. Geo- logical Survey Open-File Report 2011–1073*. Reston, Virginia.
- Descloitres, J. (2002). Visible Earth - Caucasus Mountains Mesopotamia. Retrieved from <https://visibleearth.nasa.gov/images/58437/caucasus-mountains-to-mesopotamia?size=all>
- FAO. (2016a). AQUASTAT Euphrates – Tigris River Basin. Retrieved October 9, 2019, from <http://www.fao.org/nr/water/aquastat/basins/euphrates-tigris/index.stm>
- FAO. (2016b). AQUASTAT Kura – Araks River Basin. Retrieved October 9, 2019, from <http://www.fao.org/nr/water/aquastat/basins/kura-araks/index.stm>
- Fugger, S. (2018). *Snow cover in the Greater Alpine Region (2000 – 2017): regional patterns, controls and change*. University of Natural Resources and Life Sciences(BOKU), Austria. Retrieved from https://forschung.boku.ac.at/fis/suchen.person_betreuungen?sprache_in=en&me_nue_id_in=107&id_in=130688
- Google Developers. (n.d.). Google Earth Engine - Introduction. Retrieved October 12, 2019, from <https://developers.google.com/earth-engine>
- Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. (2017). Google Earth Engine : Planetary-scale geospatial analysis for everyone. *Remote Sensing of Environment*, 202, 18–27. <https://doi.org/10.1016/j.rse.2017.06.031>
- Hagg, W., Shahgedanova, M., Mayer, C., Lambrecht, A., & Popovnin, V. (2010). A

- sensitivity study for water availability in the Northern Caucasus based on climate projections A sensitivity study for water availability in the Northern Caucasus based on climate projections. *Global and Planetary Change*.
<https://doi.org/10.1016/j.gloplacha.2010.05.005>
- Hall, A., & Qu, X. (2006). Using the current seasonal cycle to constrain snow albedo feedback in future climate change. *Geophysical Research Letters*, 33.
<https://doi.org/10.1029/2005GL025127>,
- Hall, D. K., & Riggs, G. A. (2016). MODIS/Terra Snow Cover Daily L3 Global 500m SIN Grid, Version 6. Boulder, Colorado, USA: NASA National Snow and Ice Data Center Distributed Active Archive Center.
<https://doi.org/https://doi.org/10.5067/MODIS/MOD10A1.006>
- Hall, Dorothy K, & Riggs, G. A. (2015). *MODIS Snow Products Collection 6 User Guide*. Retrieved from <https://nsidc.org/sites/nsidc.org/files/files/MODIS-snow-user-guide-C6.pdf>
- Härer, S., Bernhardt, M., Siebers, M., & Schulz, K. (2018). On the need for a time- and location-dependent estimation of the NDSI threshold value for reducing existing uncertainties in snow cover maps at different scales. *The Cryosphere*, 12, 1629–1642. Retrieved from <https://doi.org/10.5194/tc-12-1629-2018>
- Hatfield, J. L., Boote, K. J., Kimball, B. A., Ziska, L. H., Izaurralde, R. C., Ort, D. R., ... Wolfe, D. (2011). Climate Impacts on Agriculture : Implications for Crop Production. *Agronomy Journal*, 103(2), 351–370.
<https://doi.org/10.2134/agronj2010.0303>
- Hauksson, B. Þ. (2017). *Estimation of regional snow-line elevations in Norway, Testing a method for estimating RSLE and melt slope analyses in Norway*. University of Oslo.
- Huning, L. S., & Aghakouchak, A. (2018). Mountain snowpack response to different levels of warming. *PNAS*, 115(43). <https://doi.org/10.1073/pnas.1805953115>
- Kendall, M. G. (1975). *Rank Correlation Methods*. London: Griffin.
- Kottek, M., Grieser, J., Beck, C., Rudolf, B., & Rubel, F. (2006). World Map of the Köppen-Geiger climate classification updated. *Meteorologische Zeitschrift*, 15(3), 259–263. <https://doi.org/10.1127/0941-2948/2006/0130>
- Krajčí, P., Holko, L., & Parajka, J. (2016). Variability of snow line elevation, snow cover area and depletion in the main Slovak basins in winters 2001-2014. *Journal of Hydrology and Hydromechanics*, 64(1), 12–22.
<https://doi.org/10.1515/johh-2016-0011>
- Krajčí, P., Holko, L., Perdigião, R. A. P., & Parajka, J. (2014). Estimation of regional snowline elevation (RSLE) from MODIS images for seasonally snow covered mountain basins. *Journal of Hydrology*, 519, 1769–1778.
<https://doi.org/10.1016/j.jhydrol.2014.08.064>
- Lur'e, P. M., & Panov, V. D. (2011). Problems of exploration level of hydrometeorological regime of the Northern Caucasus territory. *Russian Meteorology and Hydrology*, 36(4), 273–278.

<https://doi.org/10.3103/S1068373911040091>

Mann, H. B. (1945). Nonparametric Tests Against Trend. *Econometrica*, 13(3), 245–259. Retrieved from <https://www.jstor.org/stable/1907187>

NASA. (n.d.). MODIS. Retrieved October 12, 2019, from <https://modis.gsfc.nasa.gov/data/>

Özdoğan, M. (2011). Climate change impacts on snow water availability in the Euphrates-Tigris basin. *Hydrology and Earth System Sciences*, 15(9), 2789–2803. <https://doi.org/10.5194/hess-15-2789-2011>

Parajka, J., Pepe, M., Rampini, A., Rossi, S., & Blöschl, G. (2010). A regional snow-line method for estimating snow cover from MODIS during cloud cover. *Journal of Hydrology*, 381, 203–212. <https://doi.org/10.1016/j.jhydrol.2009.11.042>

Rets, E. P., Dzhamalov, R. G., Kireeva, M. B., Frolova, N. L., Durmanov, I. N., Telegina, A. A., ... Grigoriev, V. Y. (2018). Recent Trends of River Runoff in the North Caucasus. *Geography, Environment, Sustainability*, 11(3), 61–70. <https://doi.org/DOI-10.24057/2071-9388-2018-11-3-61-70>

Sheng Yue, Paul Pilon, & George Cavadias. (2002). Power of the Mann±Kendall and Spearman's rho tests for detecting monotonic trends in hydrological series. *Journal of Hydrology*, 259, 254±271.

Tashilova, A. A., Kesheva, L. A., Teunova, N. V, & Taubekova, Z. A. (2016). Analysis of Temperature Variability in the Mountain Regions of the North Caucasus in 1961 – 2013. *Russian Meteorology and Hydrology*, 41(9), 601–609. <https://doi.org/10.3103/S1068373916090028>

Tielidze, L. G., & Wheate, R. D. (2018). The Greater Caucasus Glacier Inventory (Russia , Georgia, Azerbaijan). *The Cryosphere*, 12, 81–94. Retrieved from <https://doi.org/10.5194/tc-12-81-2018>

Viviroli, D., Dürr, H. H., Messerli, B., Meybeck, M., & Weingartner, R. (2007). Mountains of the world , water towers for humanity : Typology , mapping , and global significance. *Water Resources Research*, 43. <https://doi.org/10.1029/2006WR005653>

Yılmaz, Y. A., & Aalstad, K. (2019). Multiple Remotely Sensed Lines of Evidence for a Depleting Seasonal Snowpack in the Near East. <https://doi.org/10.3390/rs11050483>

Yucel, I., Güventürk, A., & Lutfi, O. (2015). Climate change impacts on snowmelt runoff for mountainous transboundary basins in eastern Turkey. *International Journal of Climatology*, 35, 215–228. <https://doi.org/10.1002/joc.3974>

Yueh, S. H., Xu, X., Shah, R., Kim, Y., Garrison, J. L., Komanduru, A., & Elder, K. (2017). Remote Sensing of Snow Water Equivalent Using Coherent Reflection From Satellite Signals of Opportunity : Theoretical Modeling. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(12), 5529–5540. <https://doi.org/10.1109/JSTARS.2017.2743172>

Appendix A, Code Used

Link to original GEE code:

<https://code.earthengine.google.com/75d582cf7f6ac9ebd40f328a1a4fb32f>

Link to improved GEE code:

<https://code.earthengine.google.com/126da9f2dd0fdcad74b2e46ddc2994bd>

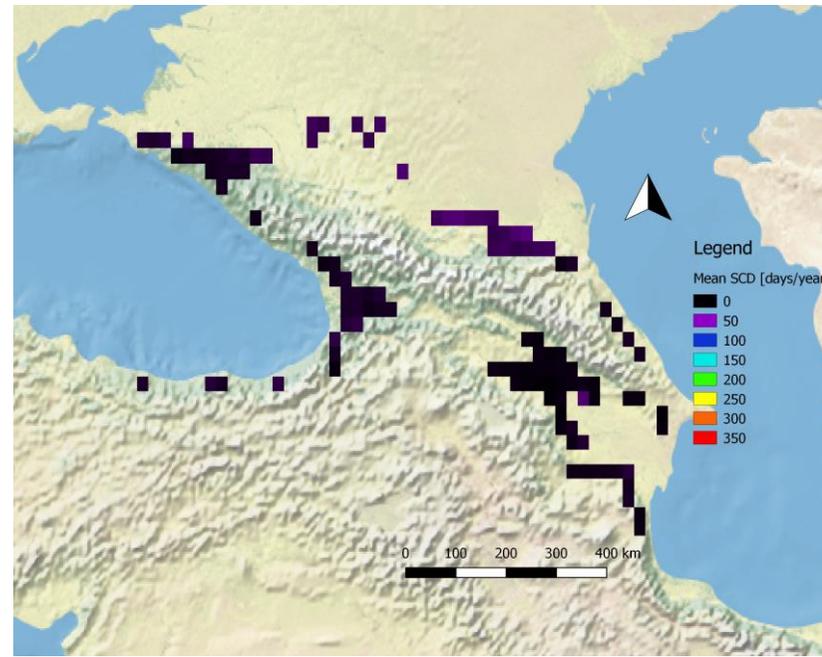
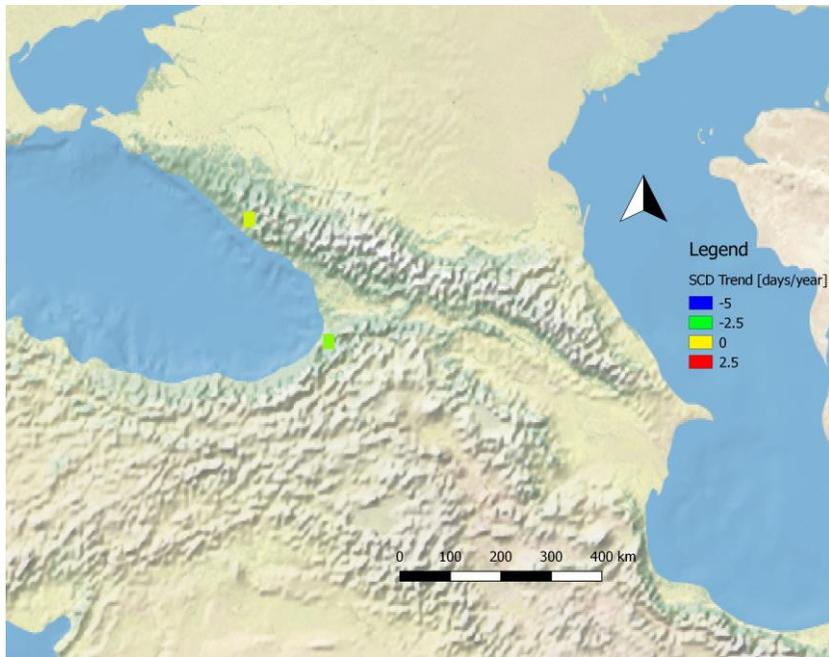
To be able to use these codes one must register for Google Earth Engine, and import the following asset

https://code.earthengine.google.com/?asset=users/afstuderenthijs/C_A

The Python codes used to pre-process the data and to calculate the SCD are available via

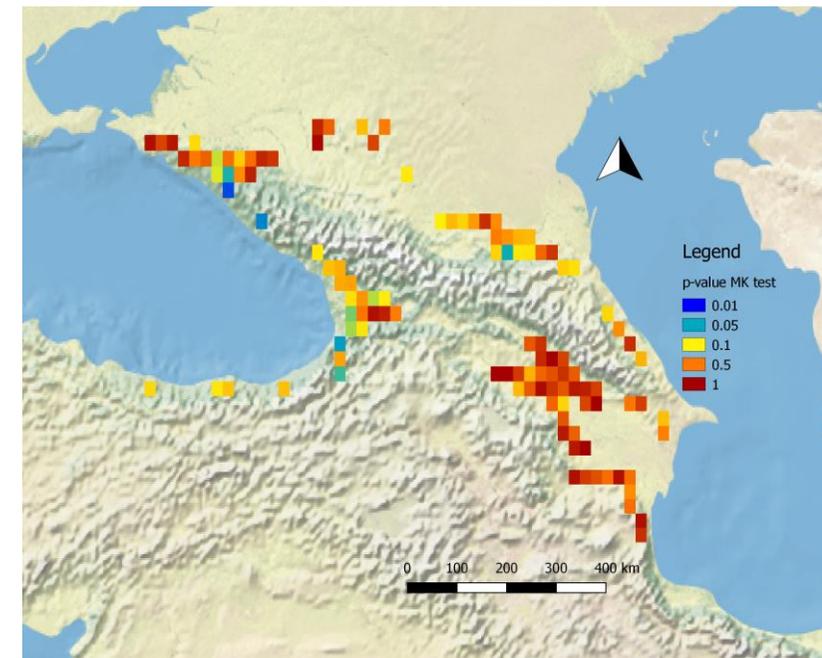
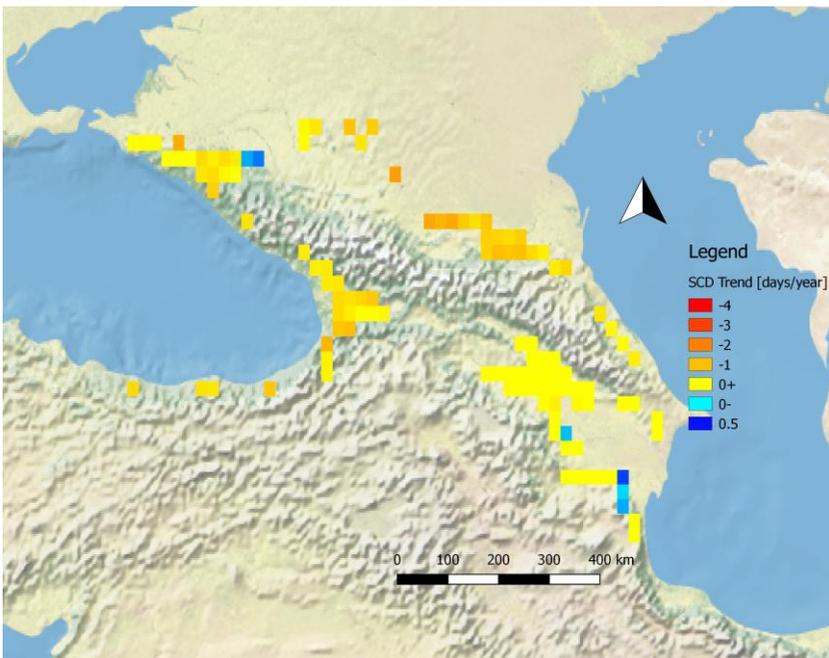
https://drive.google.com/drive/folders/1T7yrK5_Ubc55XW6TmqS_yU7QAL-pHUEJ?usp=sharing

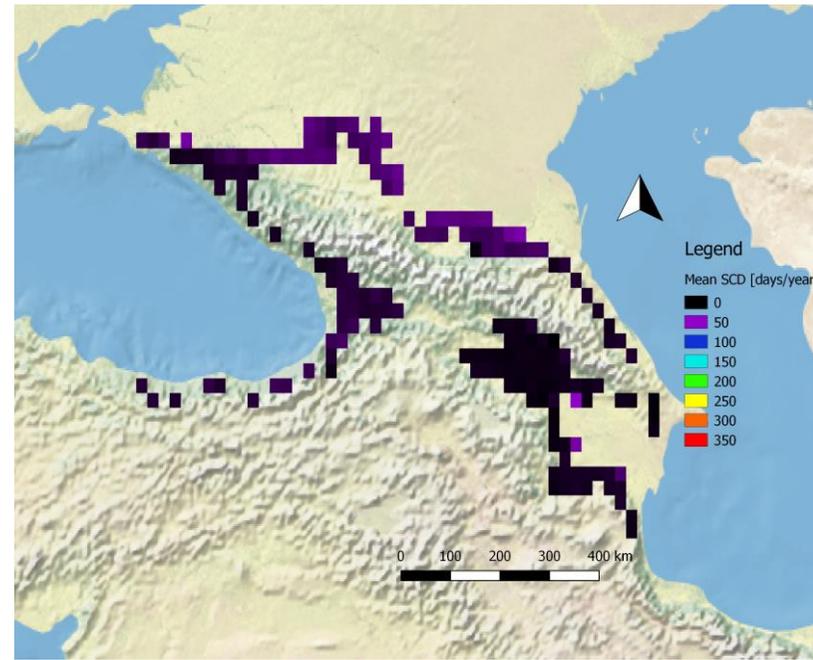
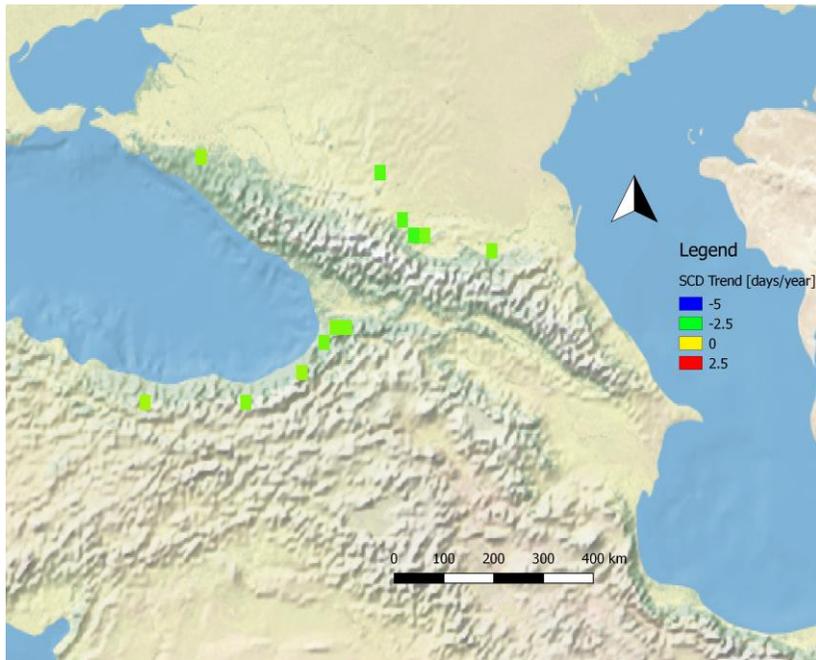
Appendix B, Trend Results for all Elevations



Elevation: 200 m

Top-Left: significant trends
Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends
Bottom-Right: p-values



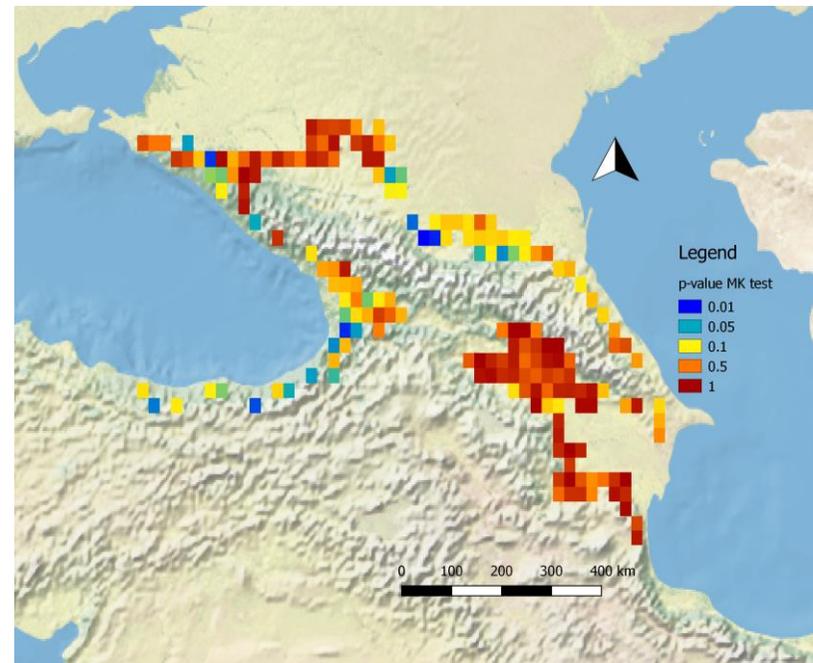
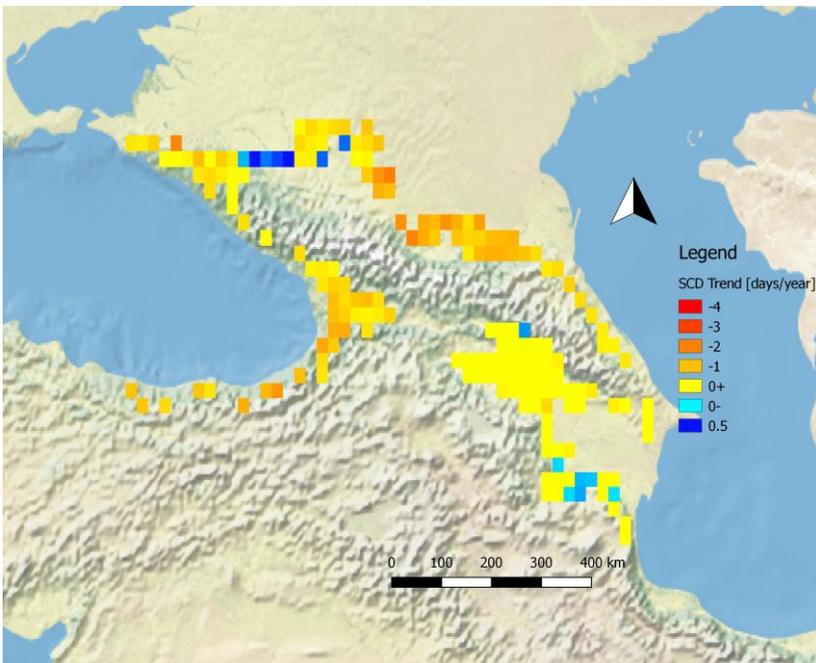


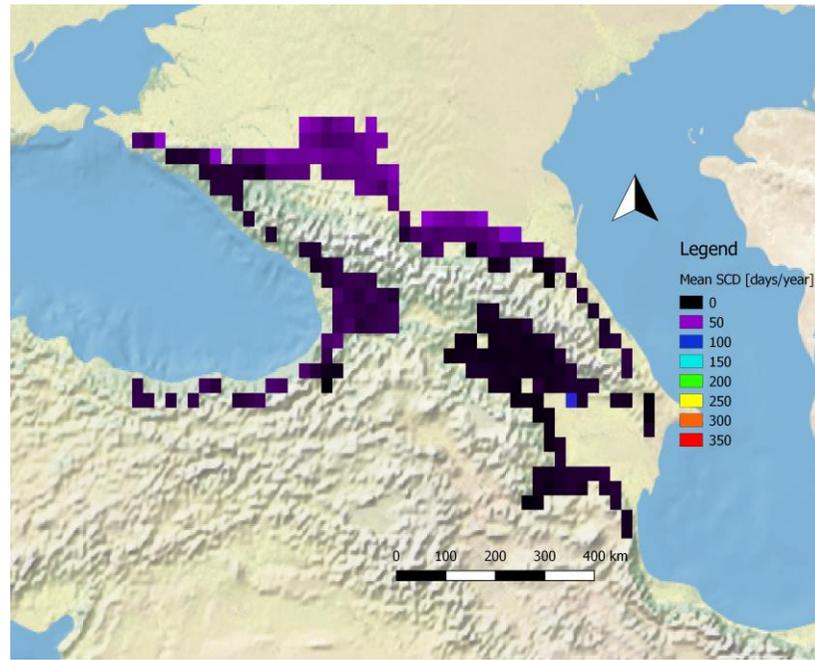
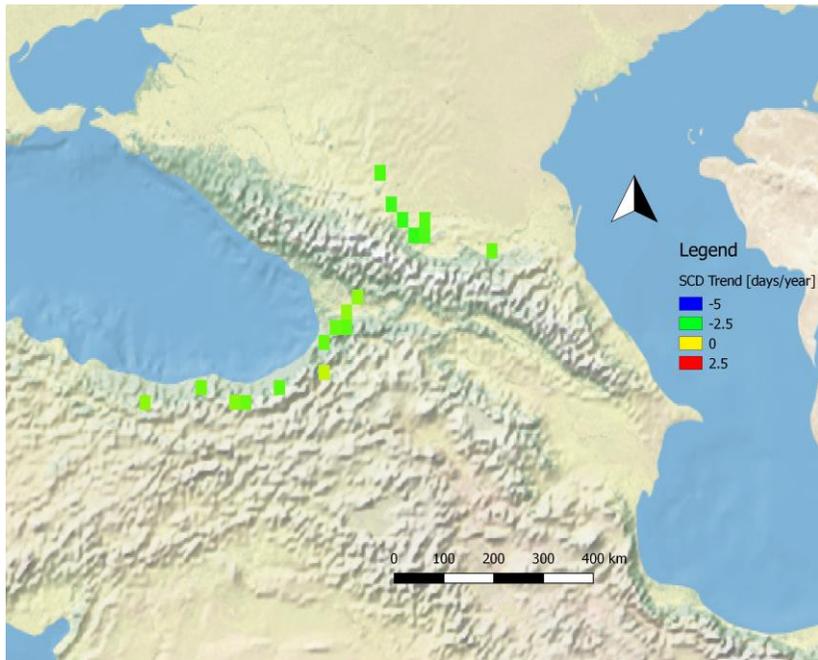
Elevation: 300 m

Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

Bottom-Right: p-values



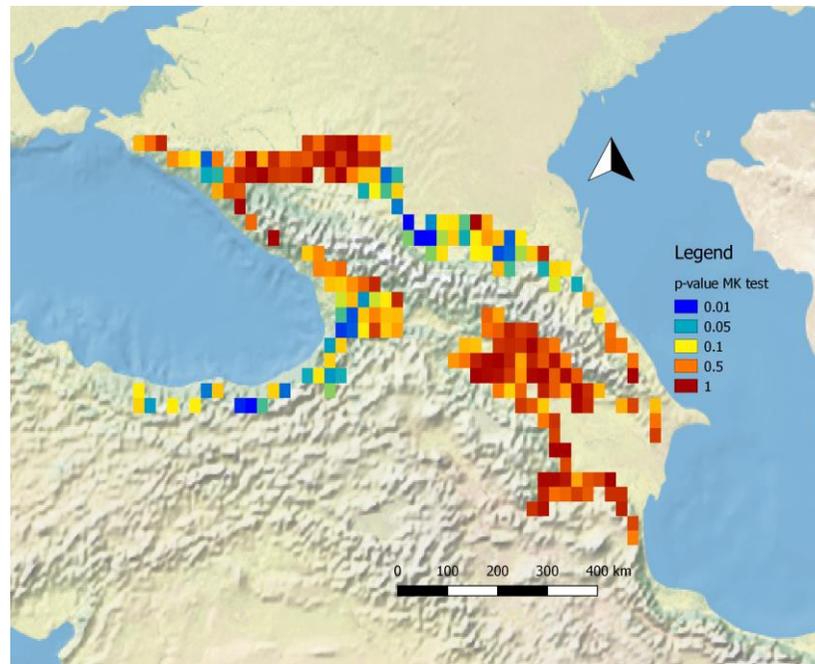
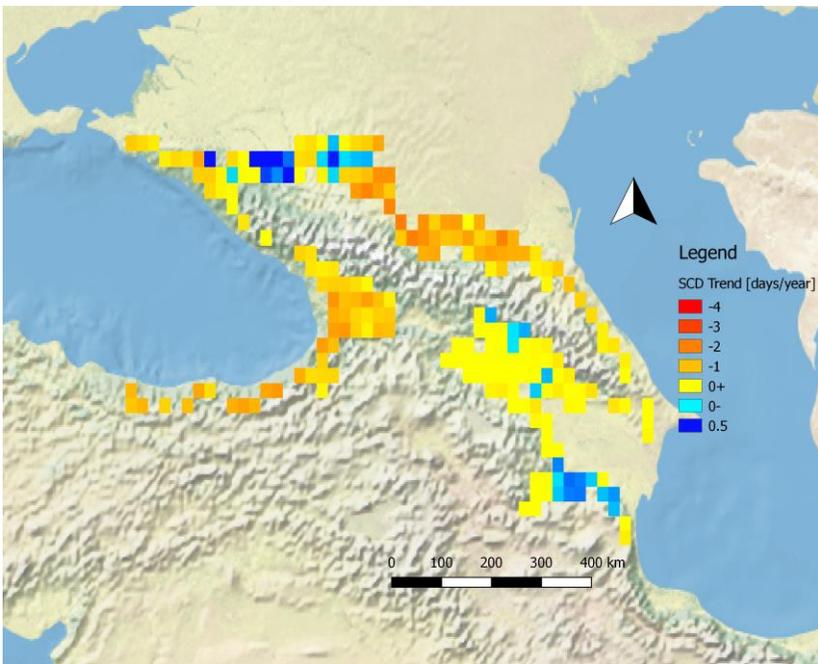


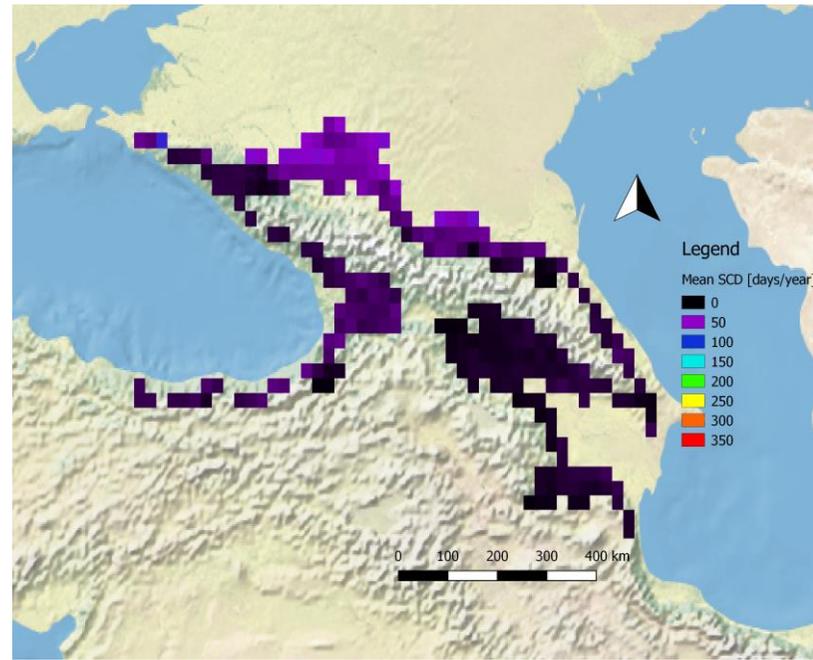
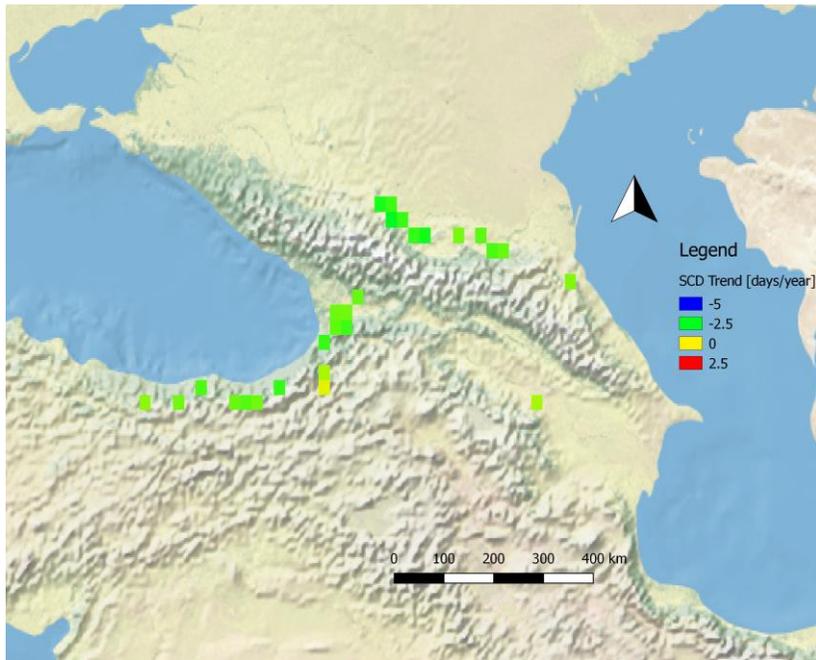
Elevation: 400 m

Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

Bottom-Right: p-values



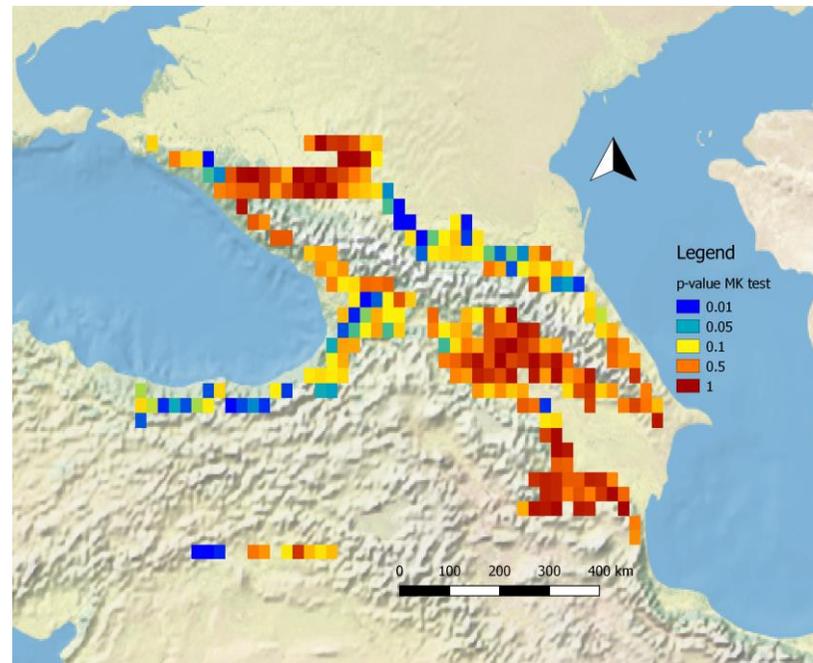
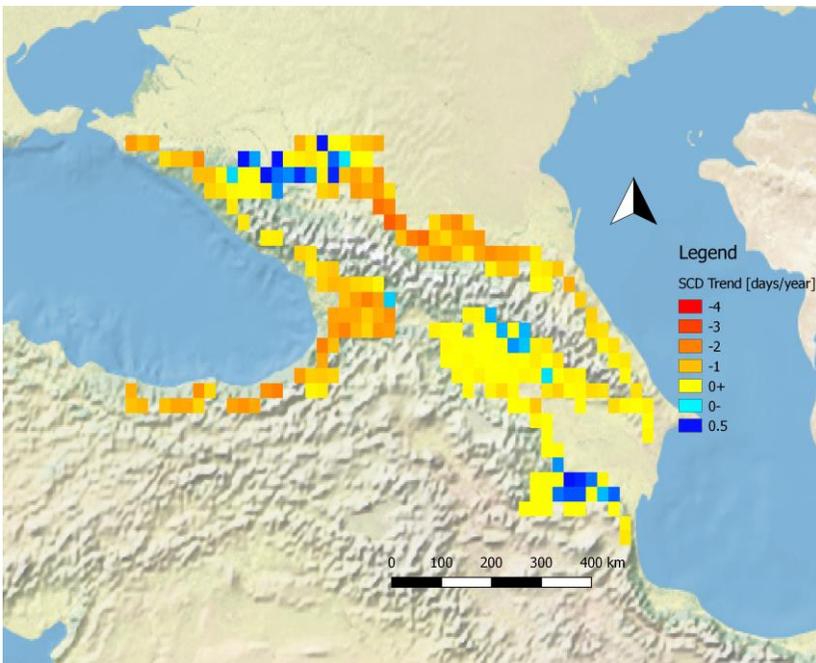


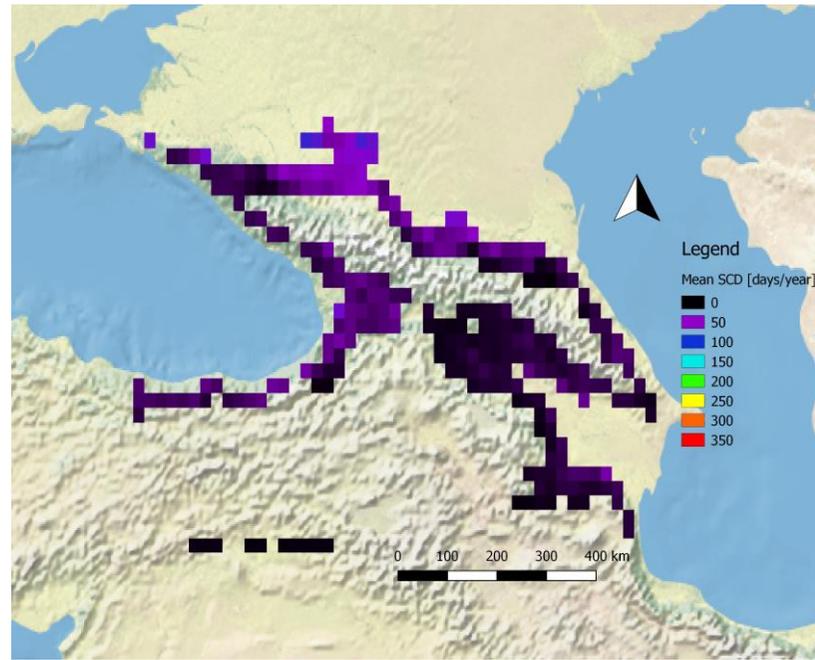
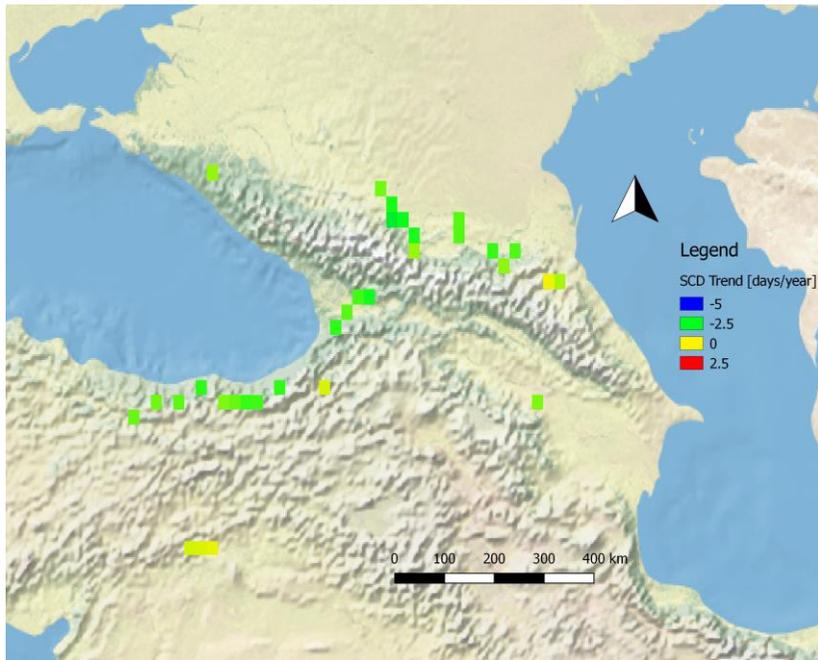
Elevation: 500 m

Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

Bottom-Right: p-values



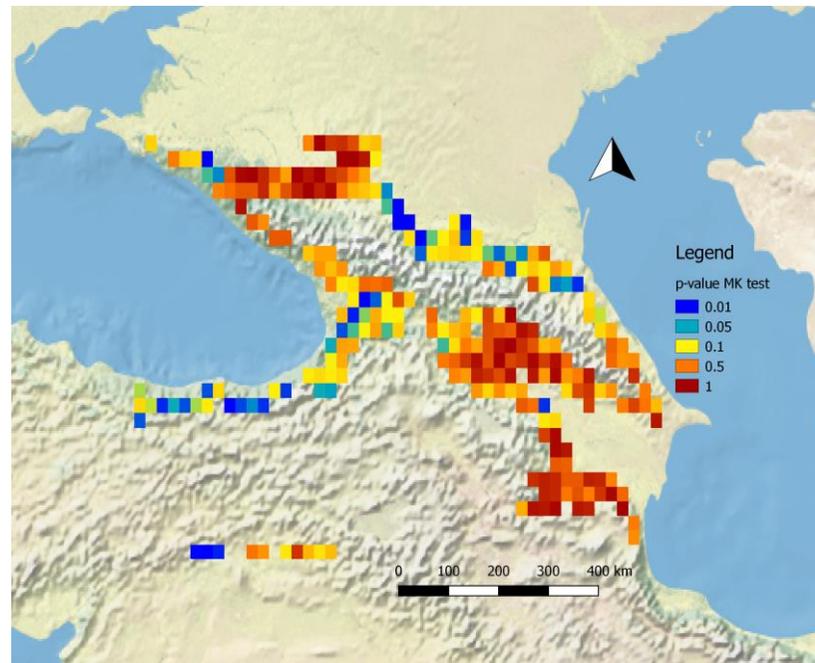
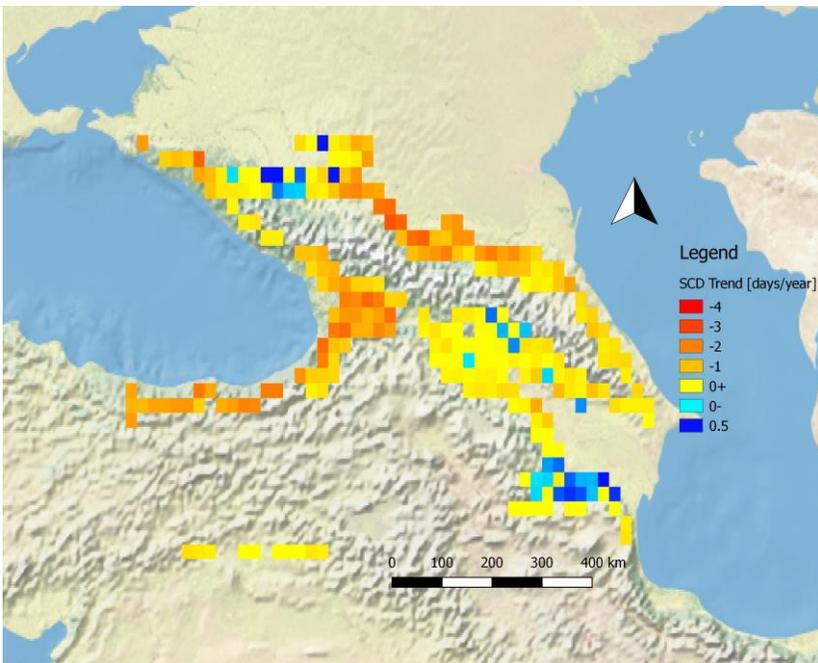


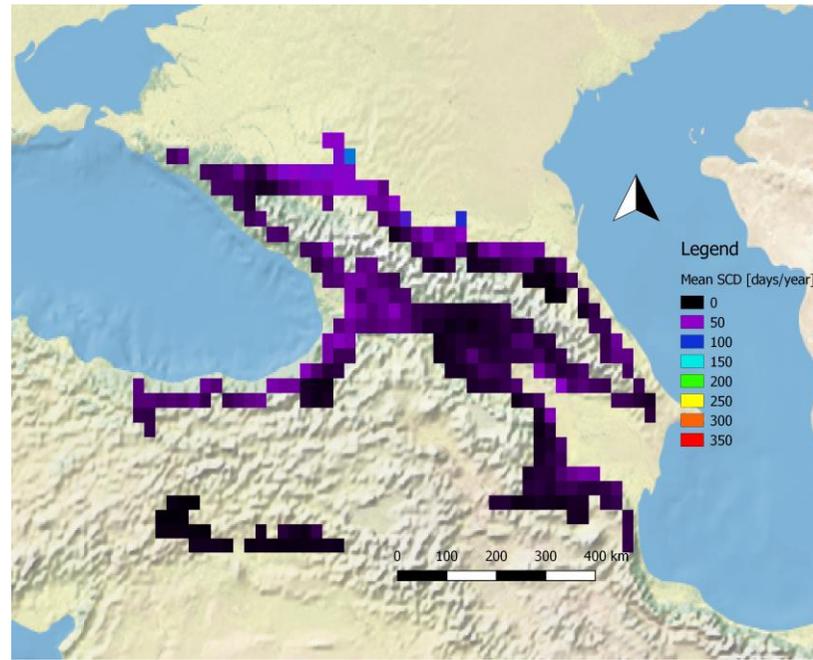
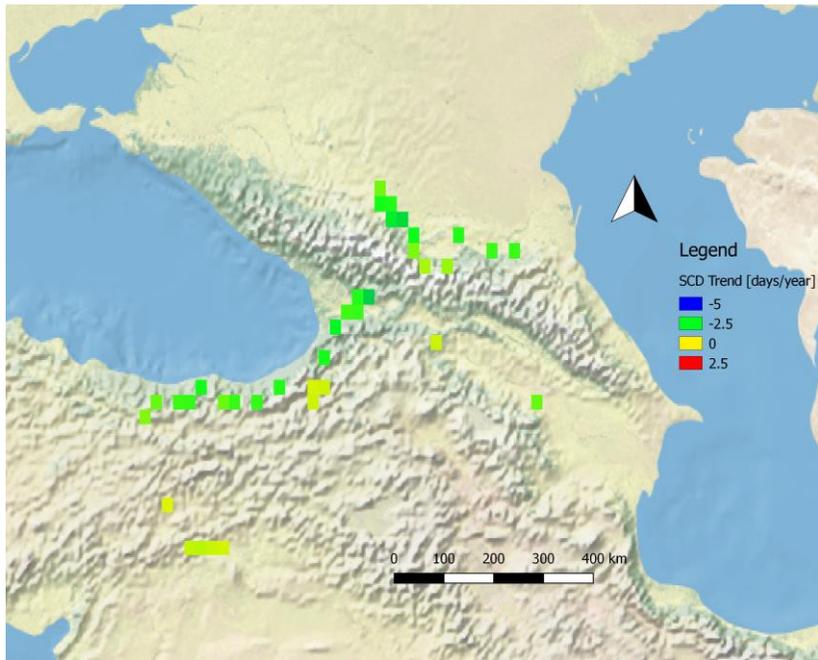
Elevation: 600 m

Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

Bottom-Right: p-values





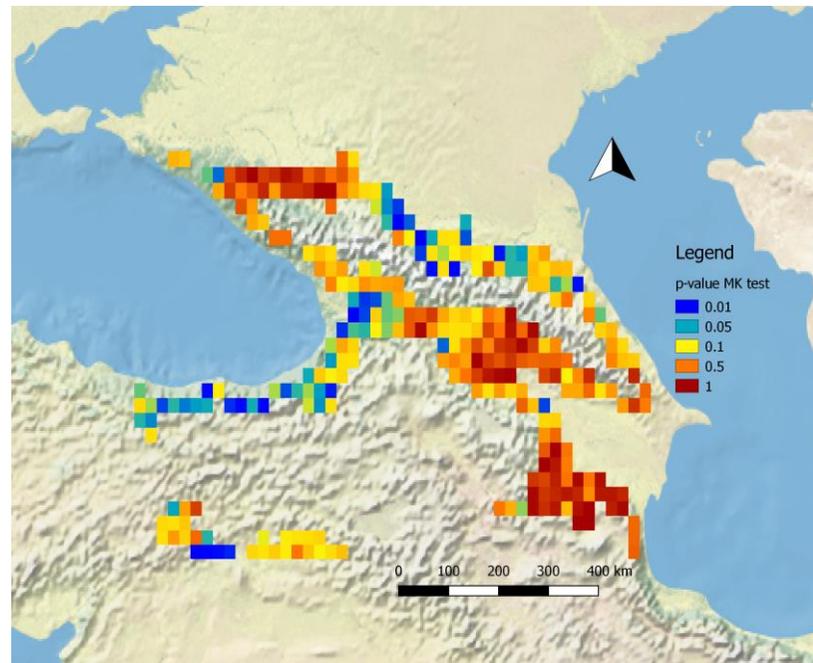
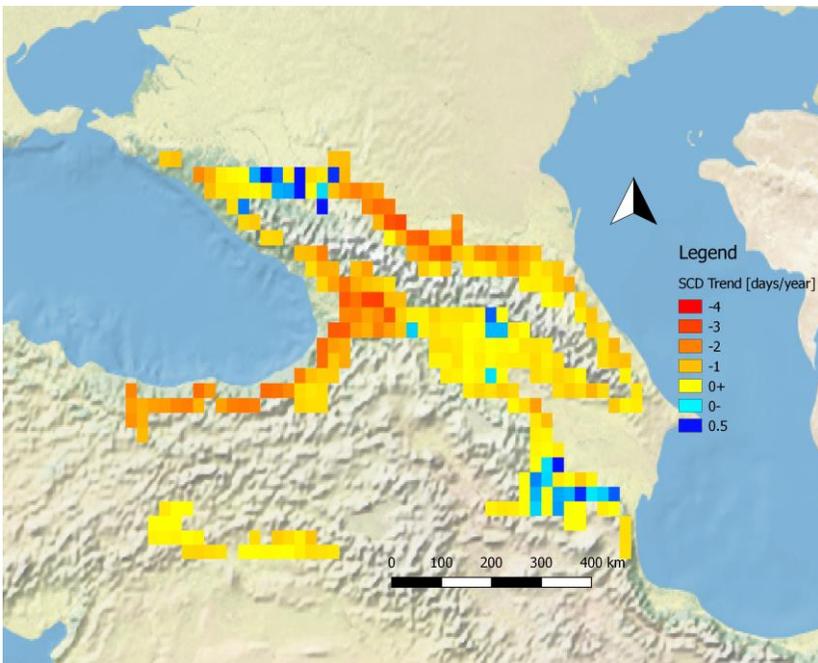
Elevation: 700 m

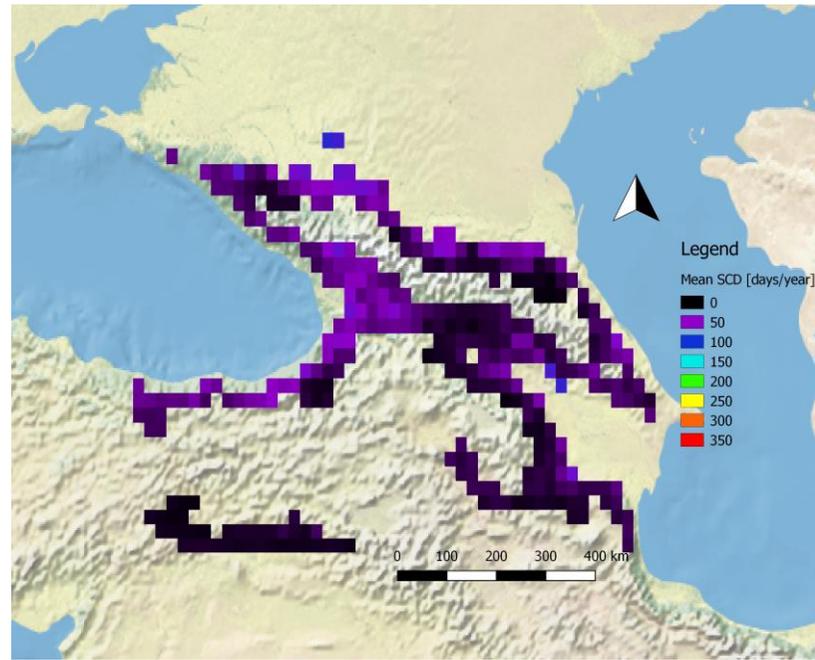
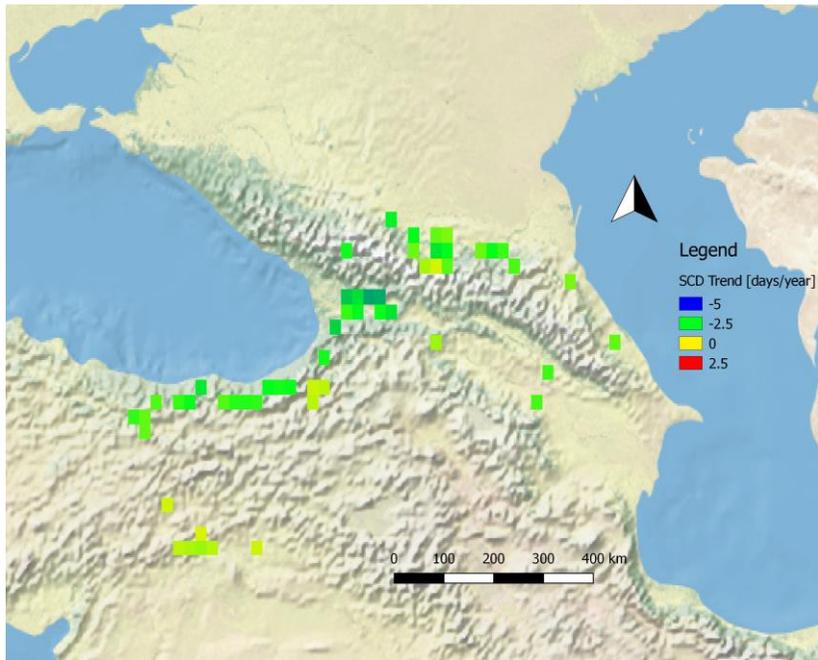
Top-Left: significant trend

Top-Right: mean SCD

Bottom-Left: All trends, including not significant trends

Bottom-Right: p-values



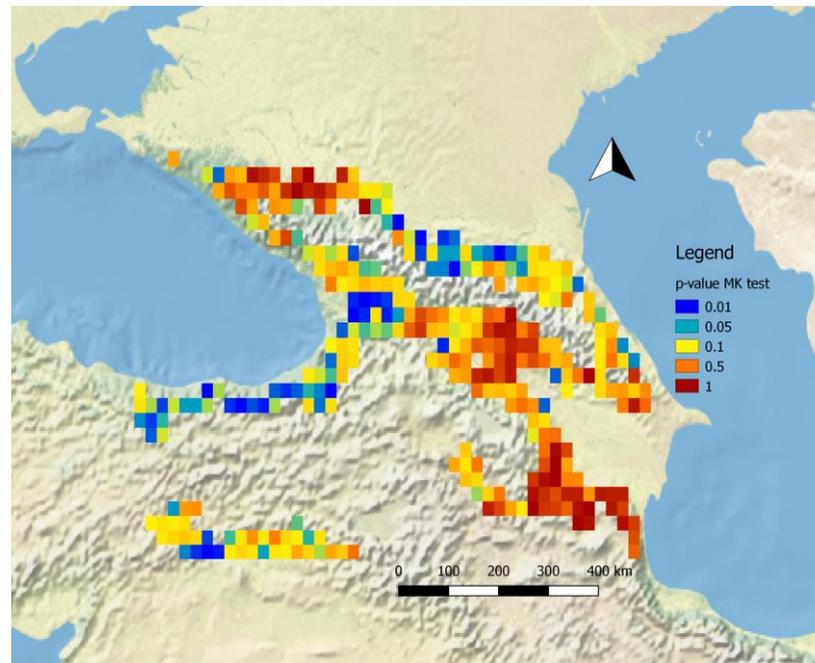
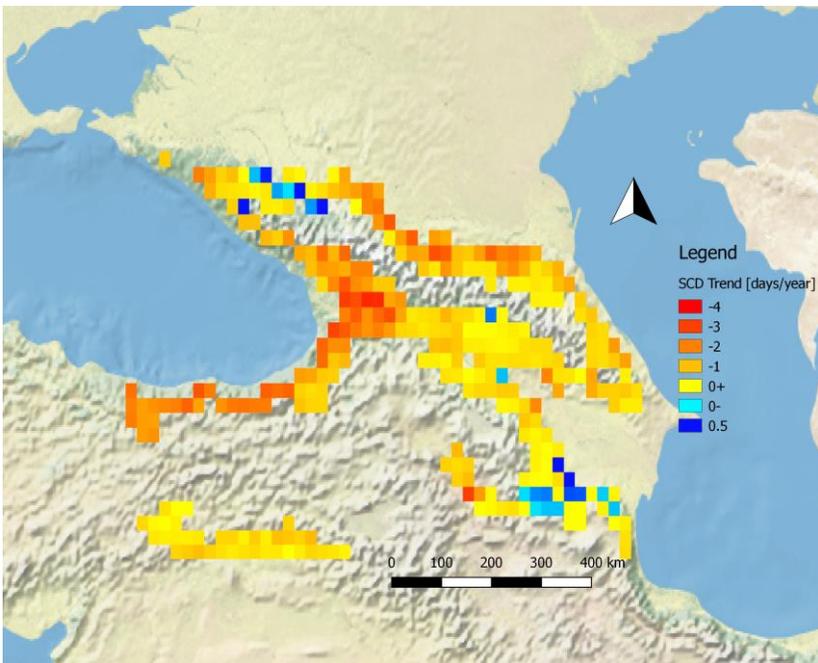


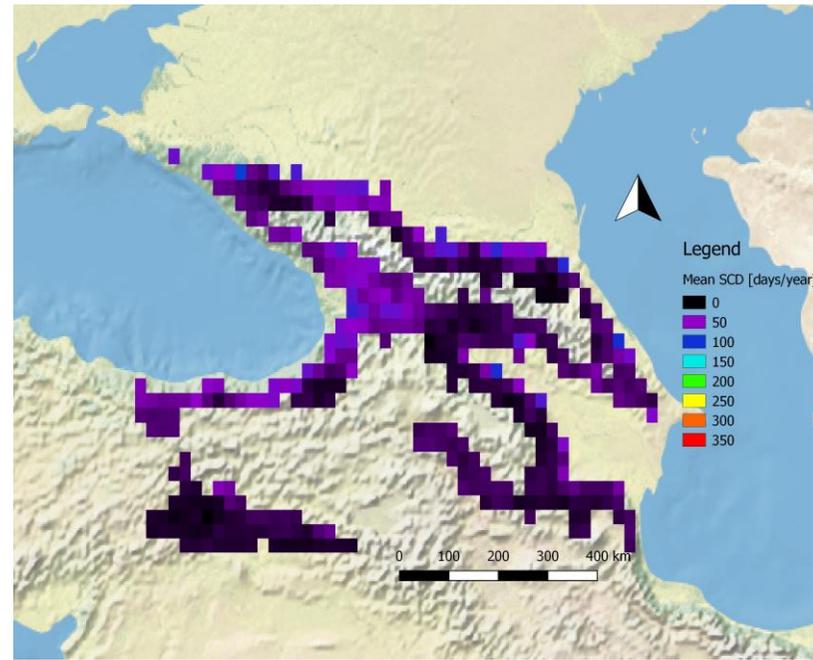
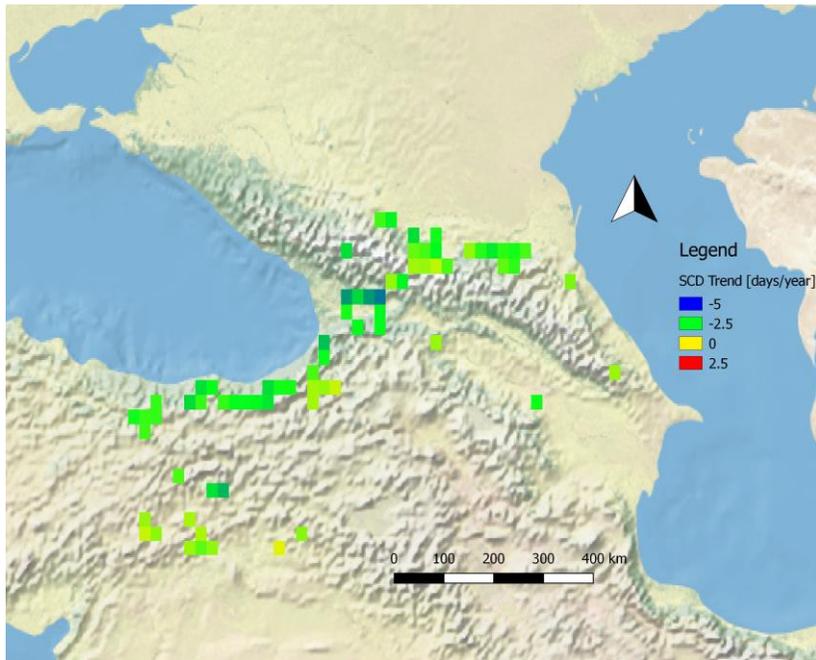
Elevation: 800 m

Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

Bottom-Right: p-values



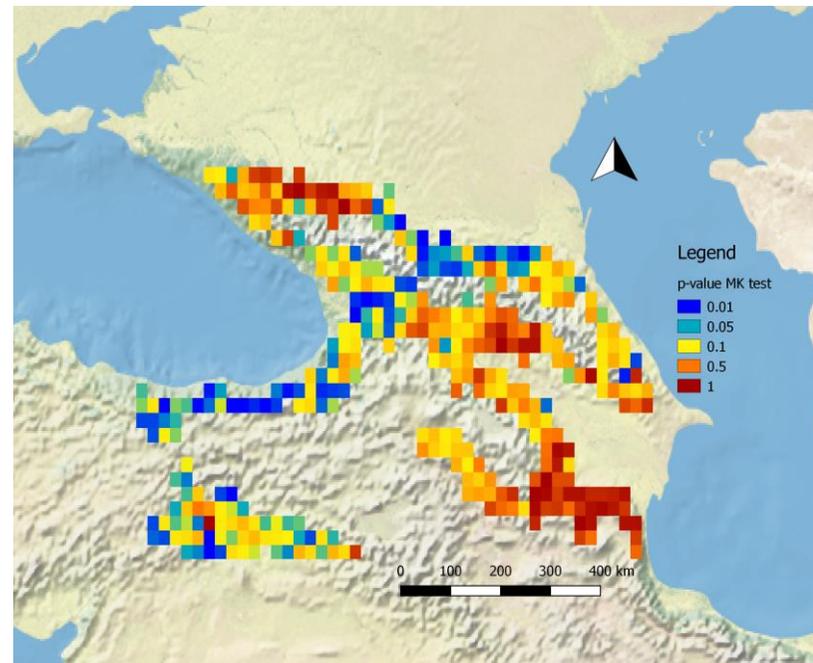
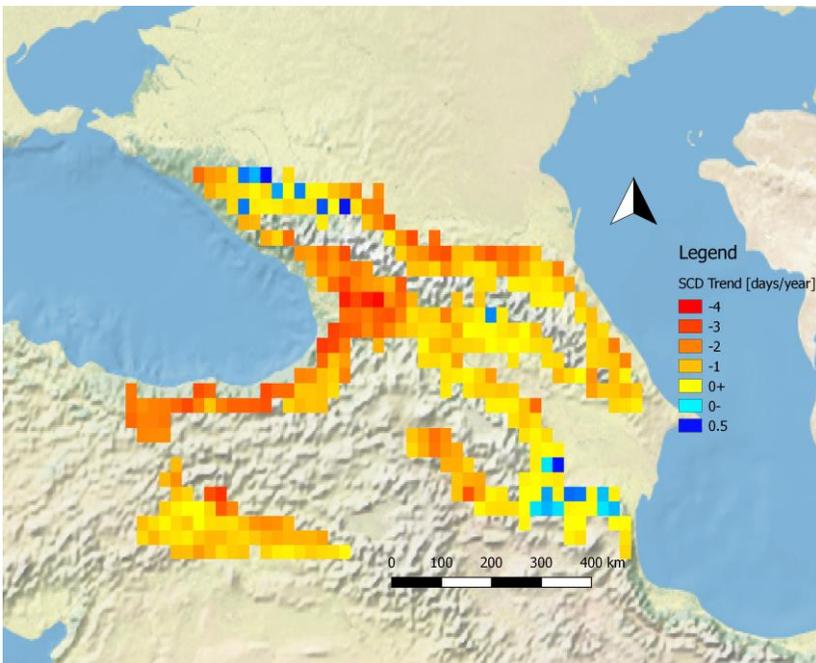


Elevation: 900 m

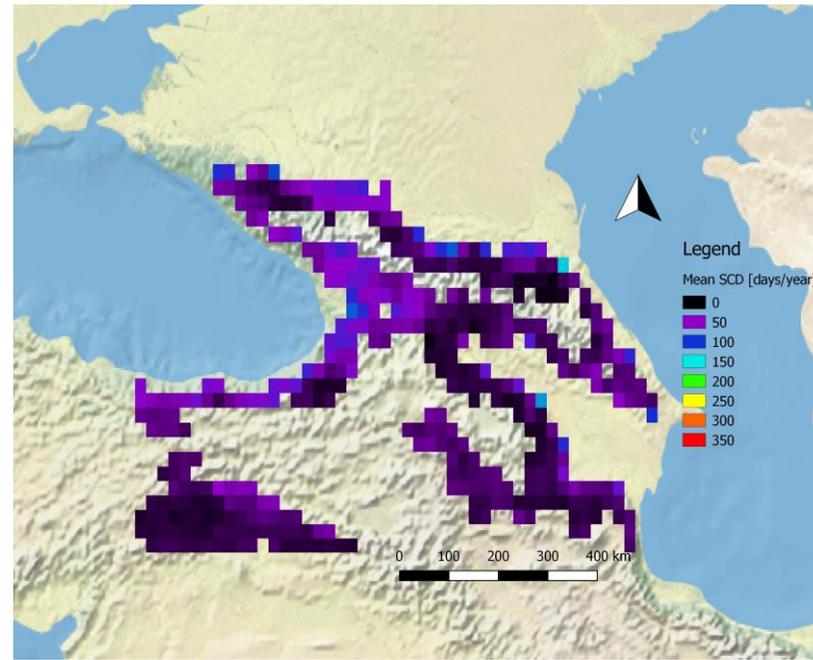
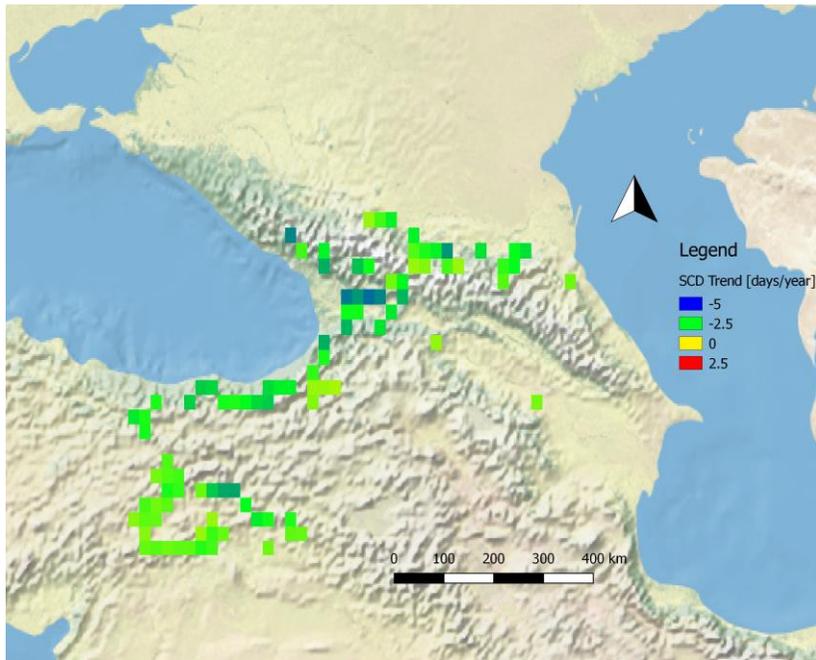
Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

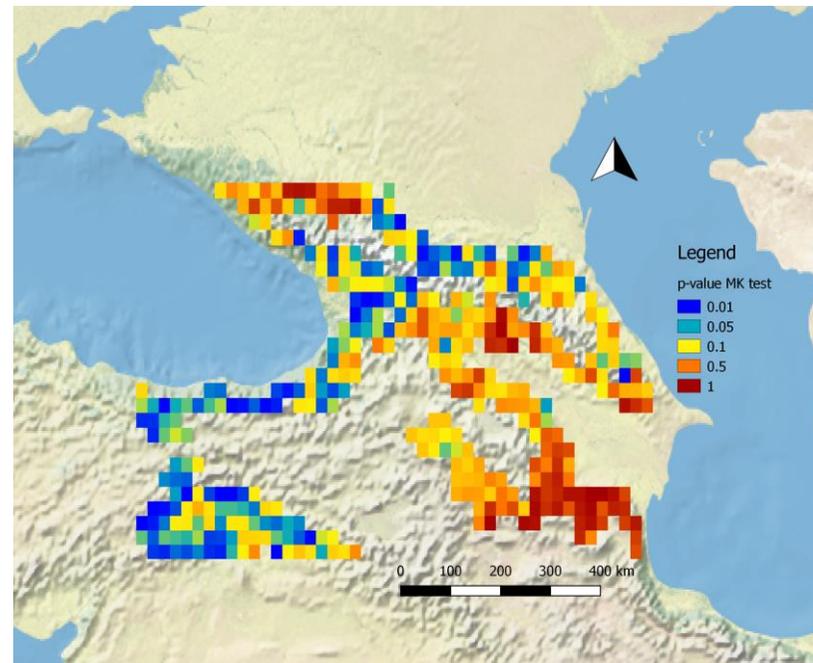
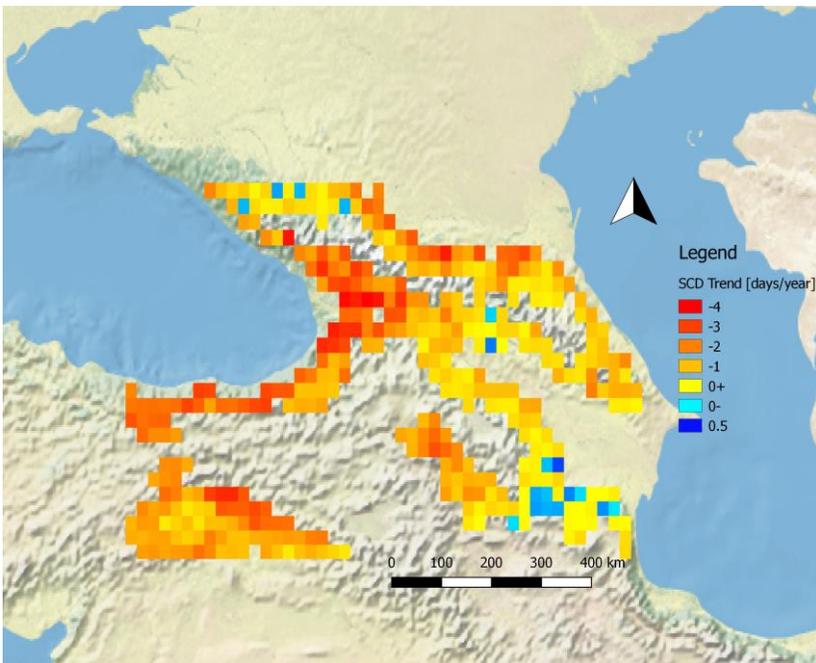
Bottom-Right: p-values

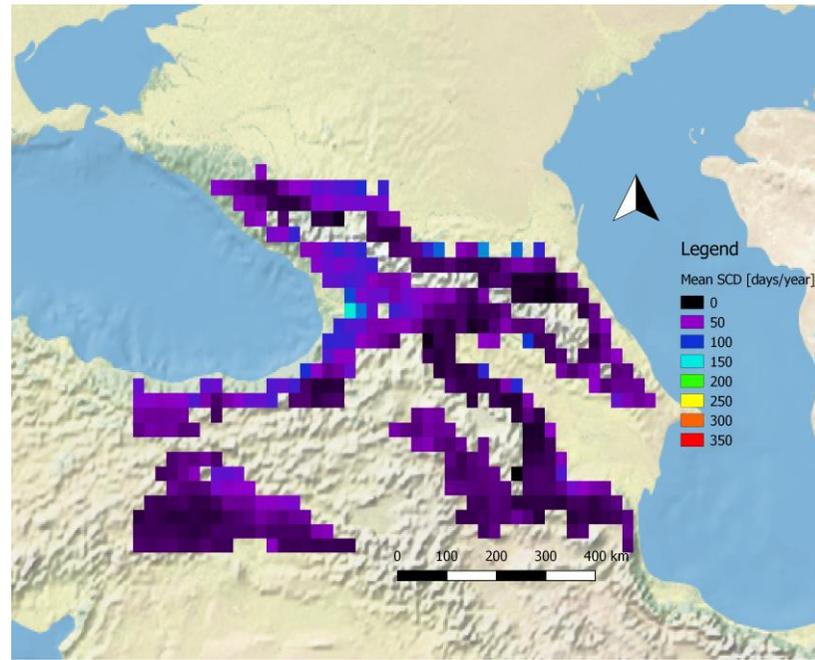
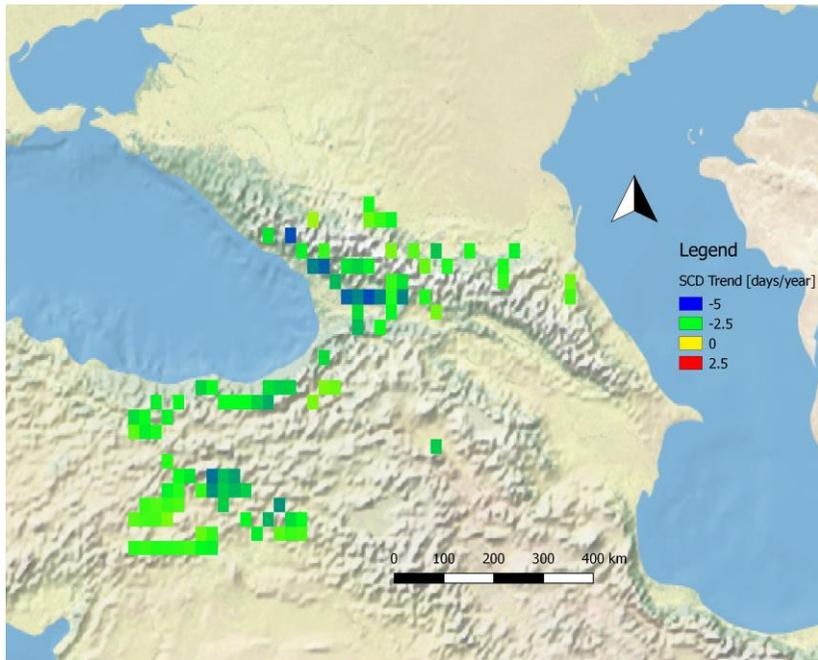


Elevation: 1000 m



Top-Left: significant trends
Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends
Bottom-Right: p-values



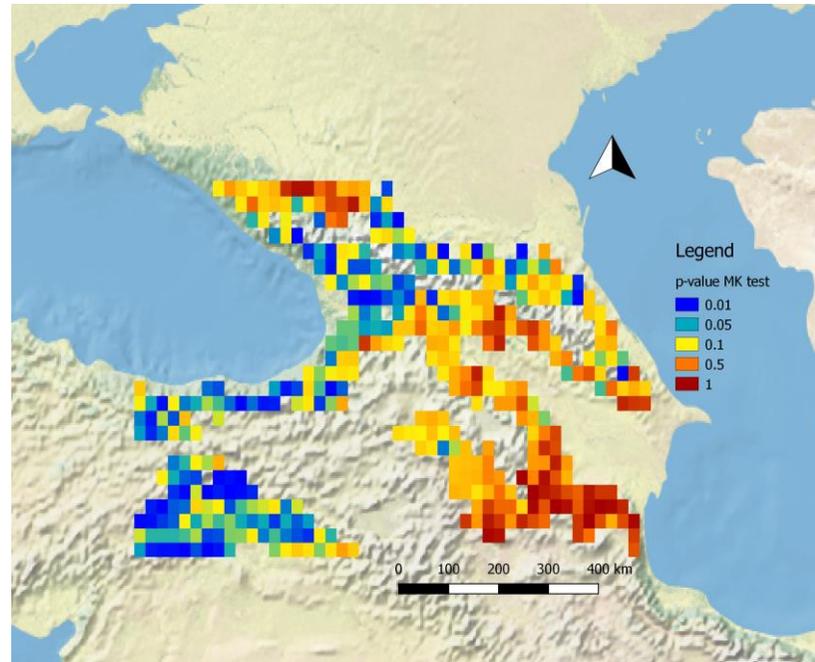
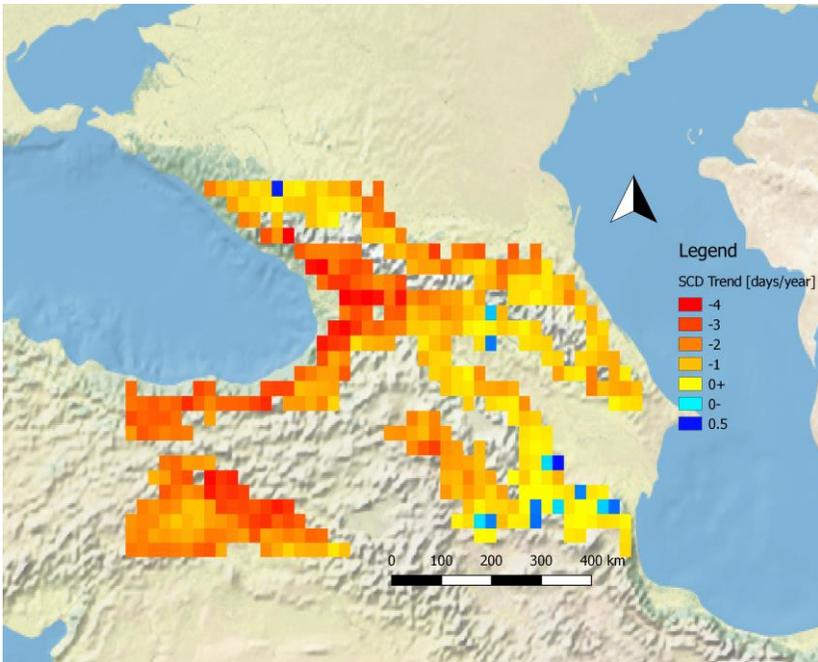


Elevation: 1100 m

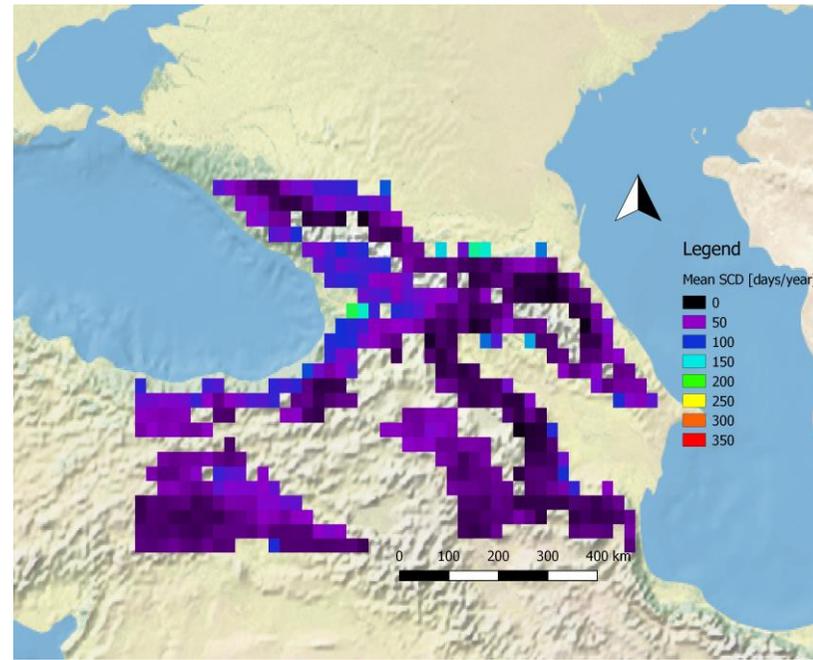
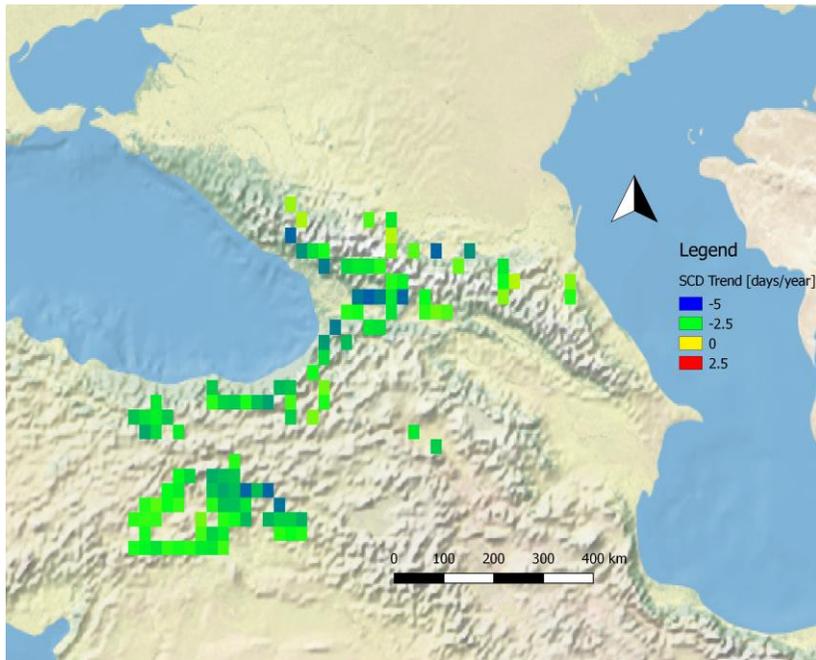
Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

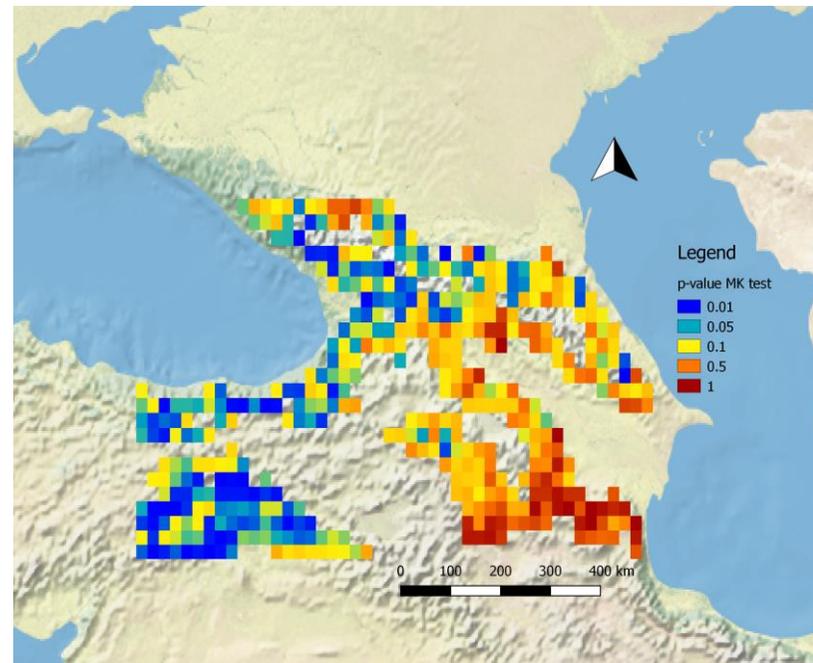
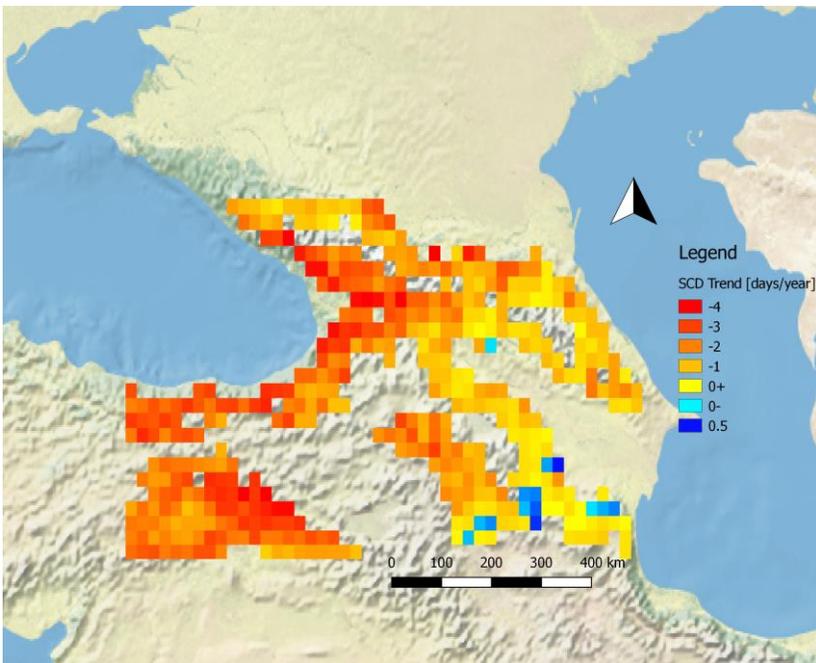
Bottom-Right: p-values



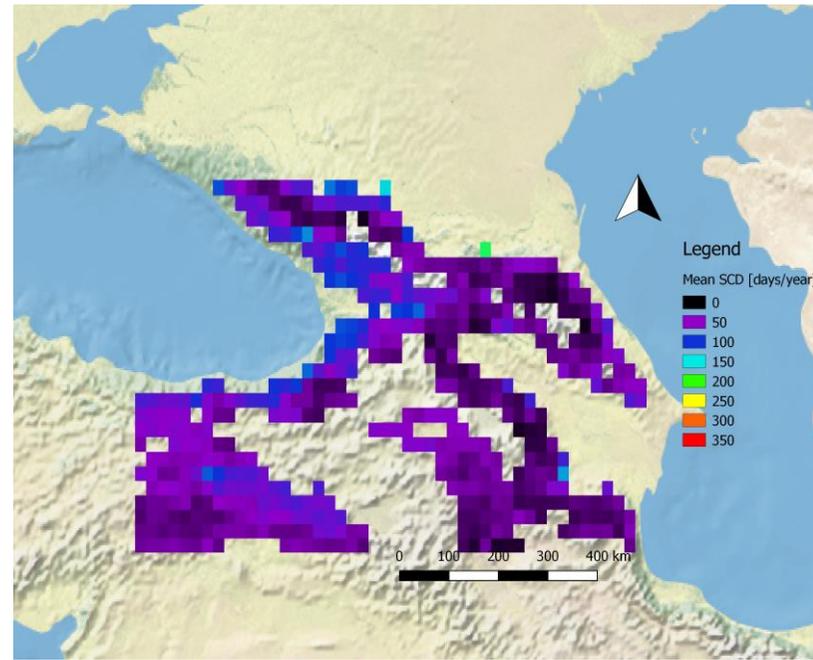
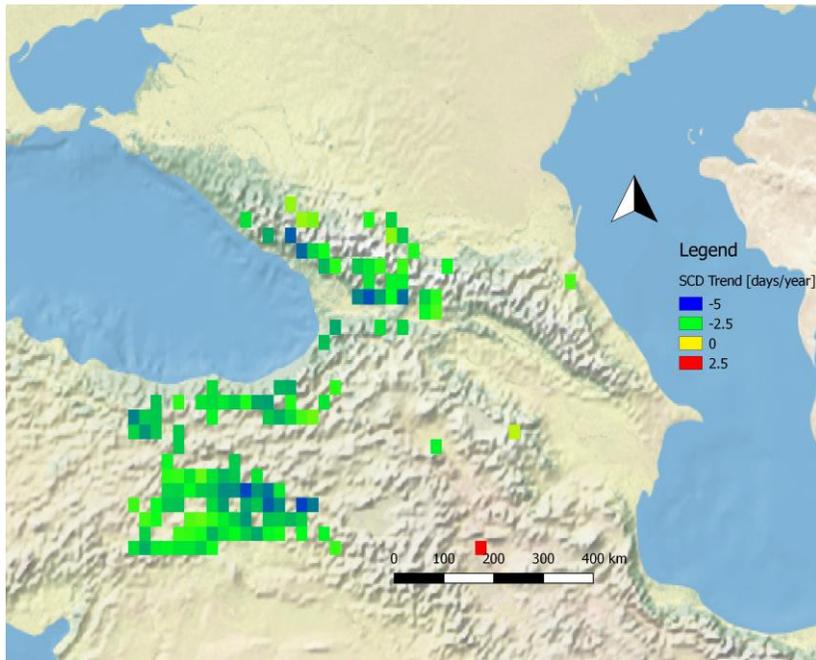
Elevation: 1200 m



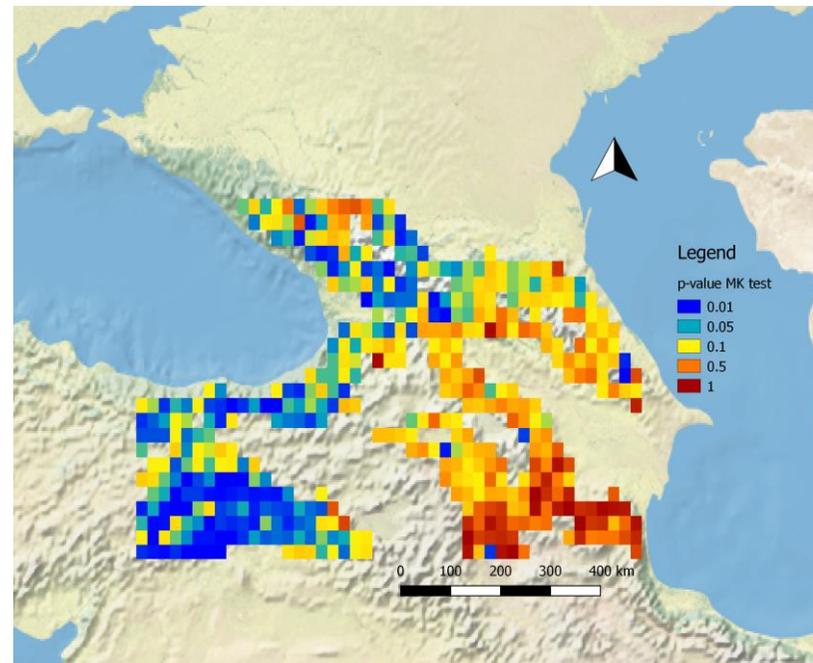
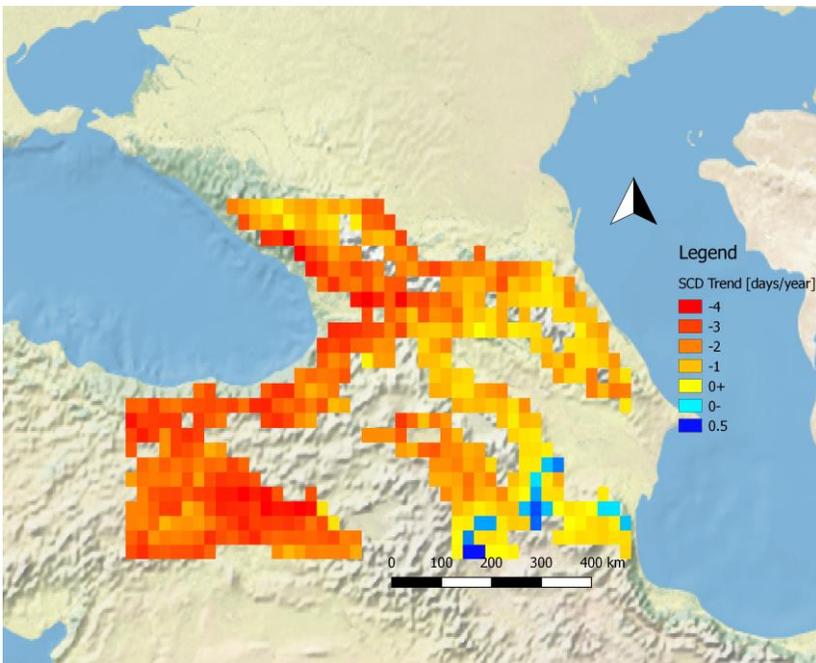
Top-Left: significant trends
Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends
Bottom-Right: p-values



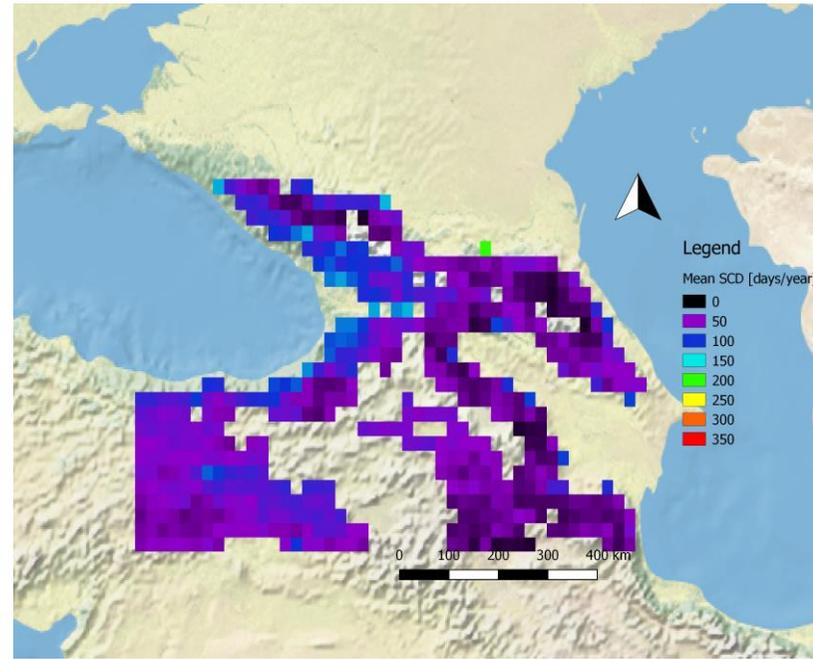
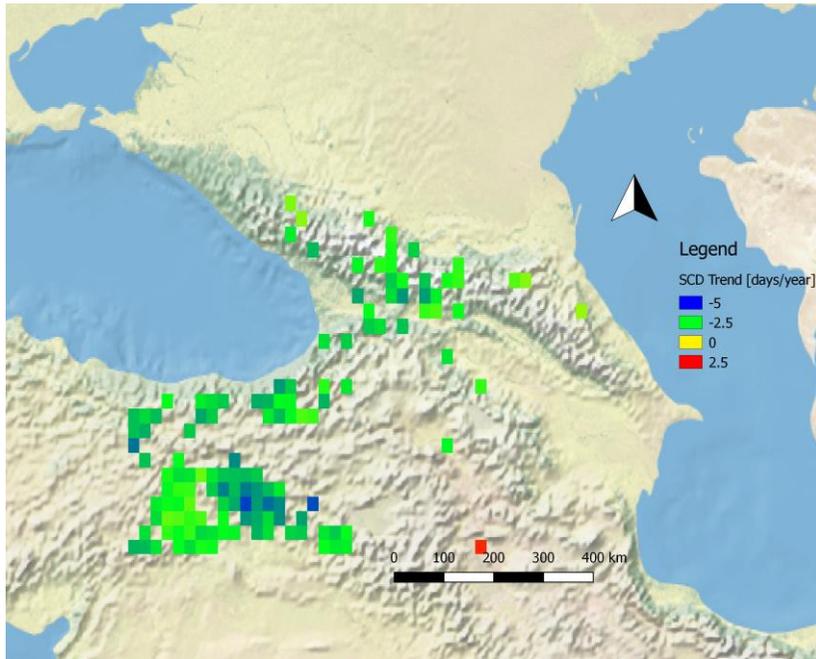
Elevation: 1300 m



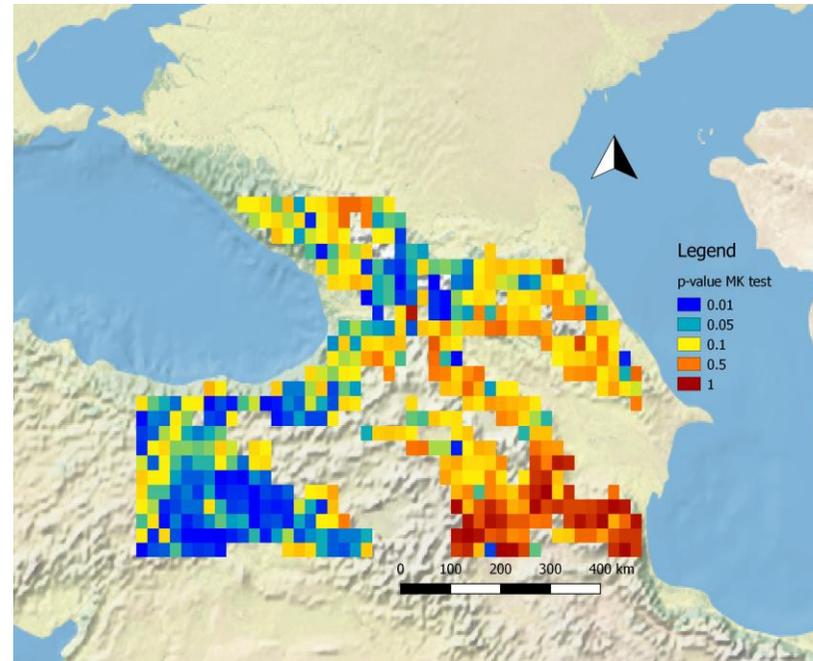
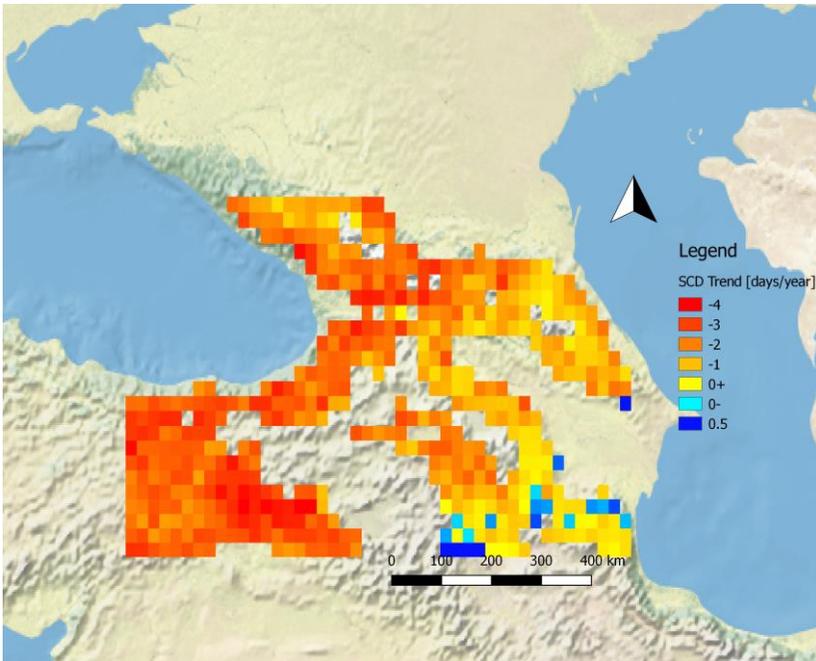
Top-Left: significant trends
Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends
Bottom-Right: p-values



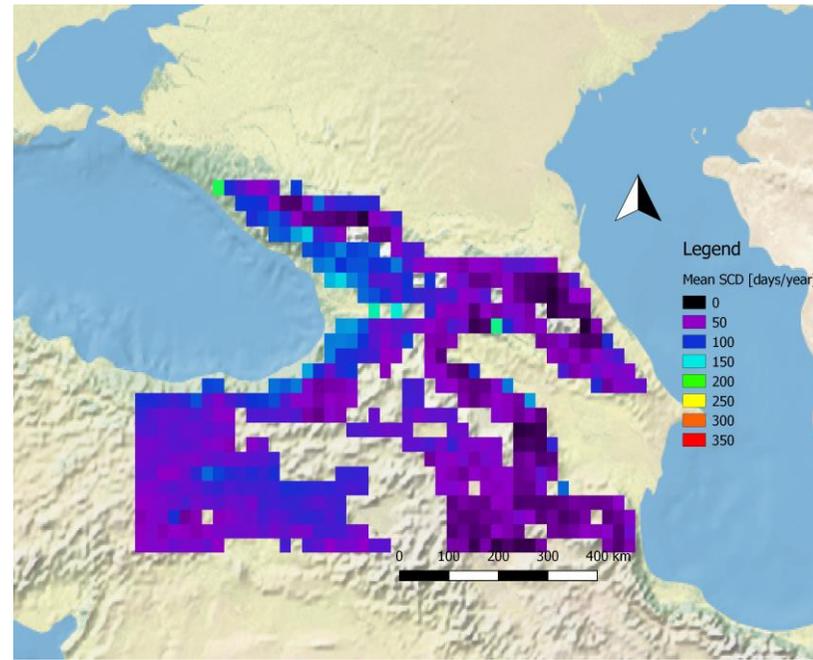
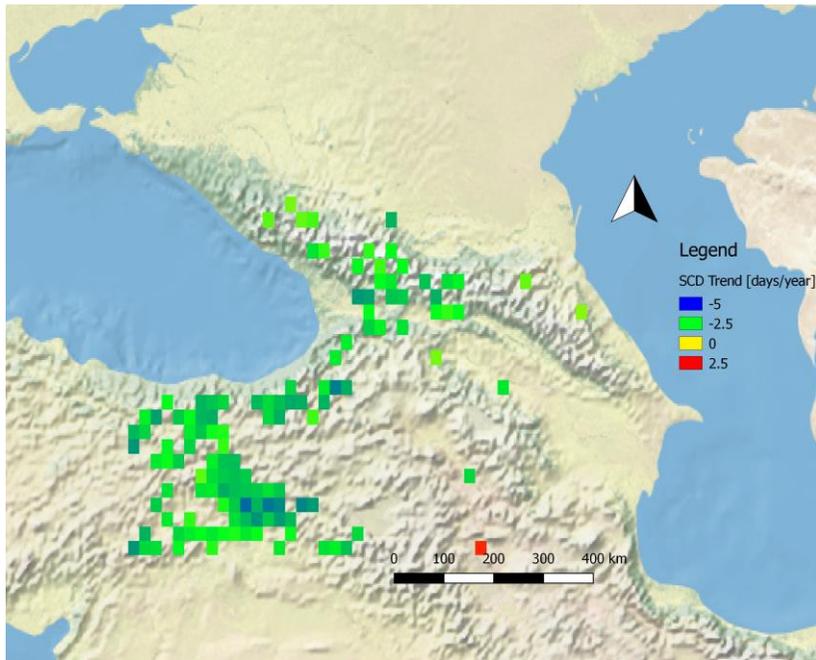
Elevation: 1400 m



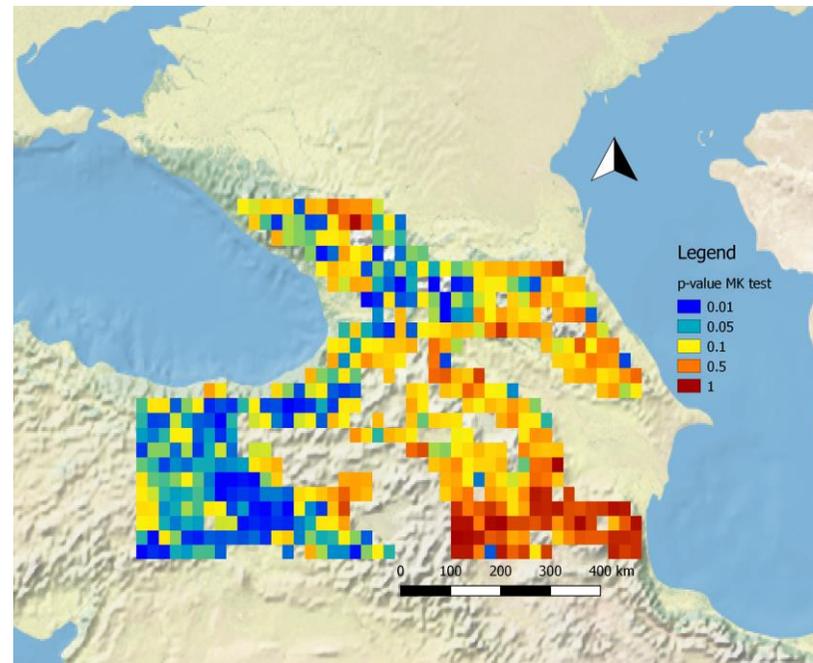
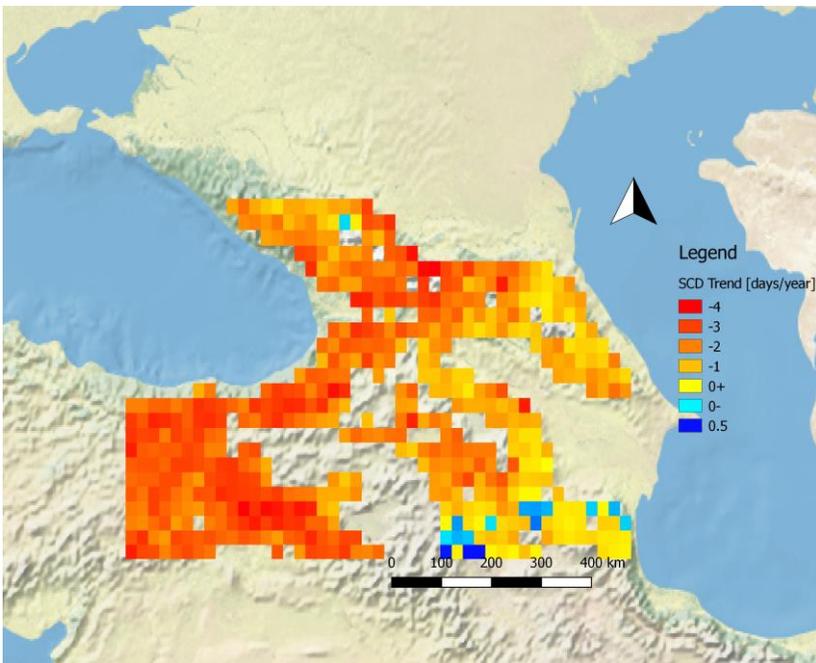
Top-Left: significant trends
Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends
Bottom-Right: p-values



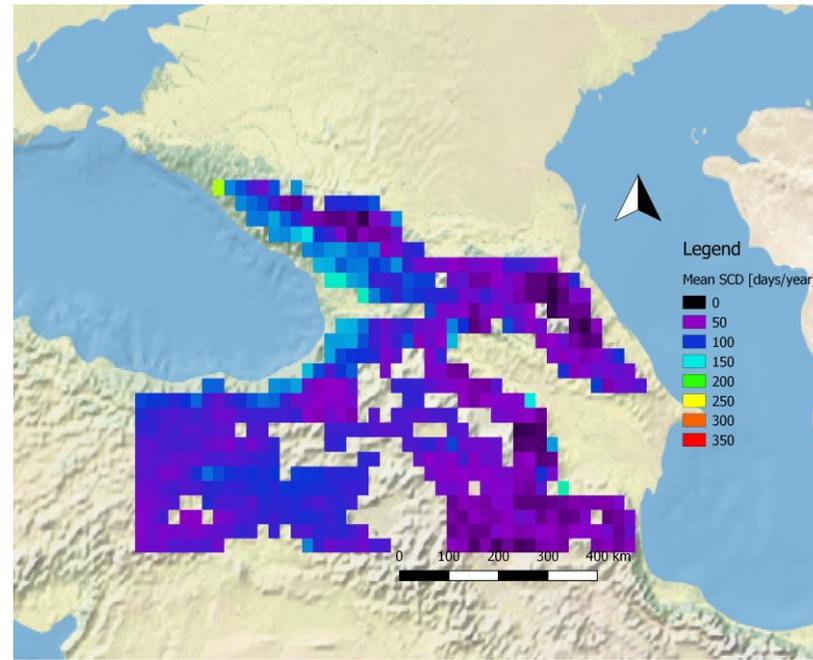
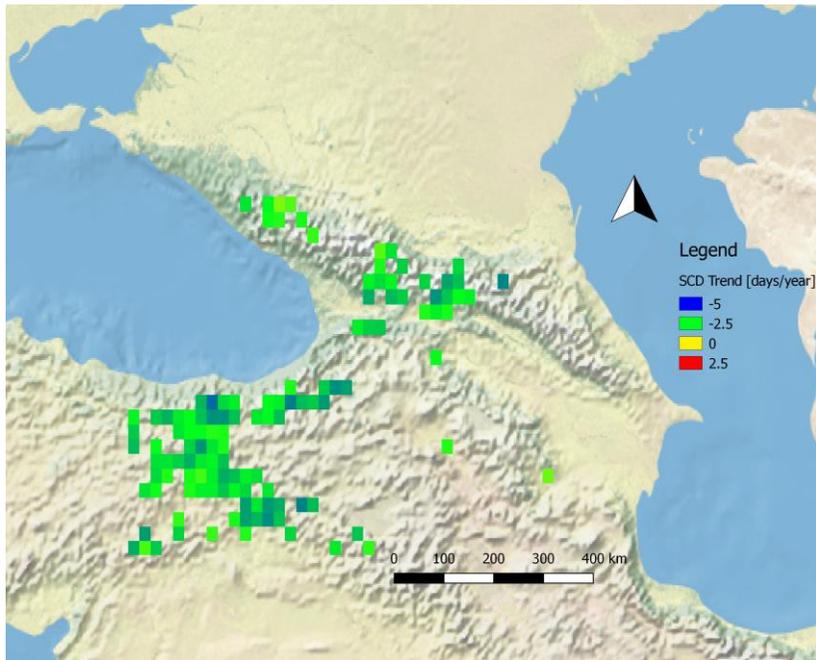
Elevation: 1500 m



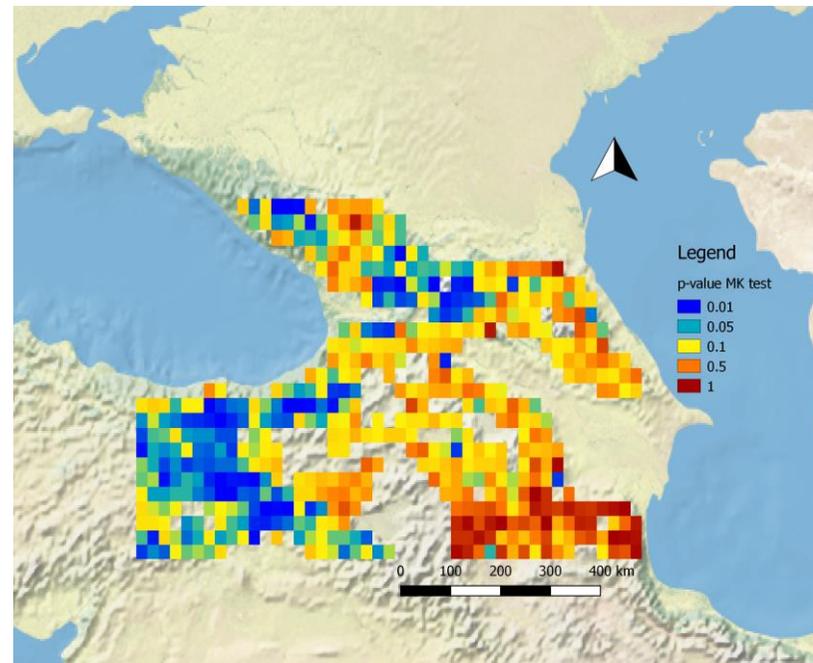
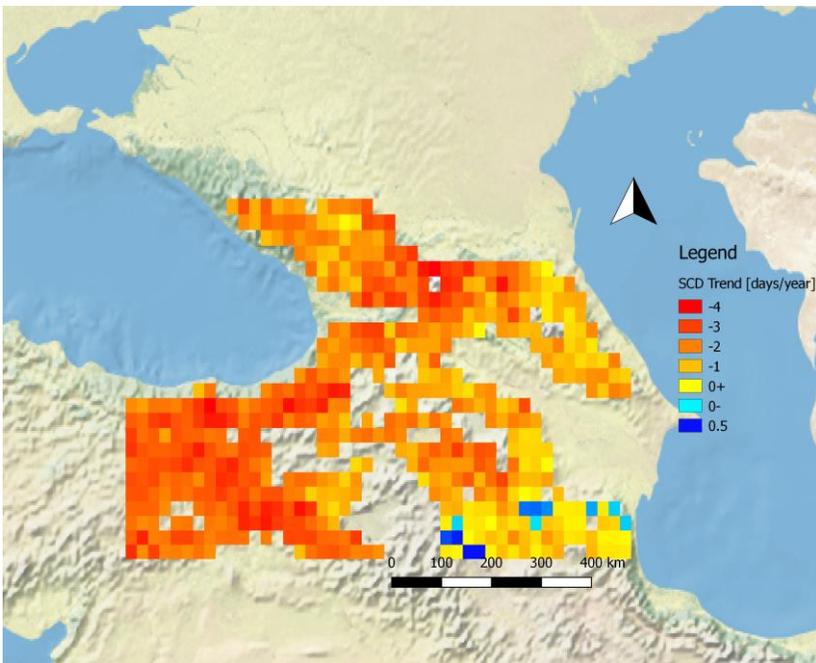
Top-Left: significant trends
Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends
Bottom-Right: p-values

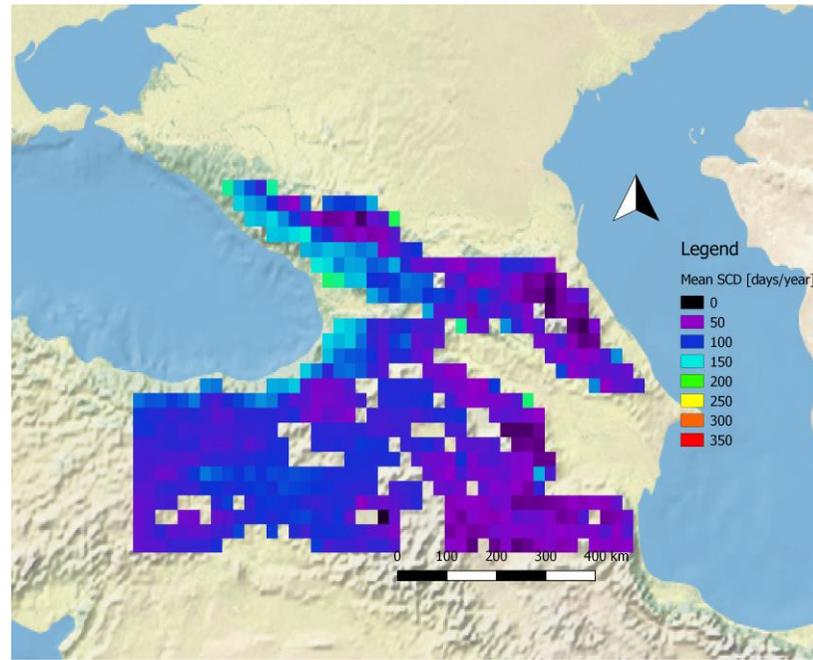
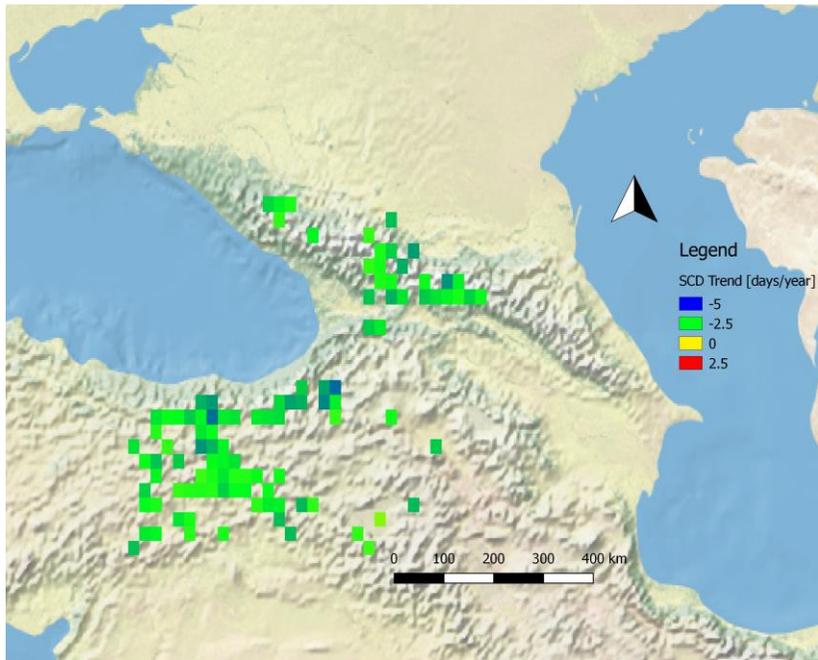


Elevation: 1600 m



Top-Left: significant trends
Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends
Bottom-Right: p-values



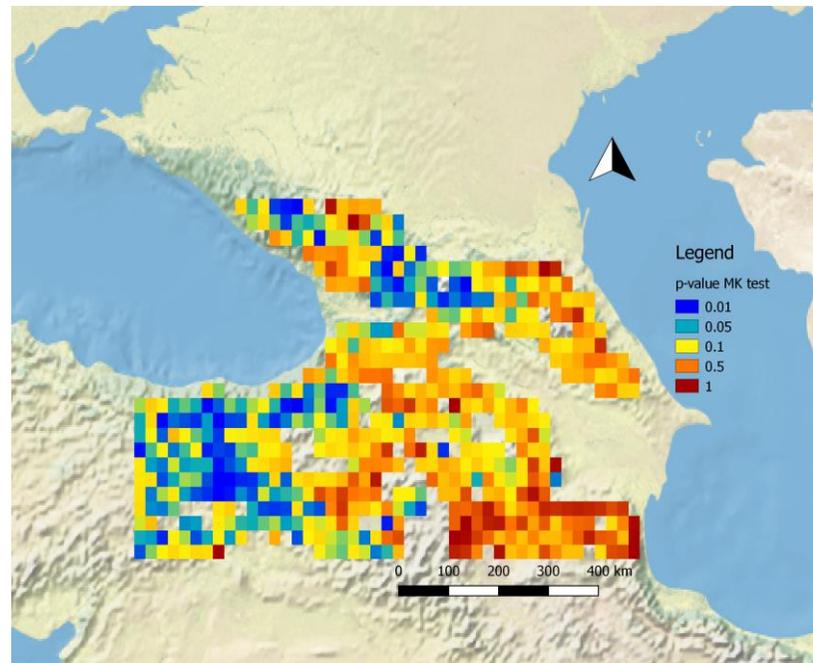
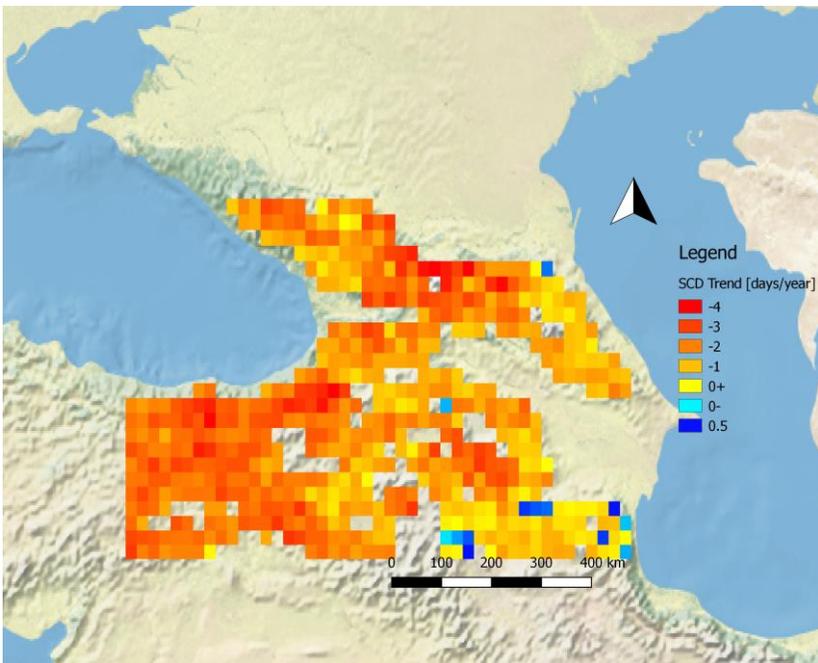


Elevation: 1700 m

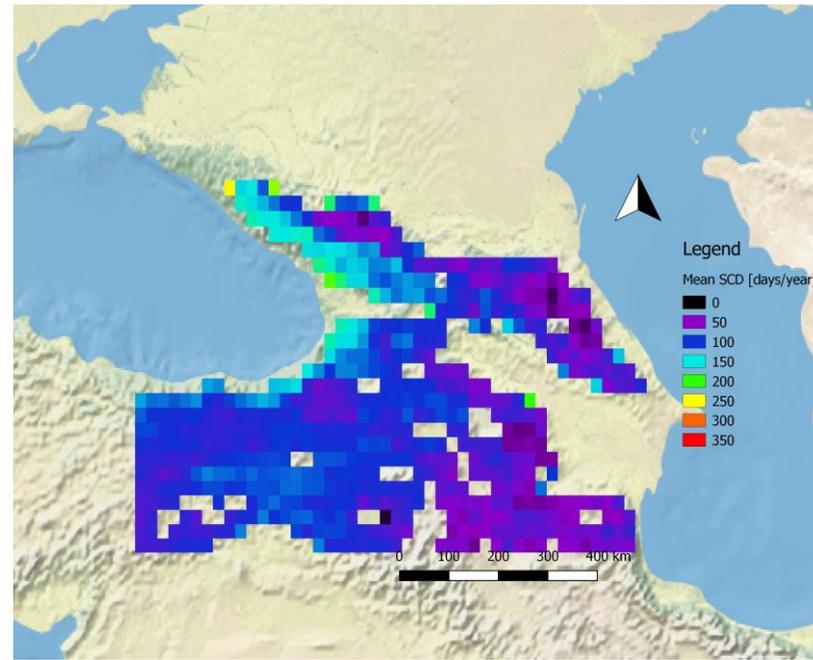
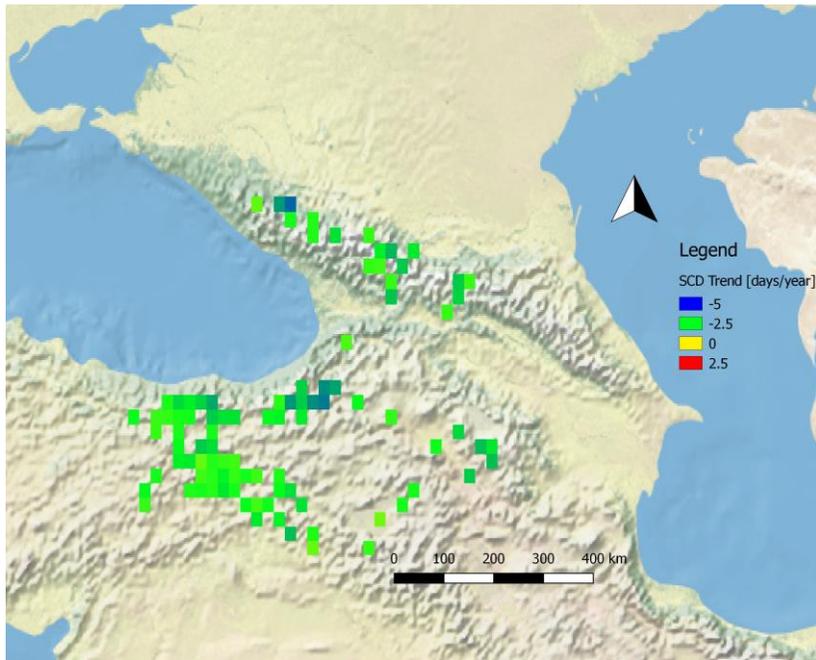
Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

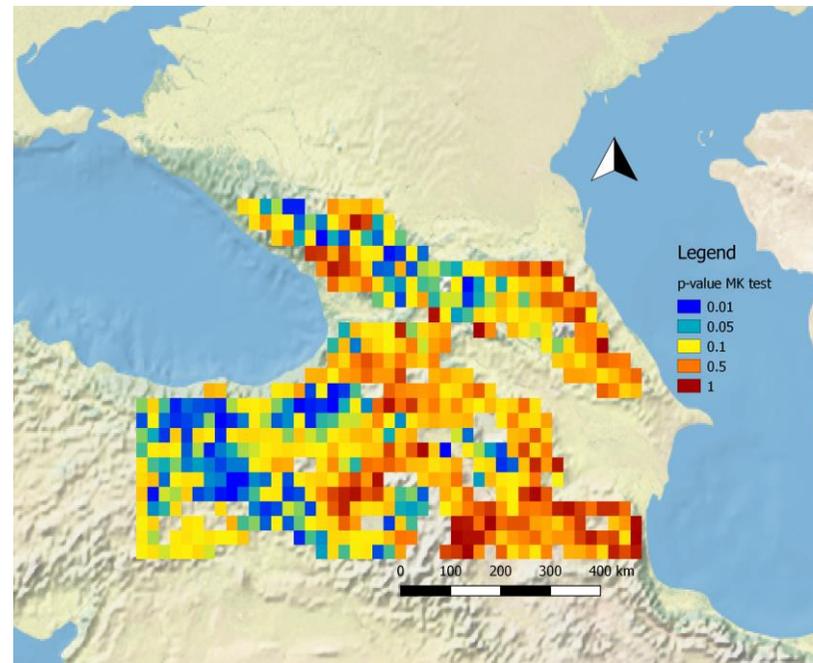
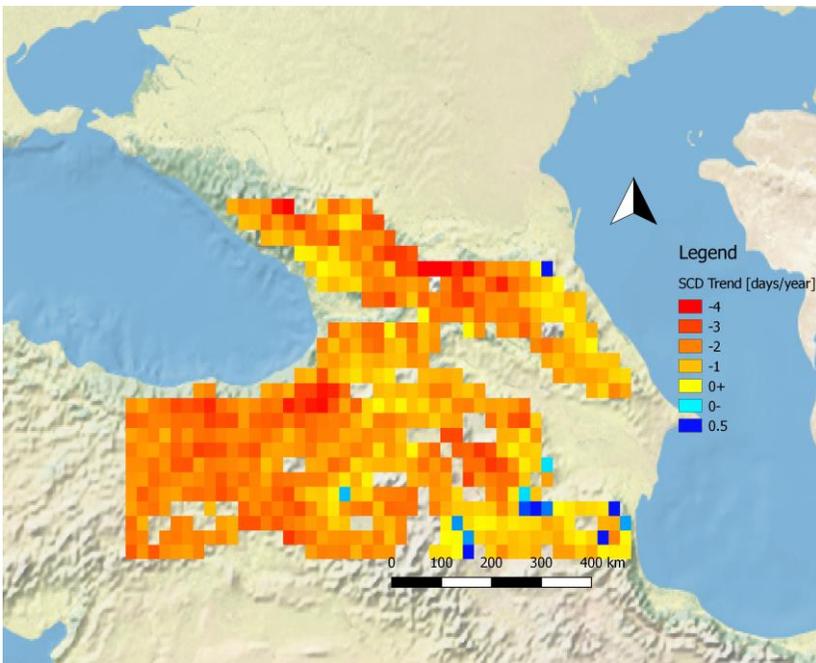
Bottom-Right: p-values

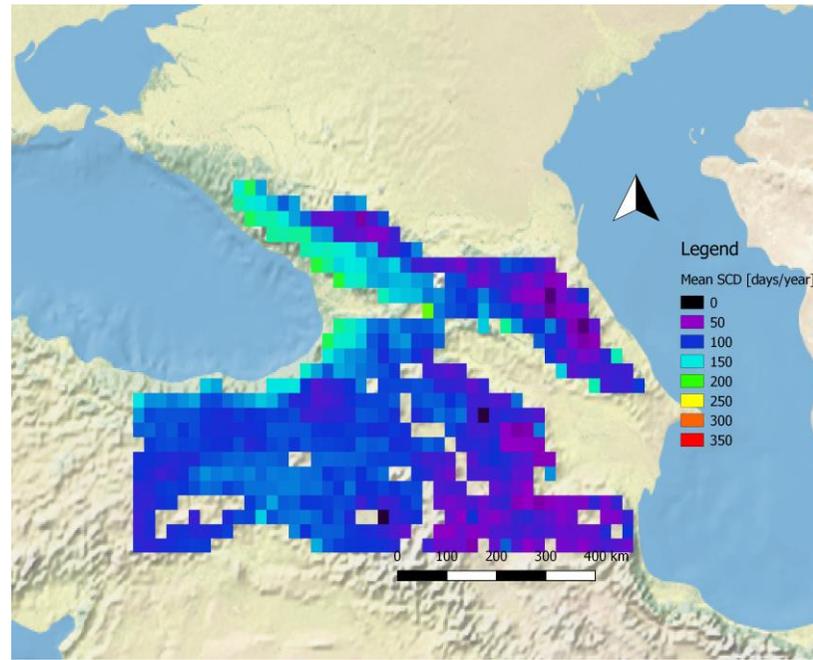
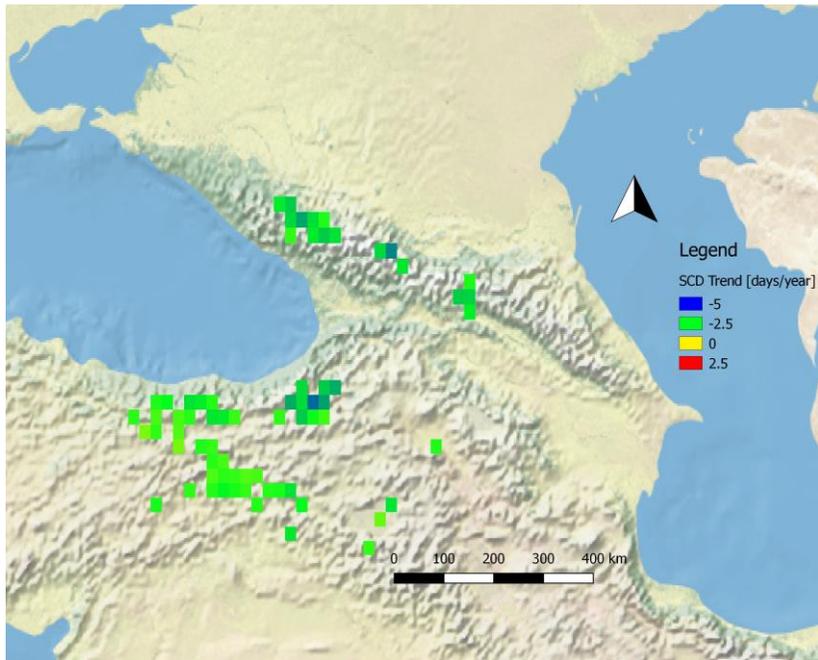


Elevation: 1800 m



Top-Left: significant trends
Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends
Bottom-Right: p-values



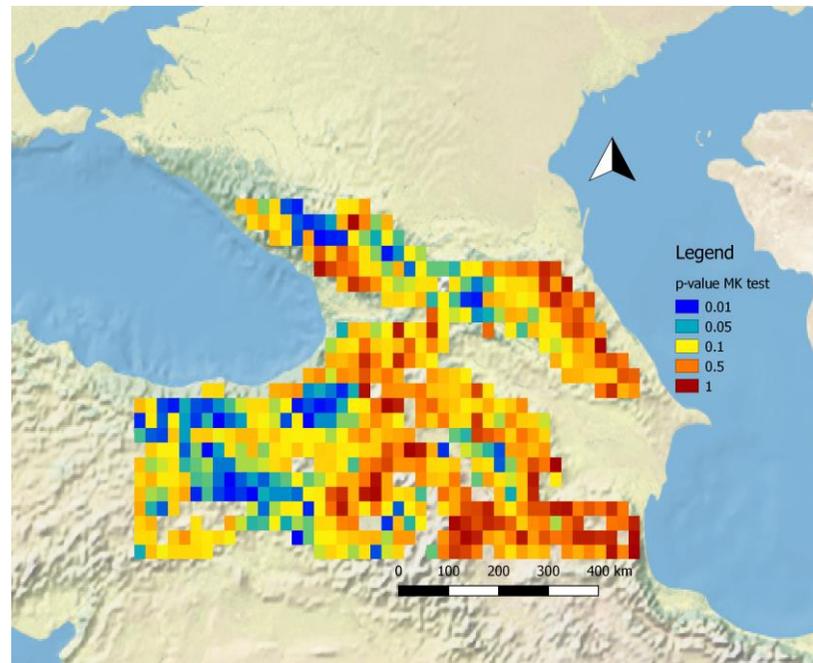
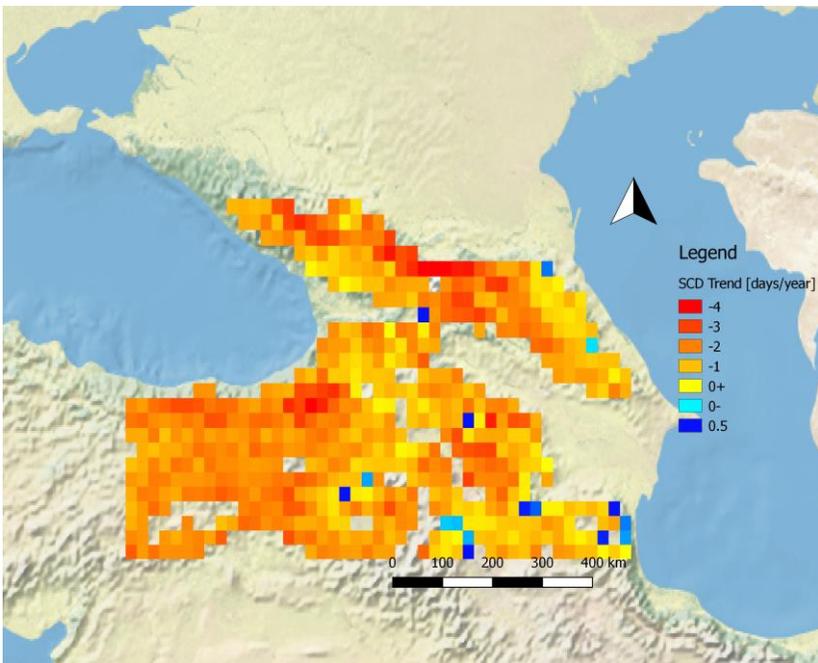


Elevation: 1900 m

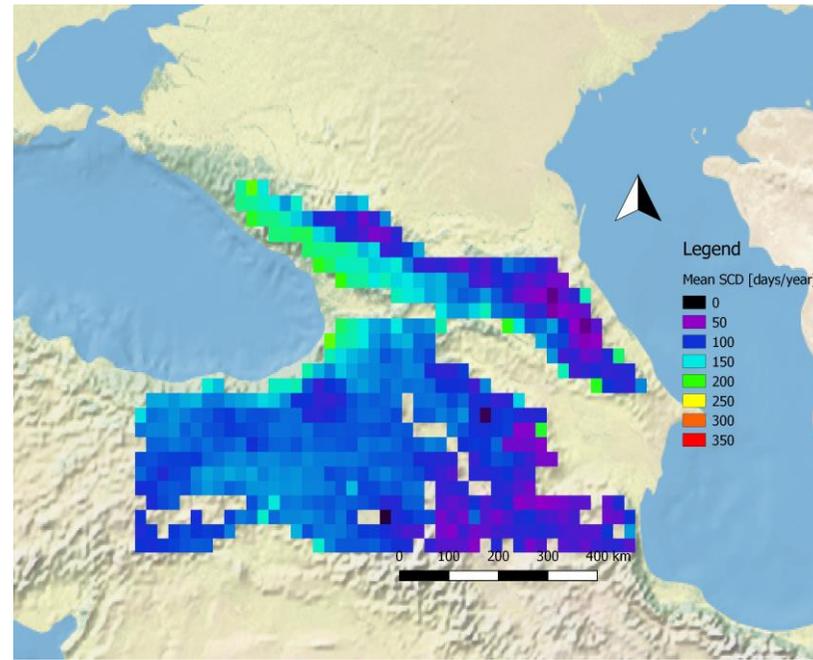
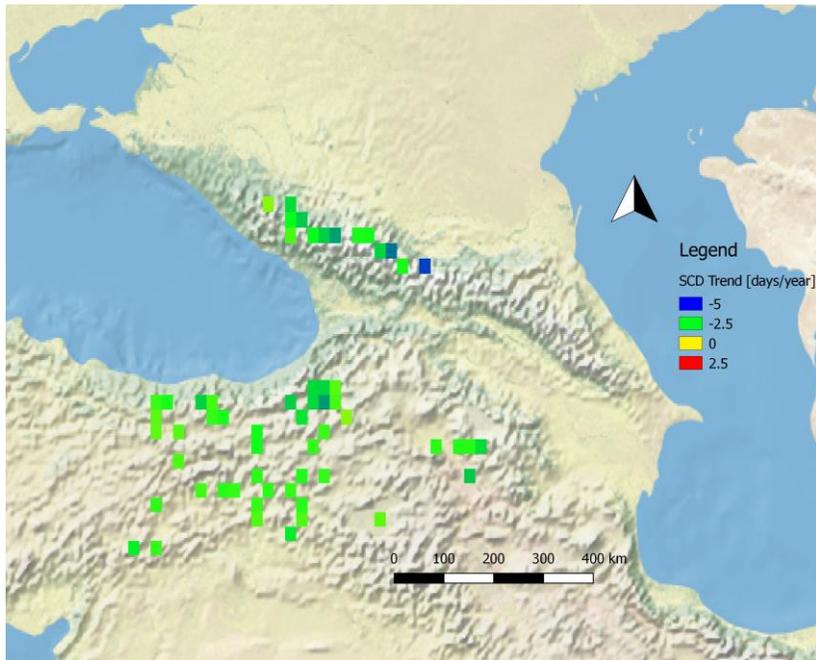
Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

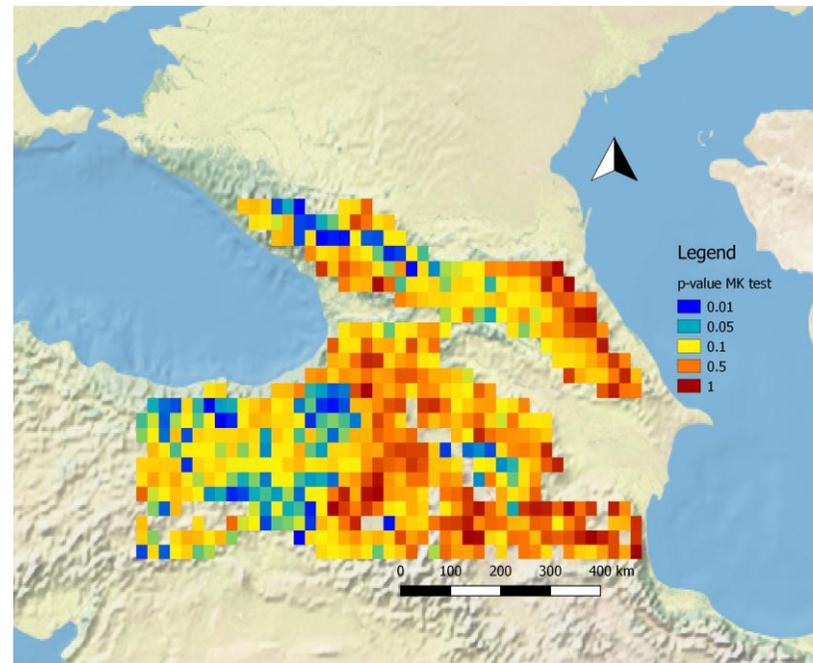
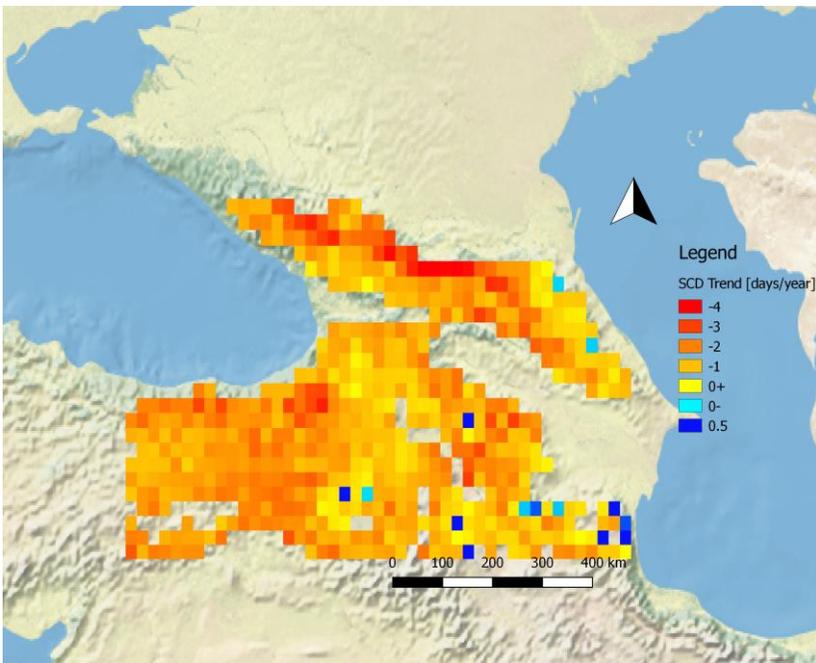
Bottom-Right: p-values



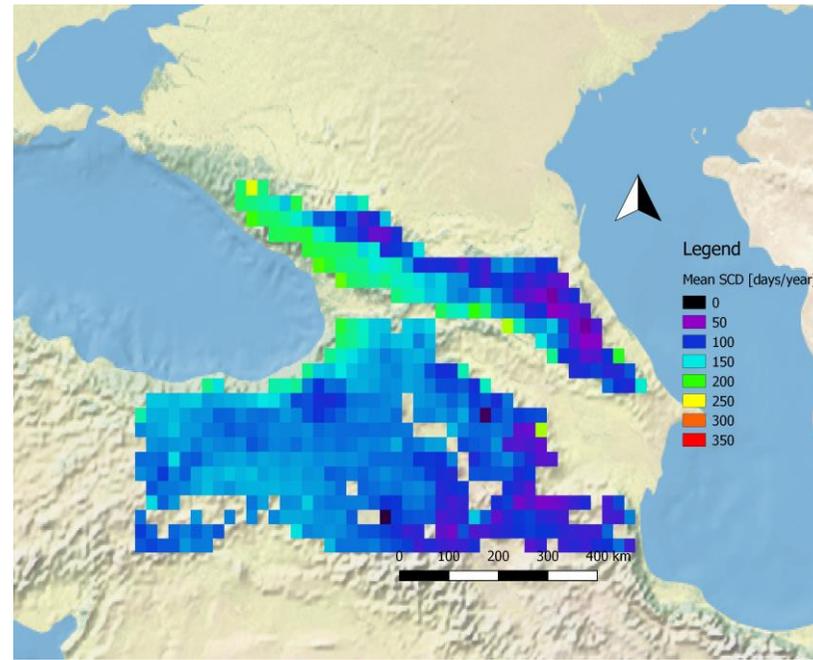
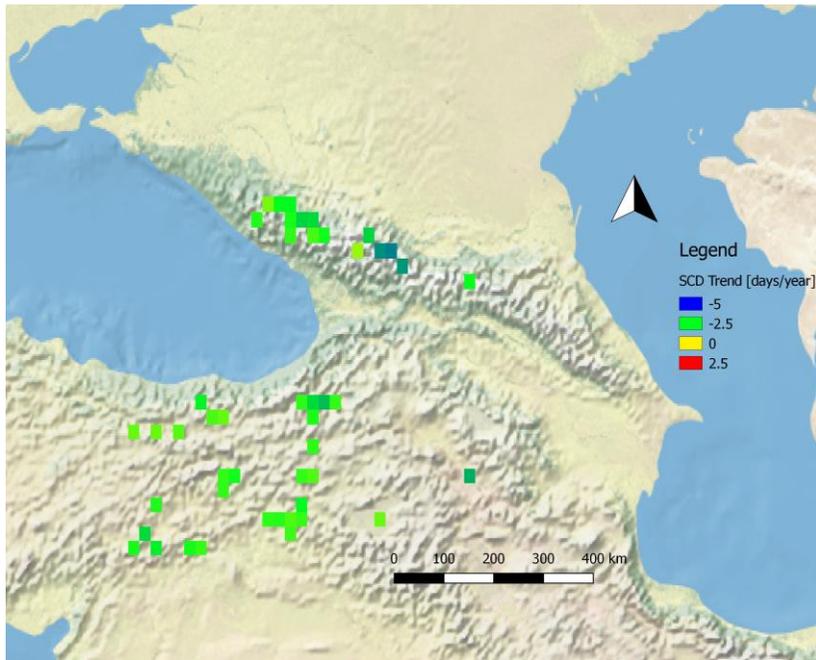
Elevation: 2000 m



Top-Left: significant trends
Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends
Bottom-Right: p-values



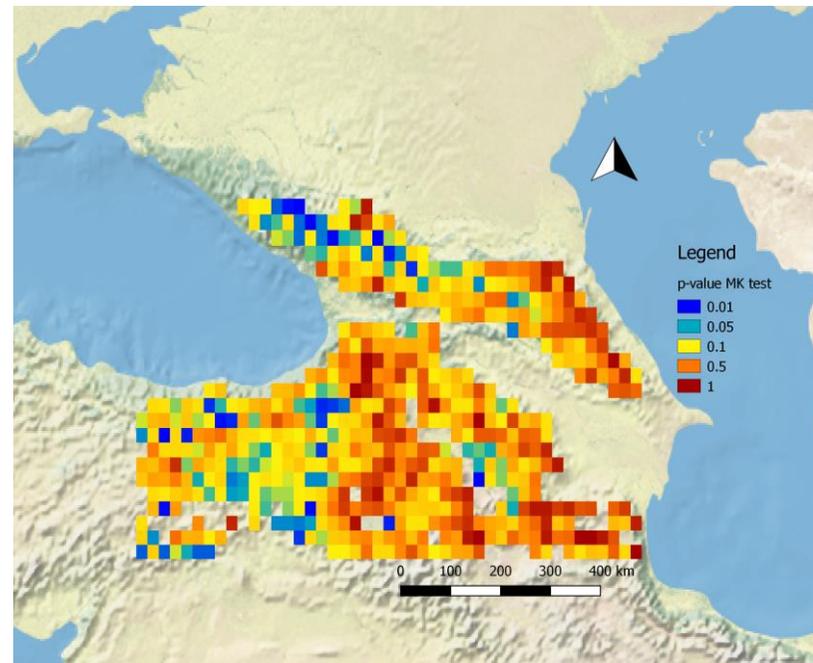
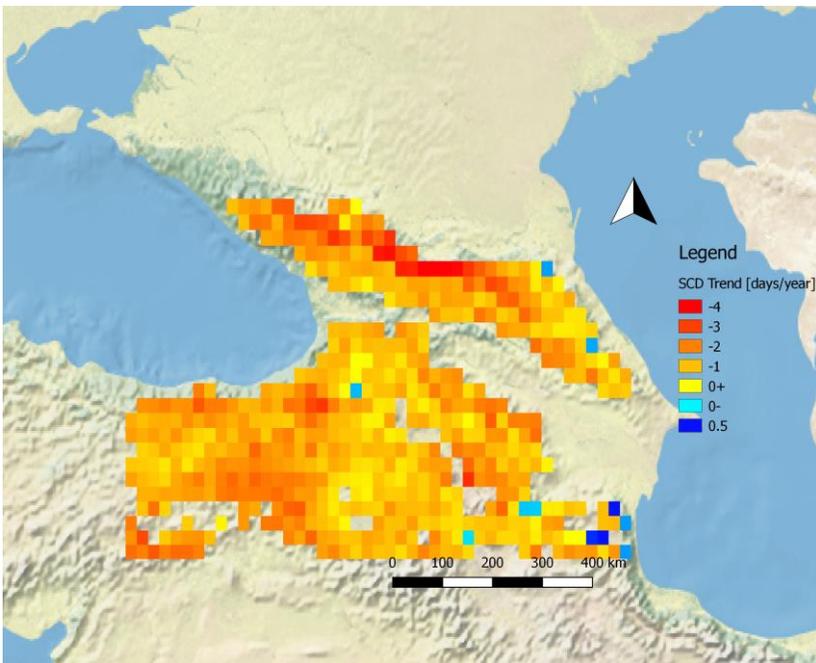
Elevation: 2100 m

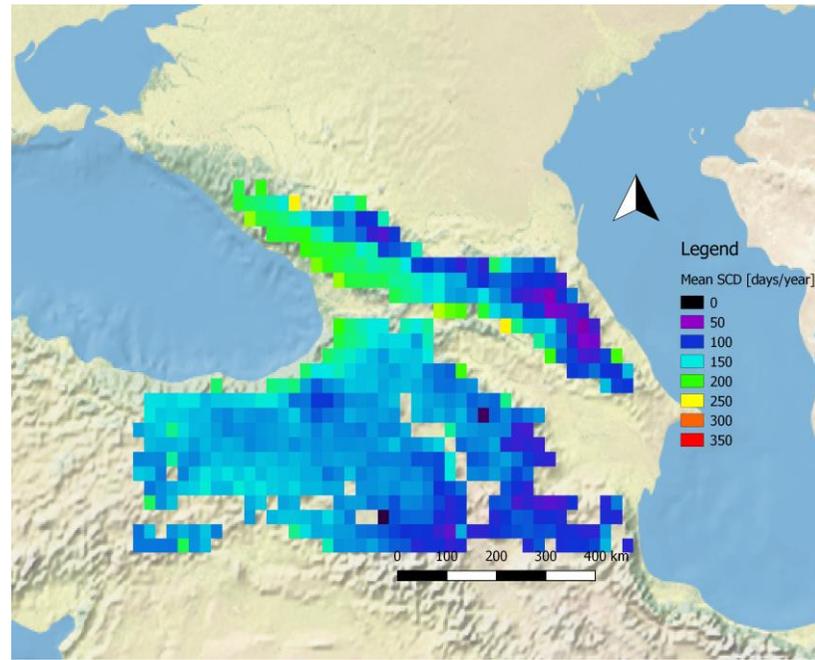
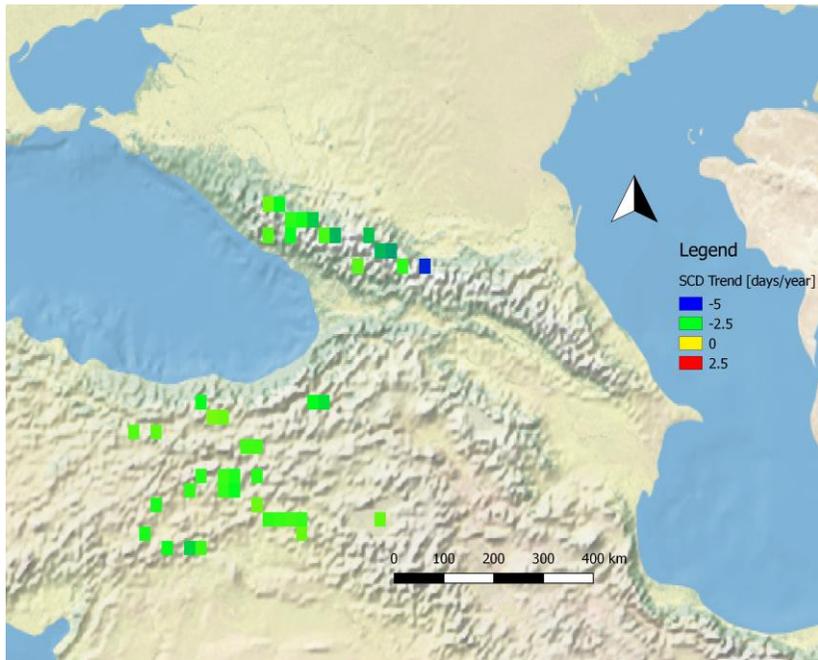


Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

Bottom-Right: p-values



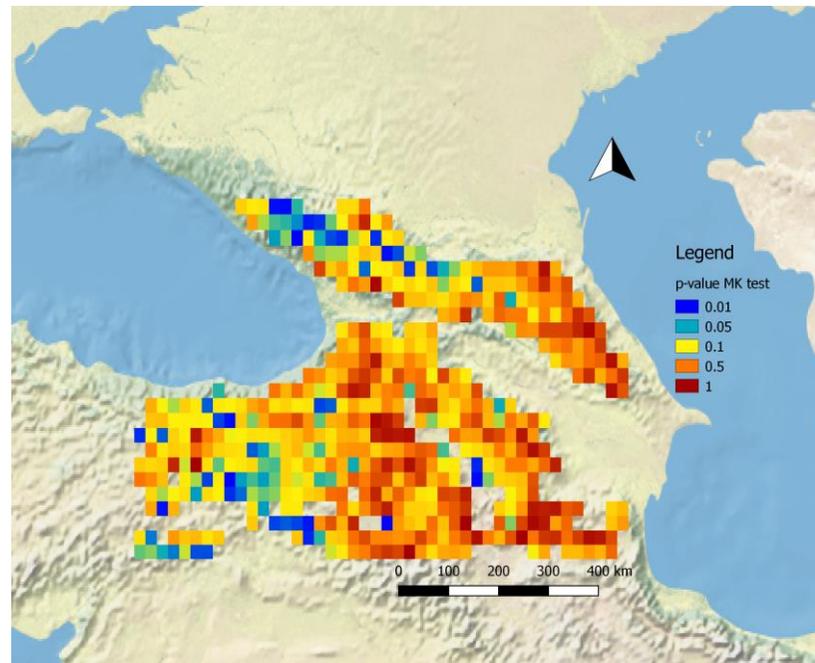
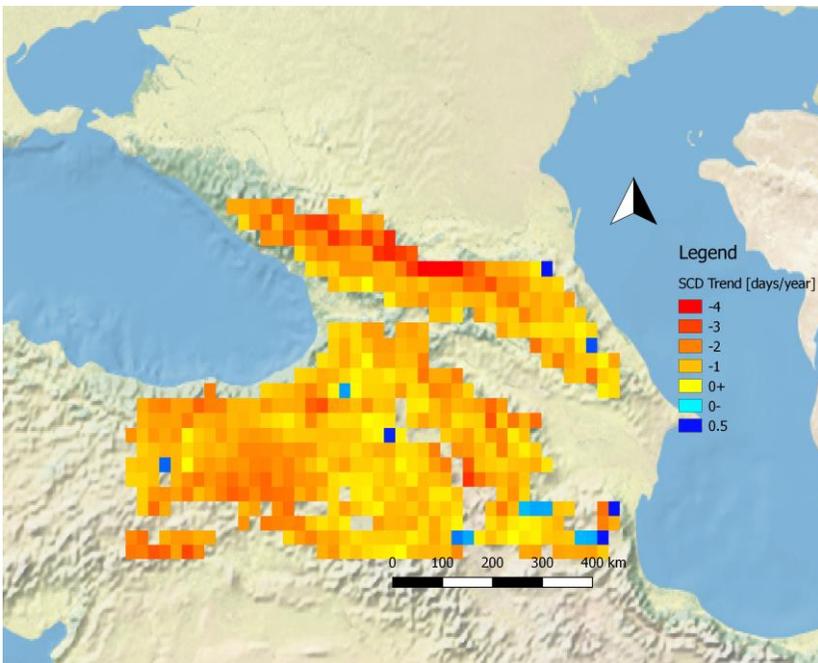


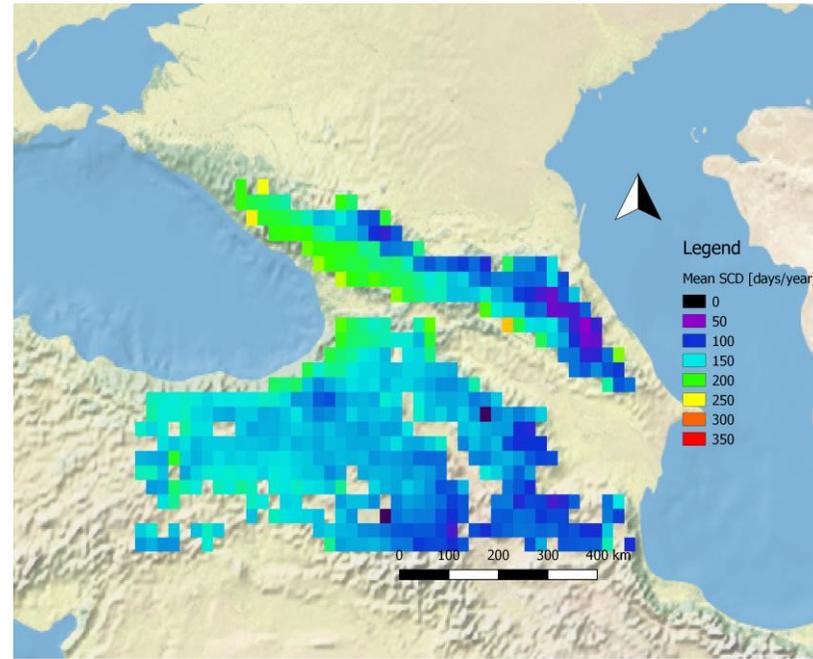
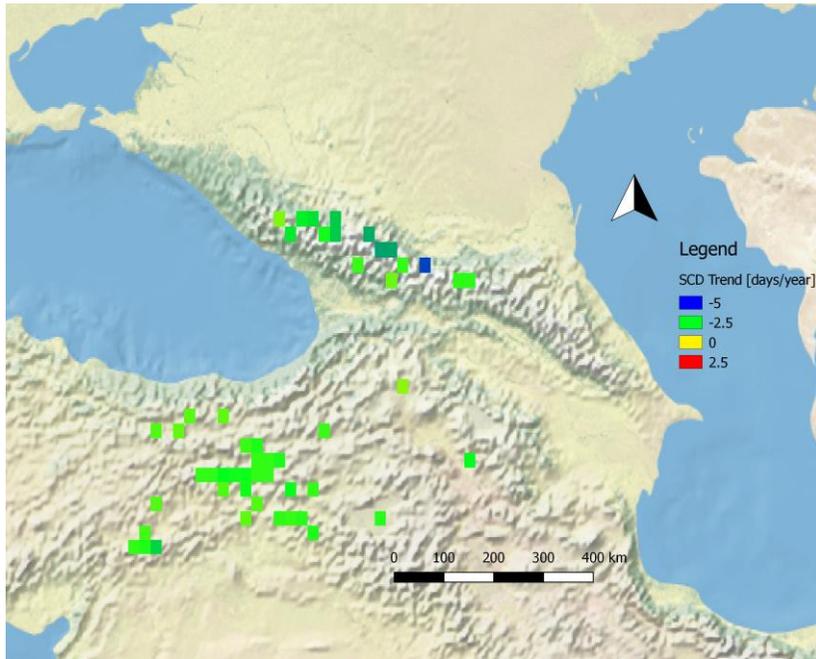
Elevation: 2200 m

Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

Bottom-Right: p-values



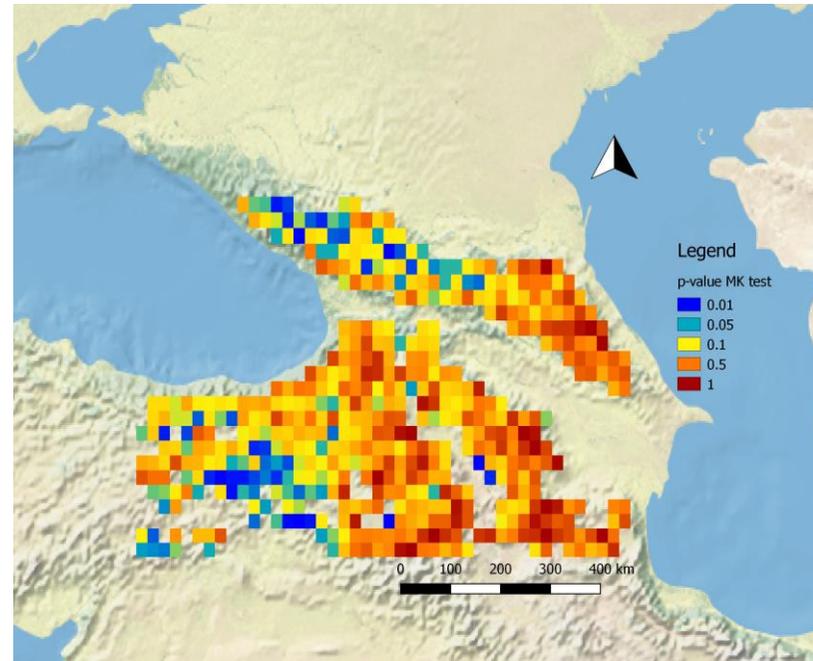
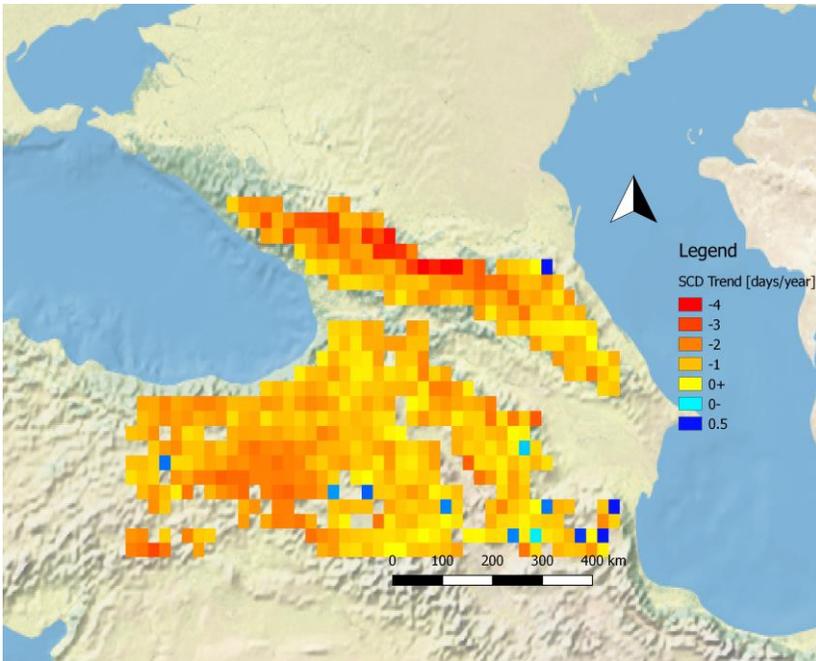


Elevation: 2300 m

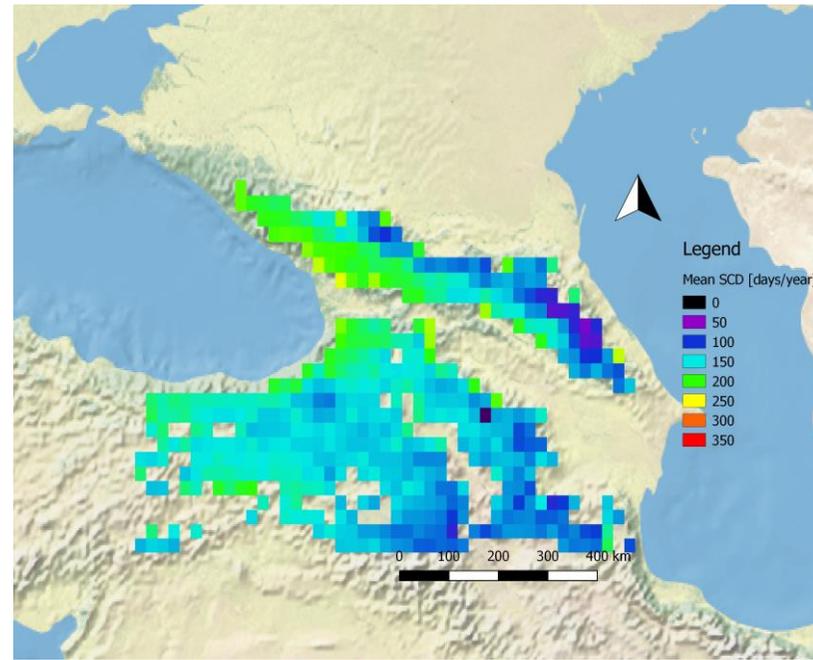
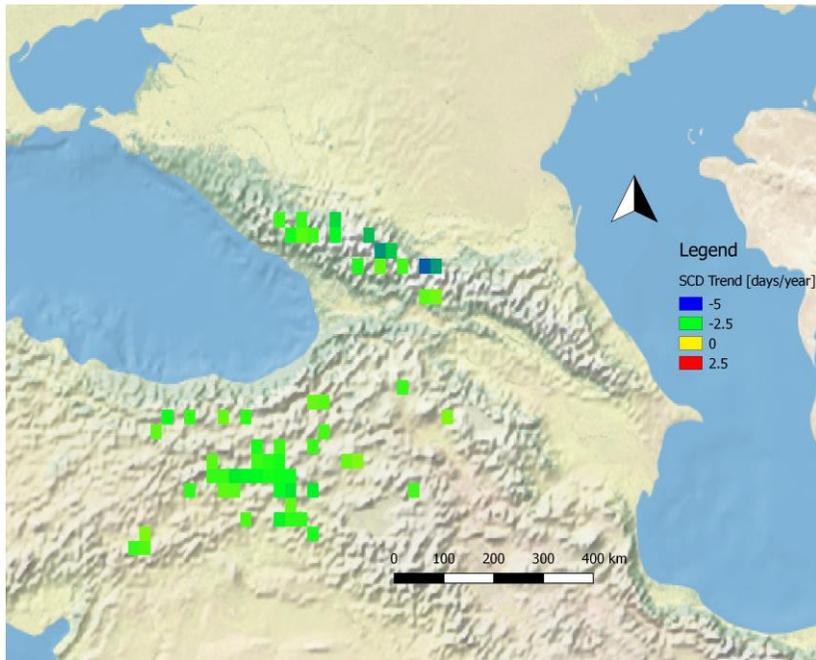
Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

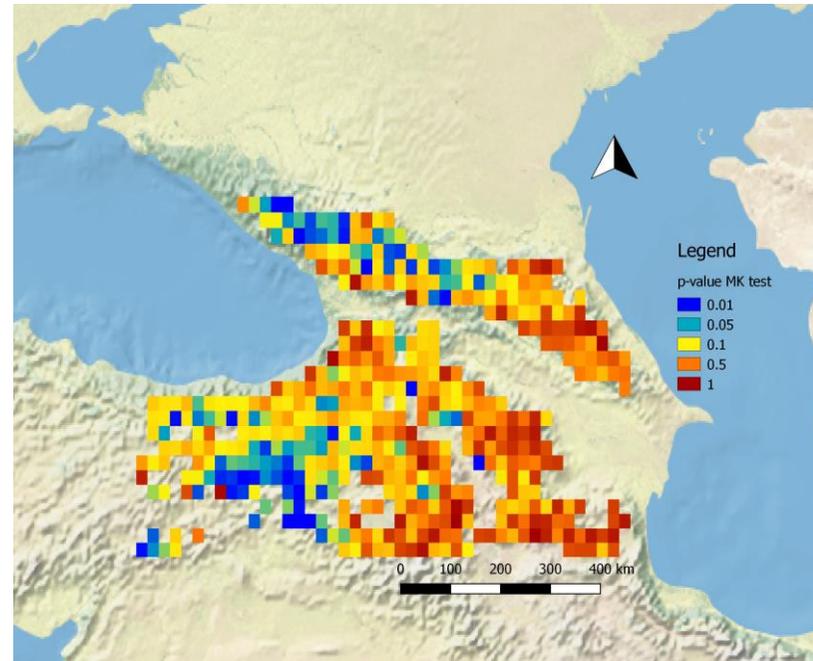
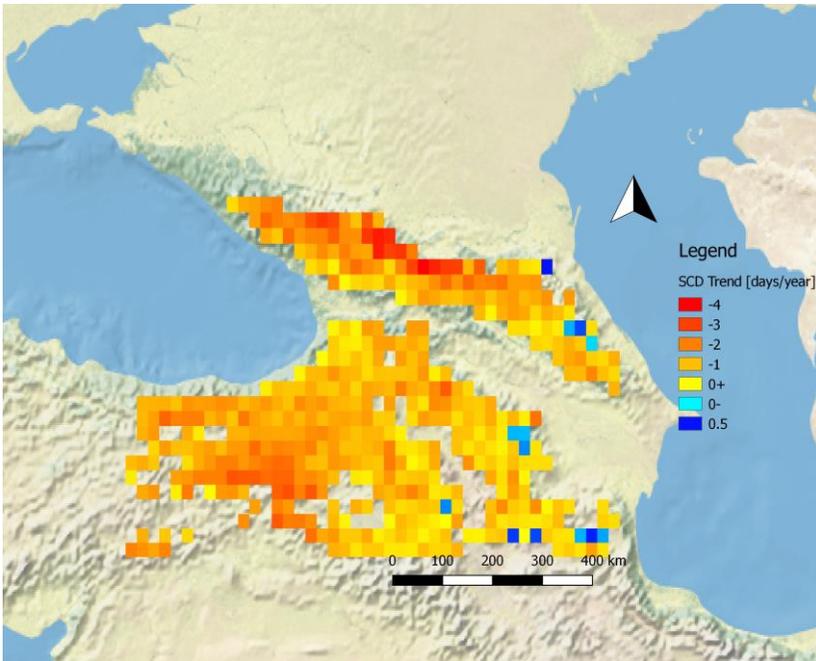
Bottom-Right: p-values

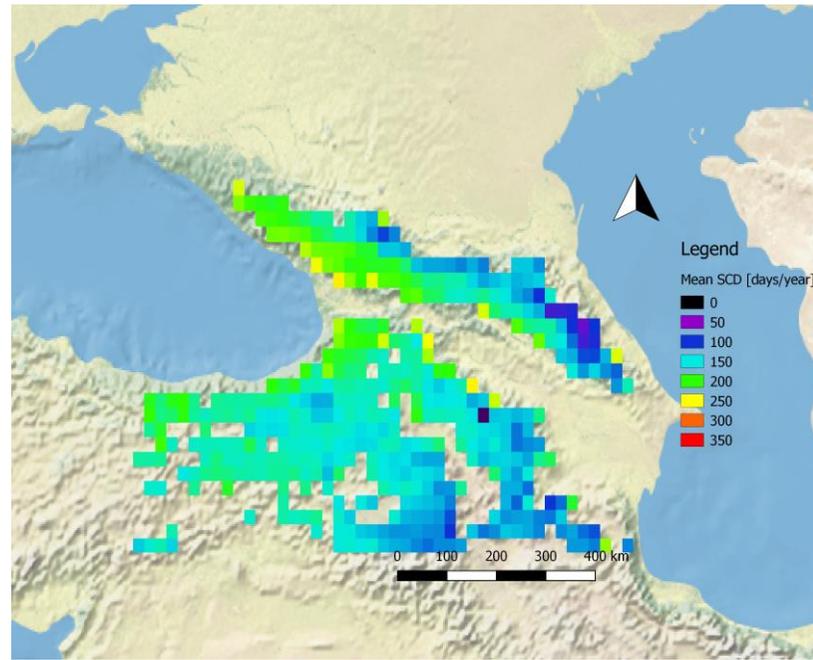
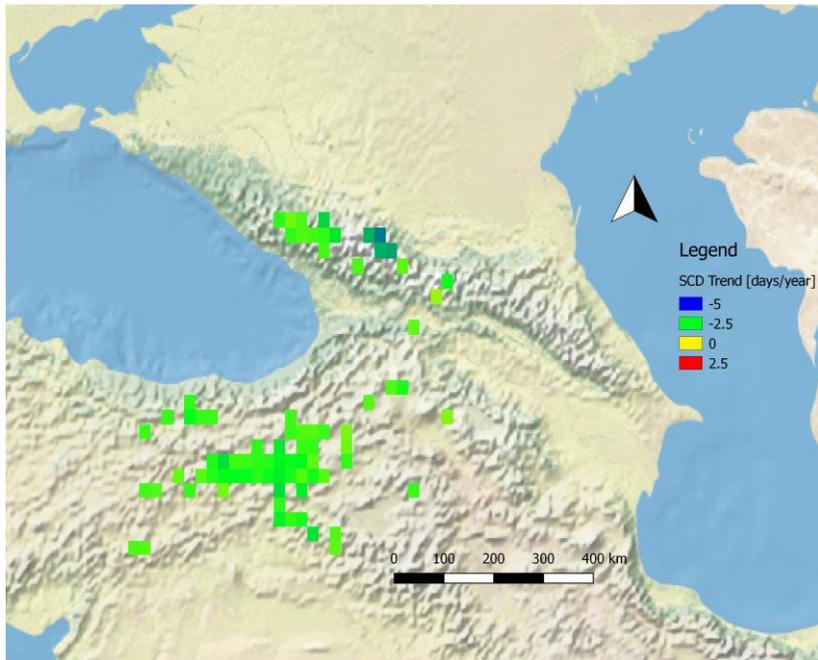


Elevation: 2400 m



Top-Left: significant trends
Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends
Bottom-Right: p-values



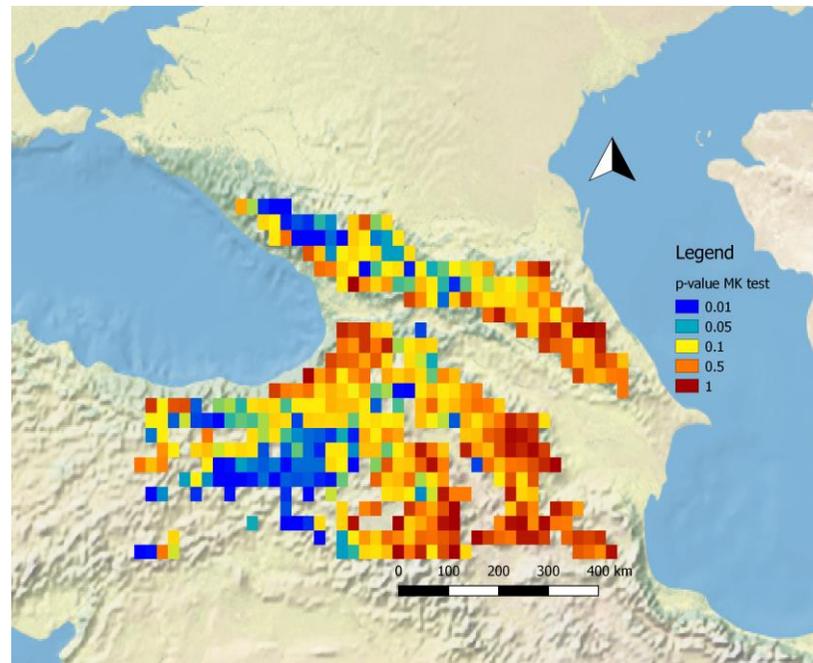
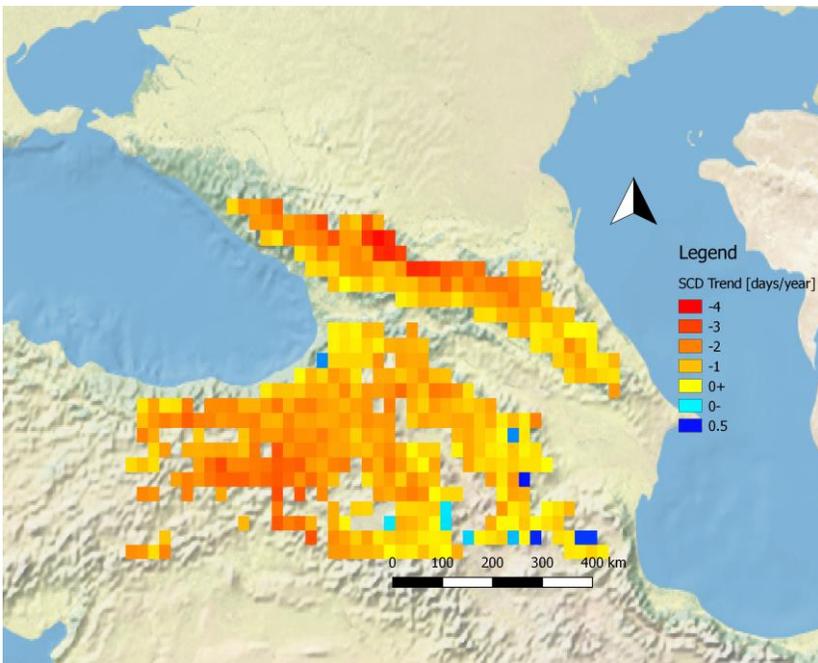


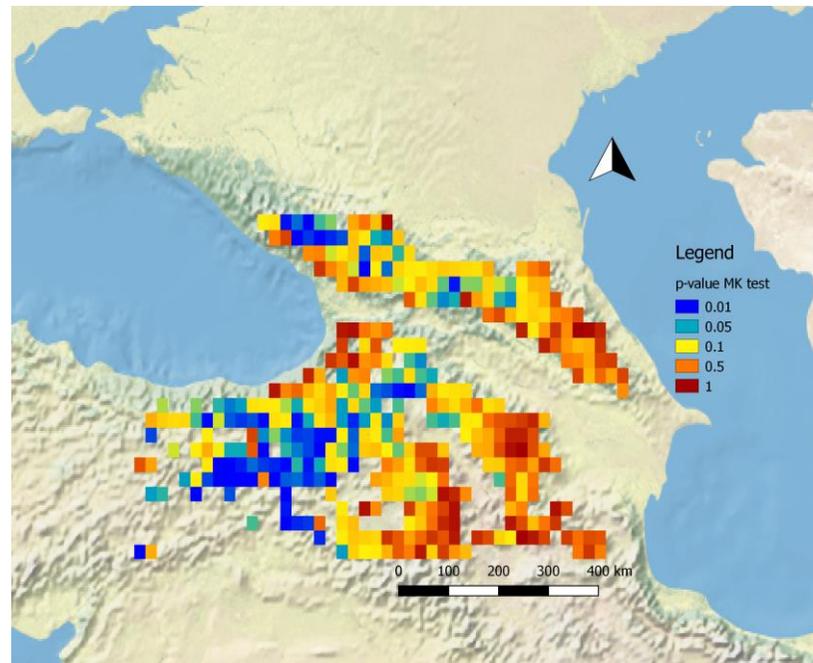
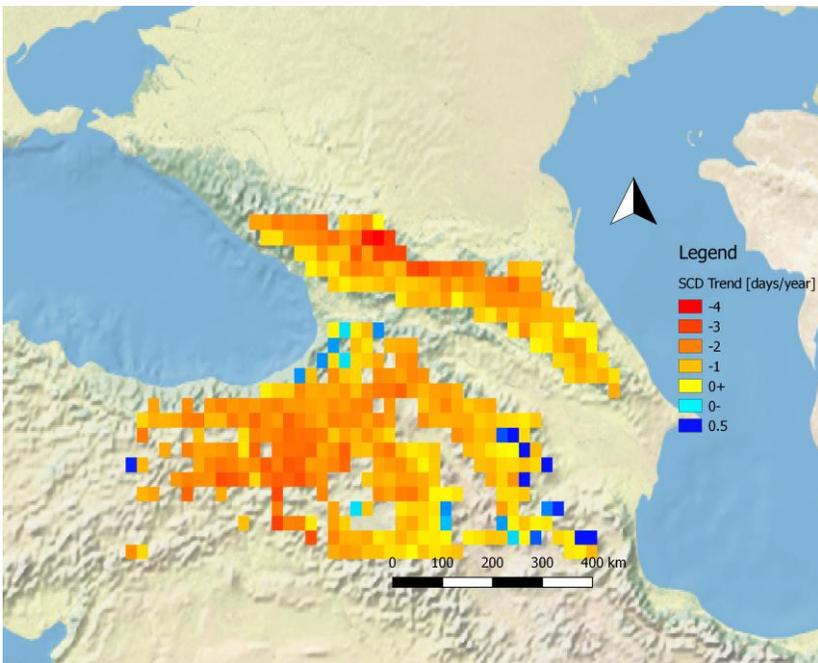
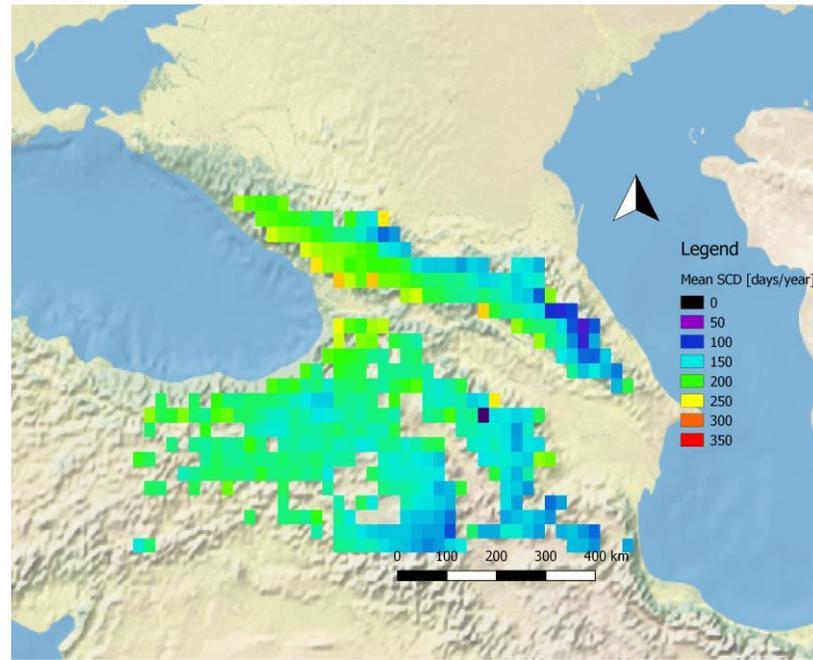
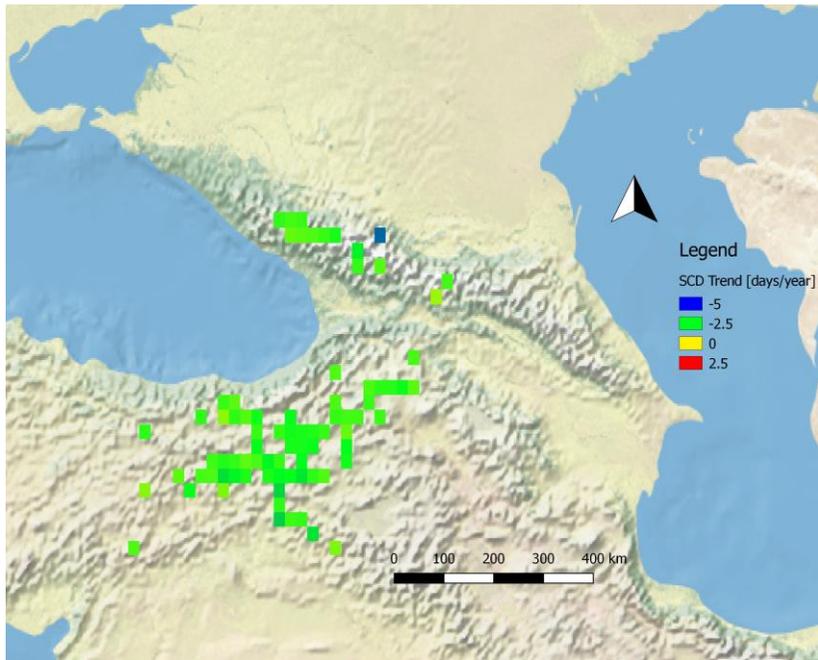
Elevation: 2500 m

Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

Bottom-Right: p-values



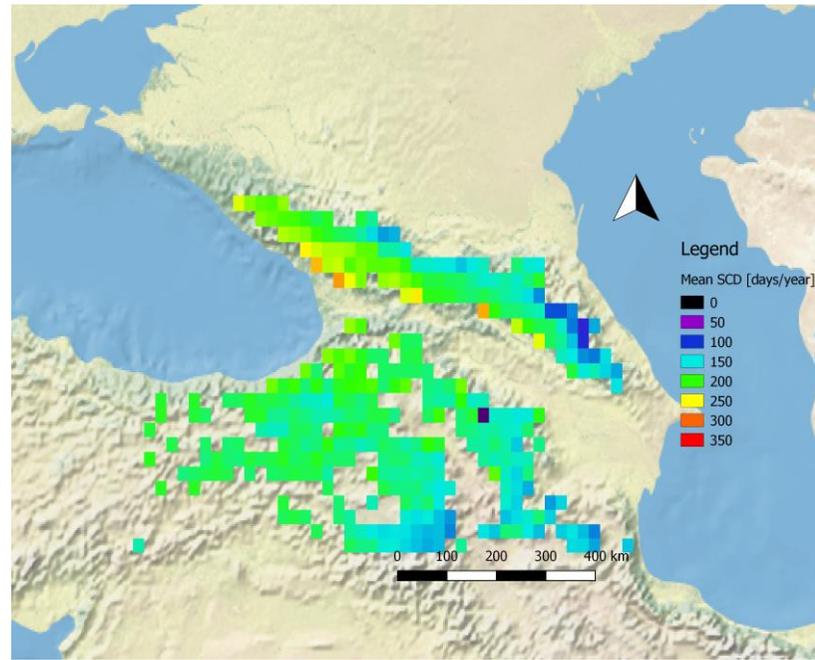
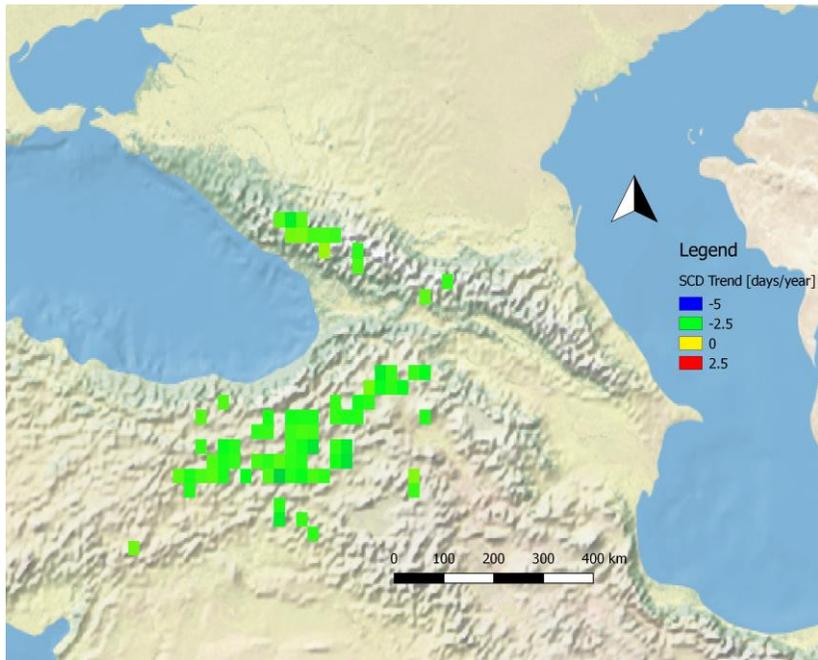


Elevation: 2600 m

Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

Bottom-Right: p-values

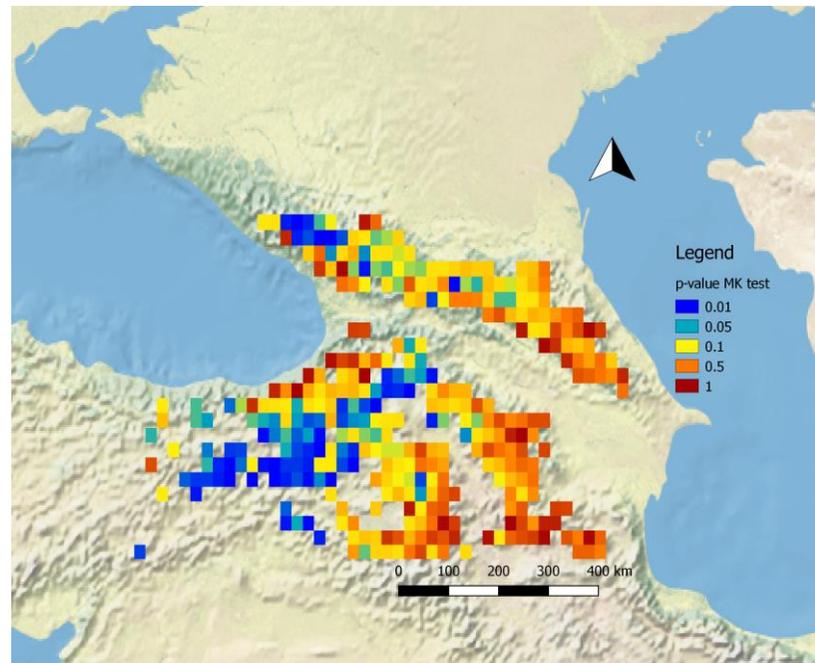
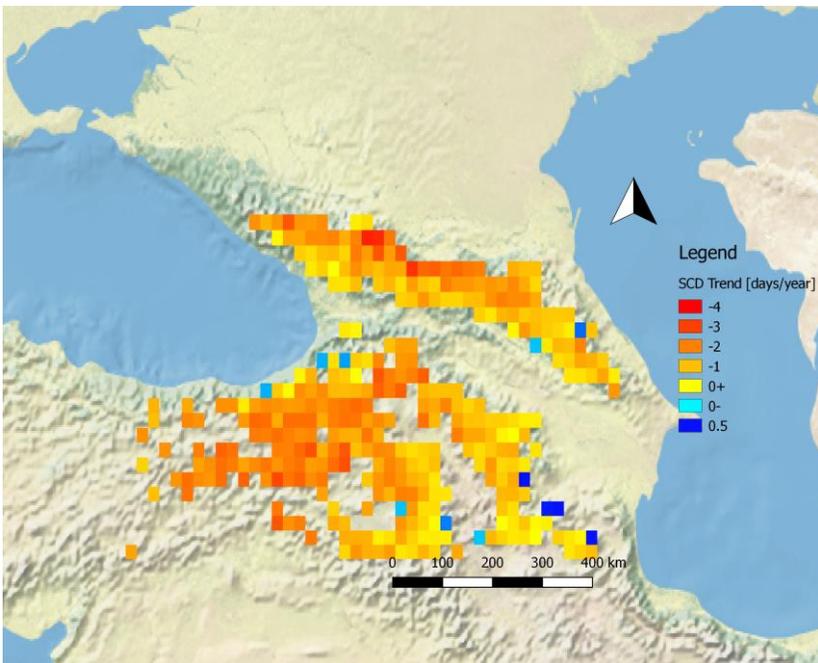


Elevation: 2700 m

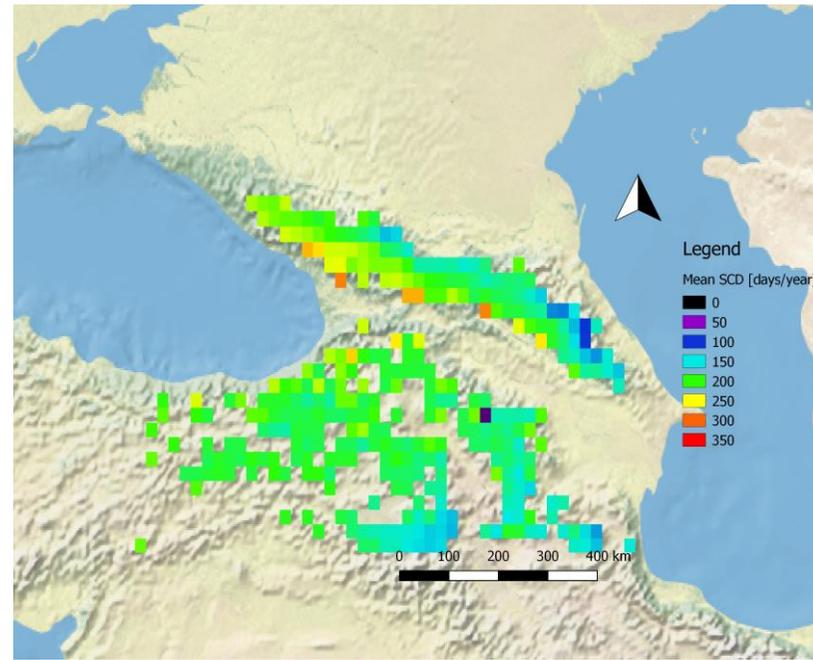
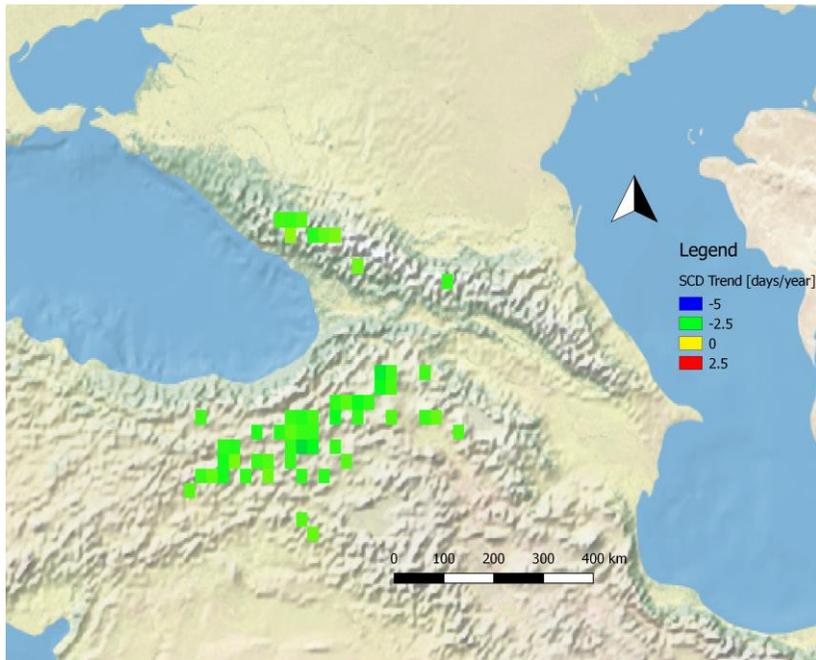
Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

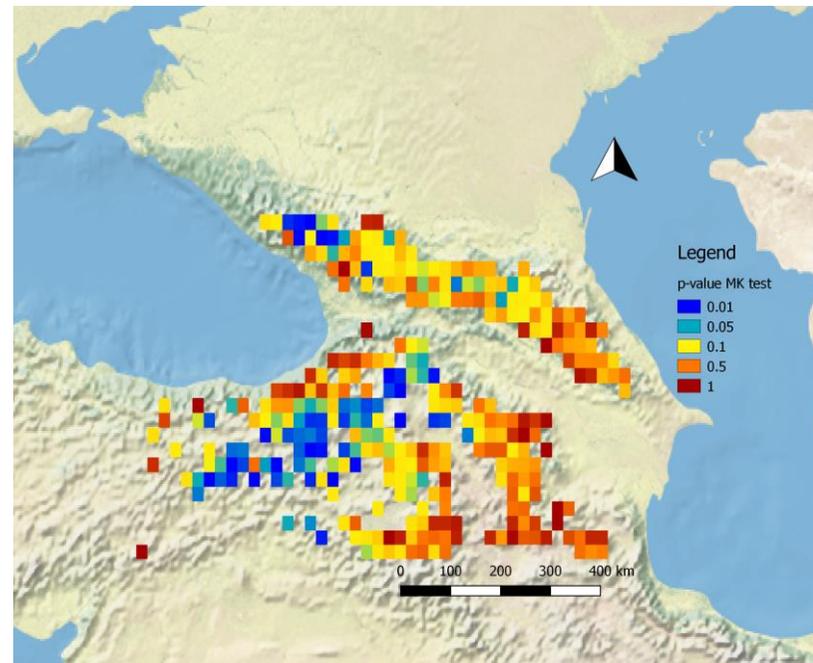
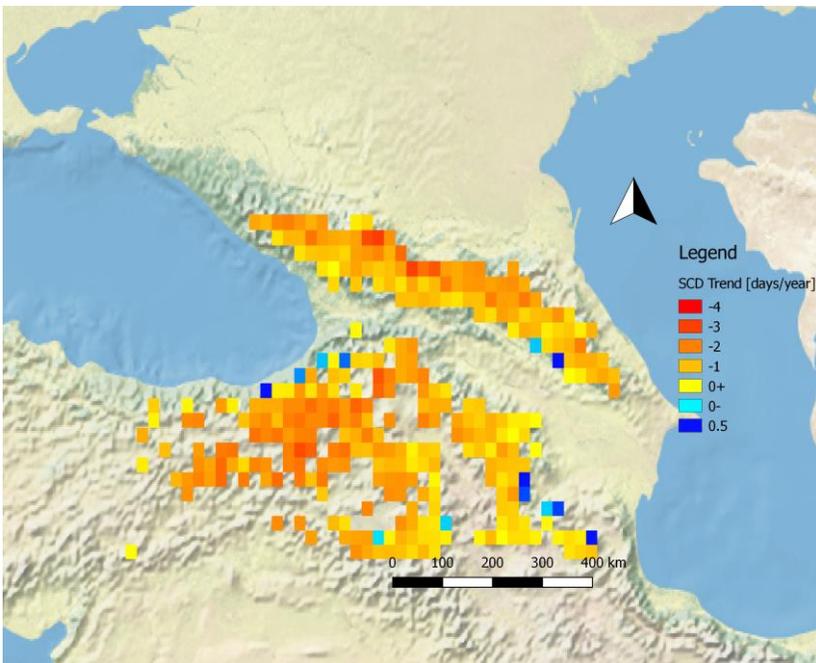
Bottom-Right: p-values

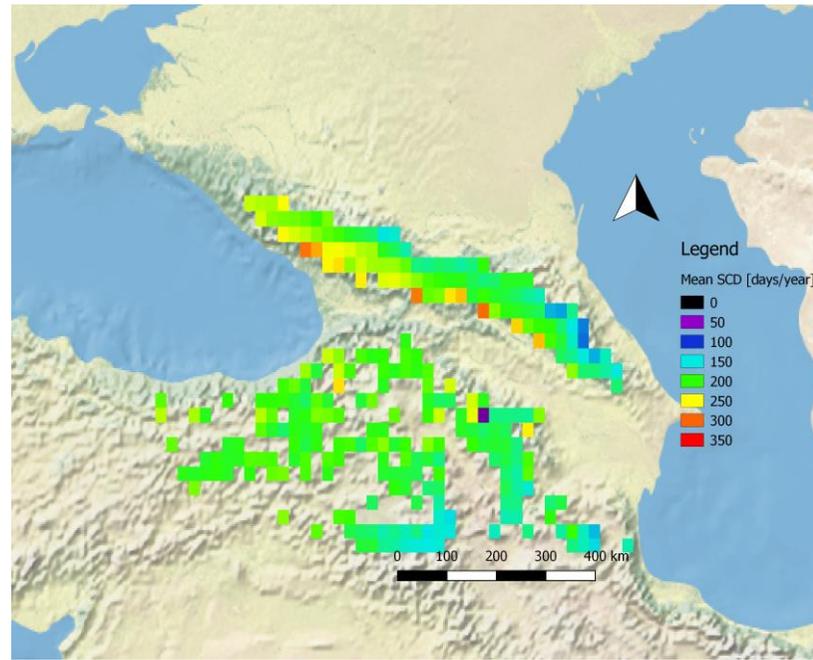
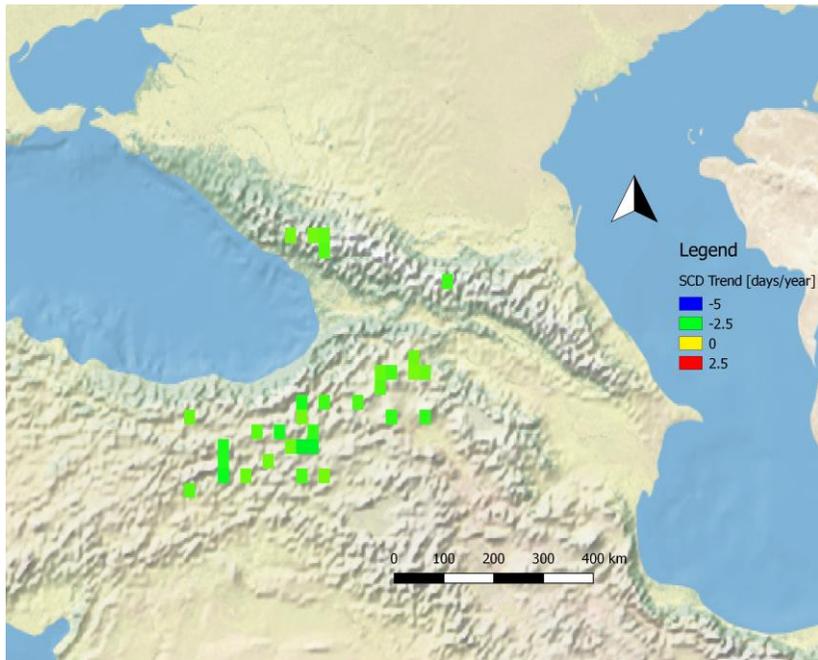


Elevation: 2800 m



Top-Left: significant trends
Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends
Bottom-Right: p-values



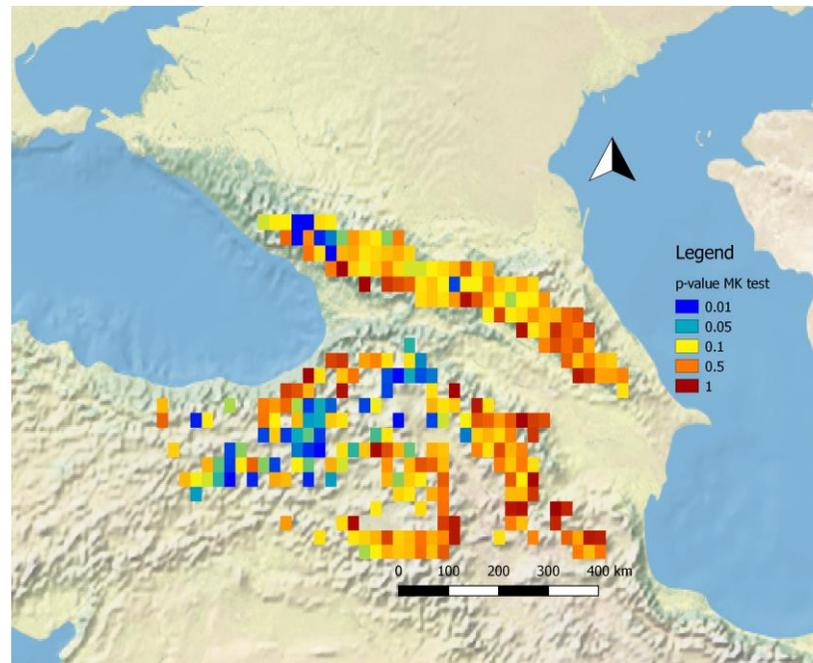
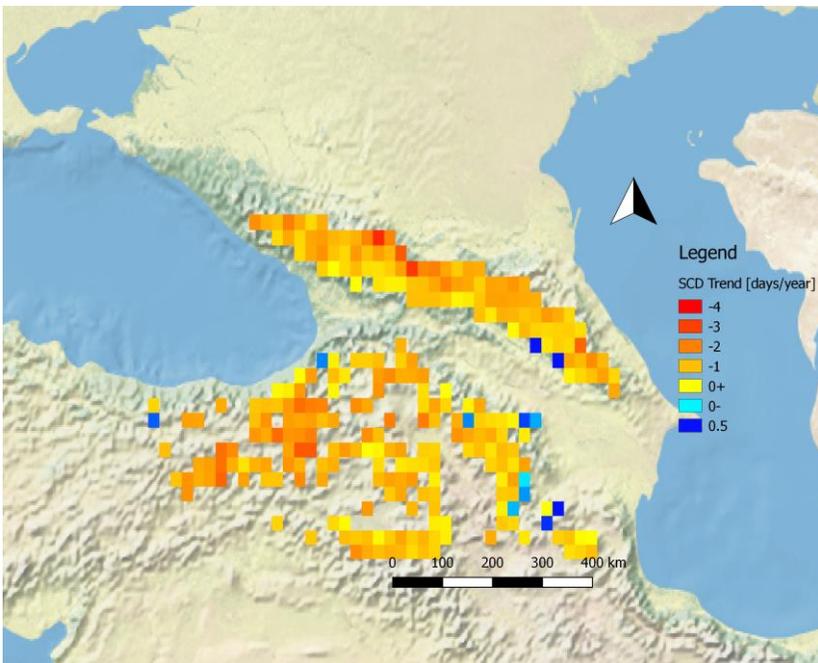


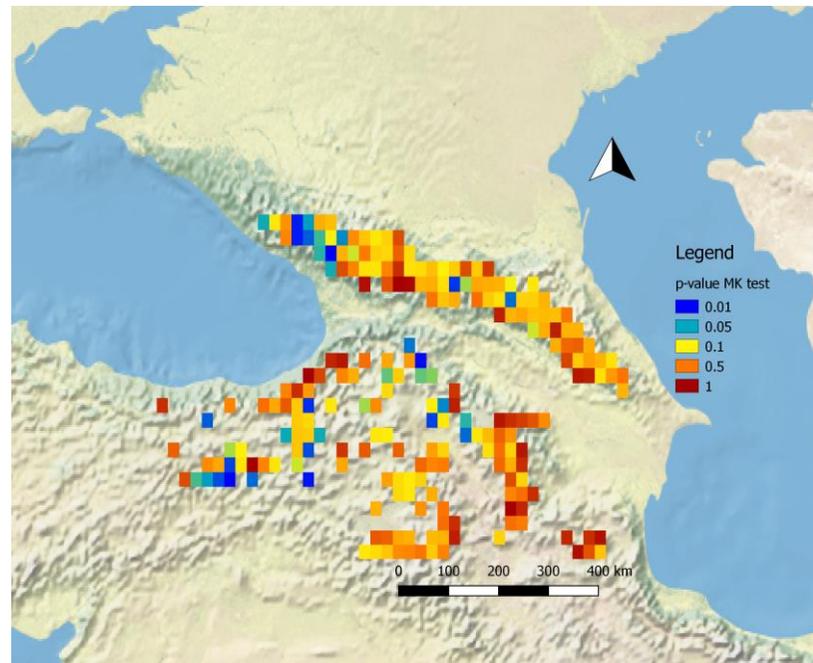
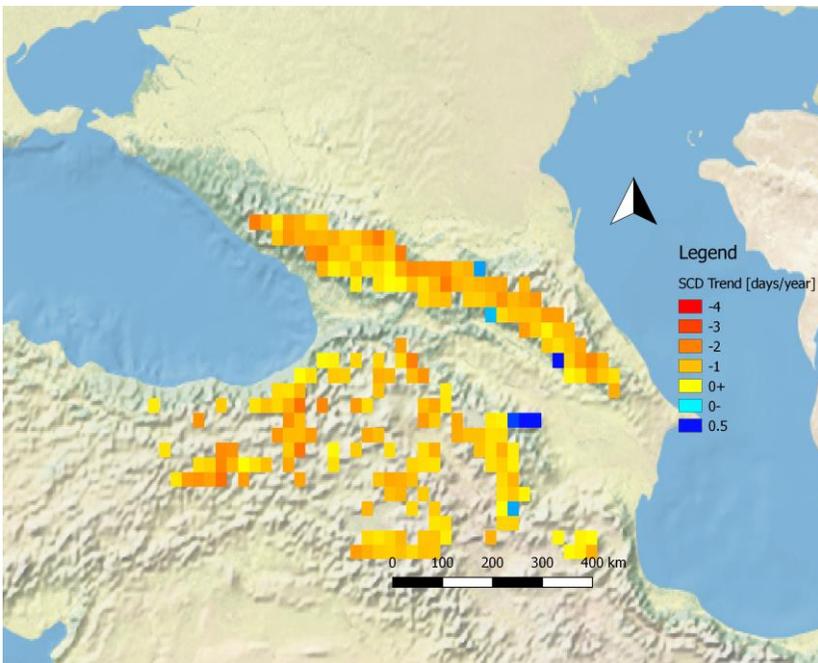
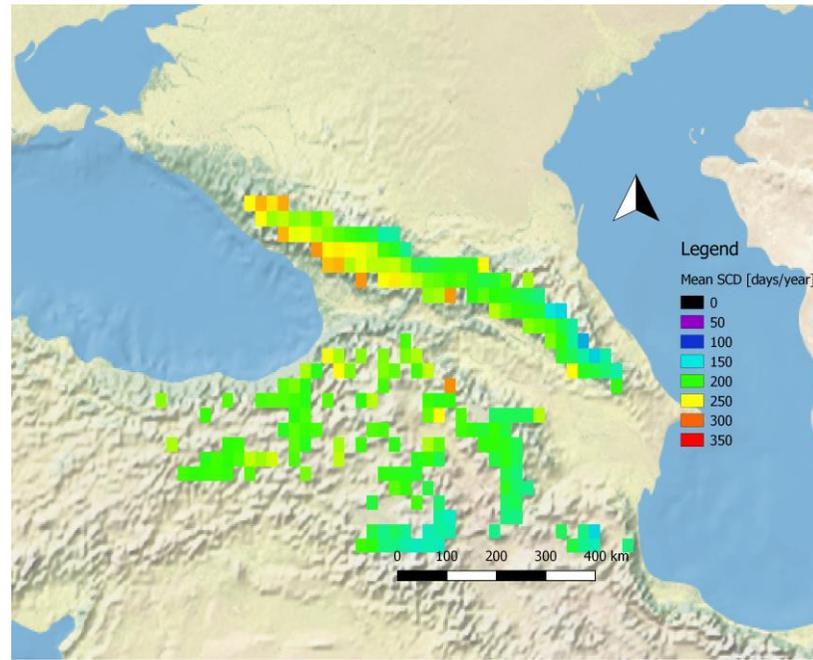
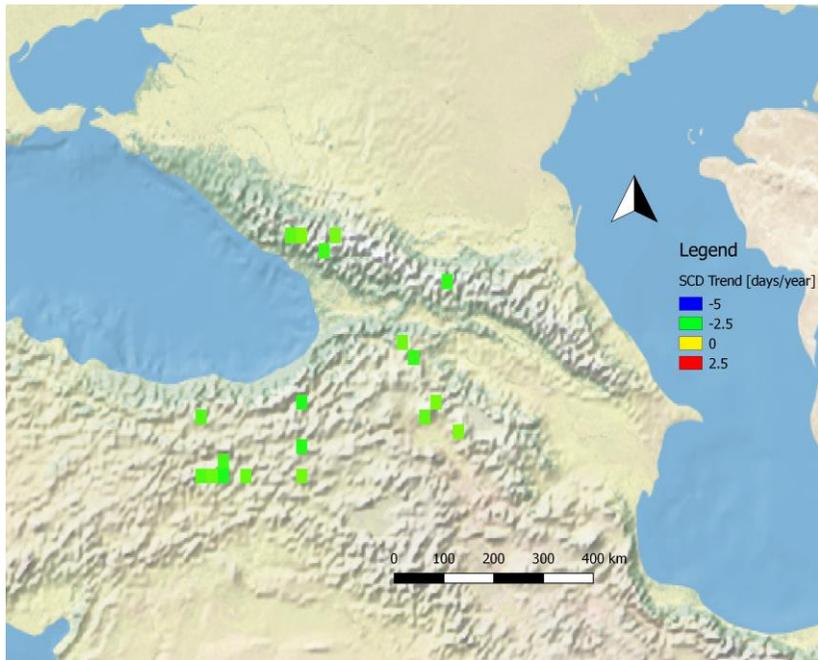
Elevation: 2900 m

Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

Bottom-Right: p-values



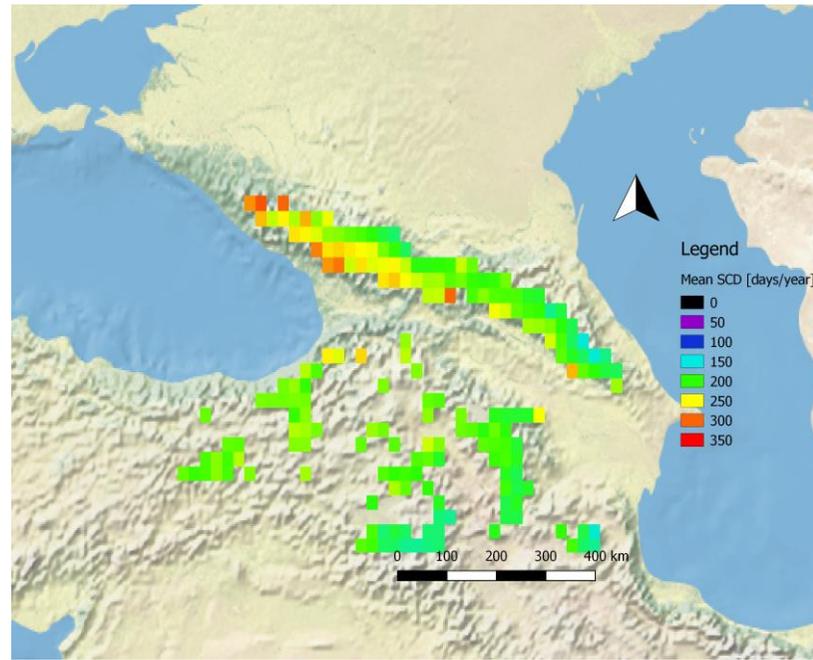
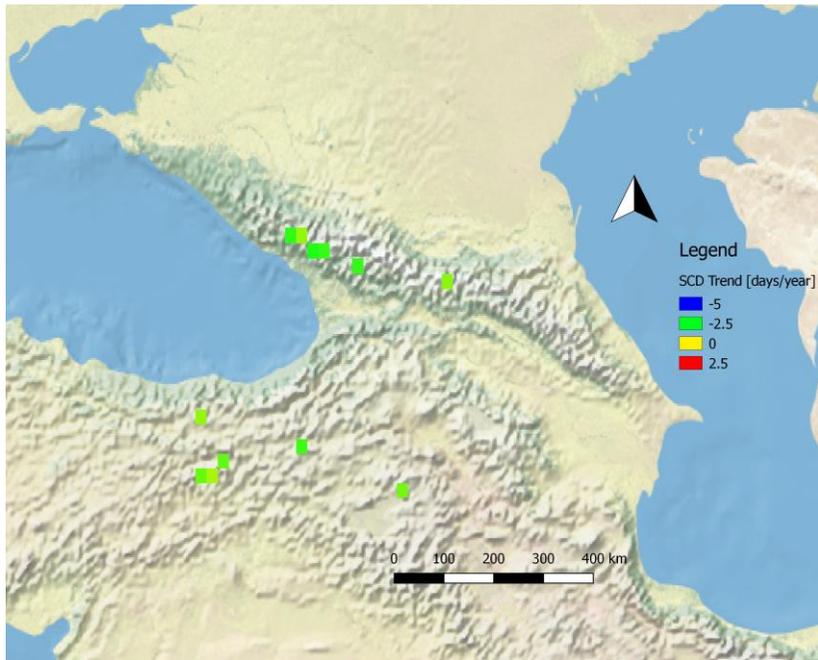


Elevation: 3000 m

Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

Bottom-Right: p-values

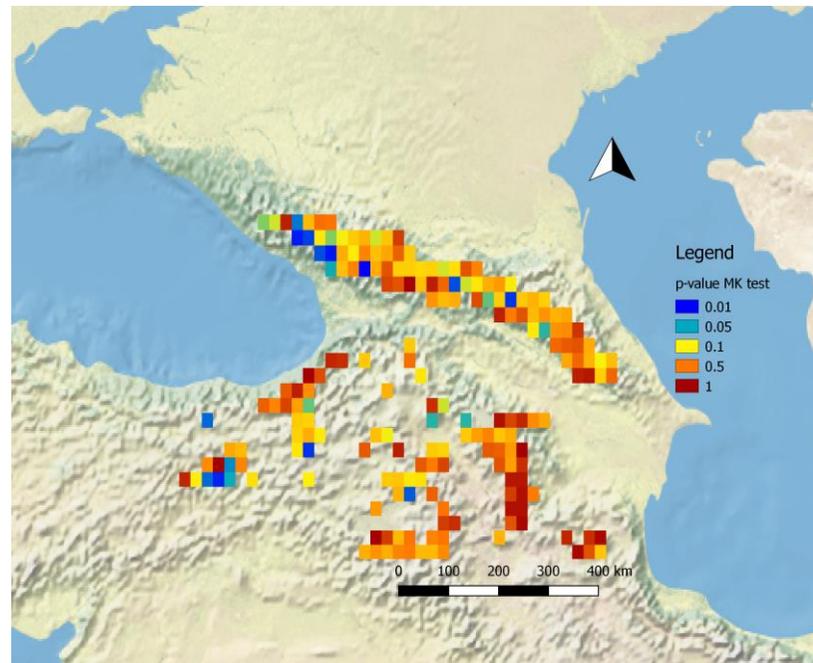
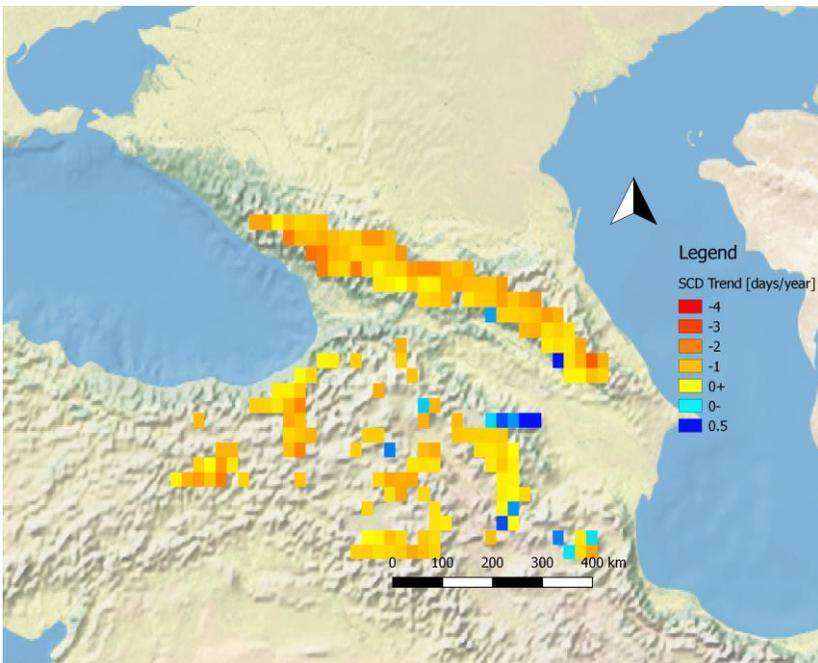


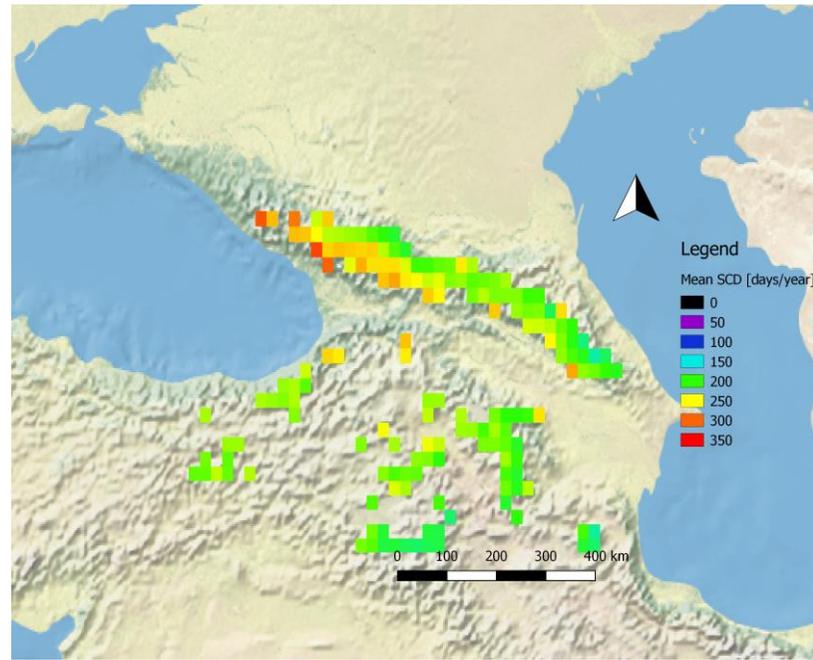
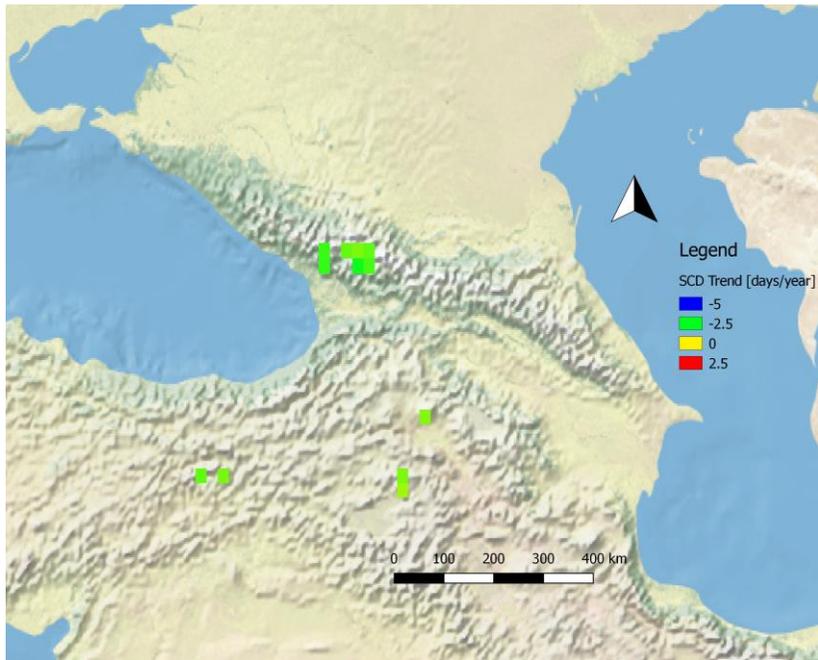
Elevation: 3100 m

Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

Bottom-Right: p-values



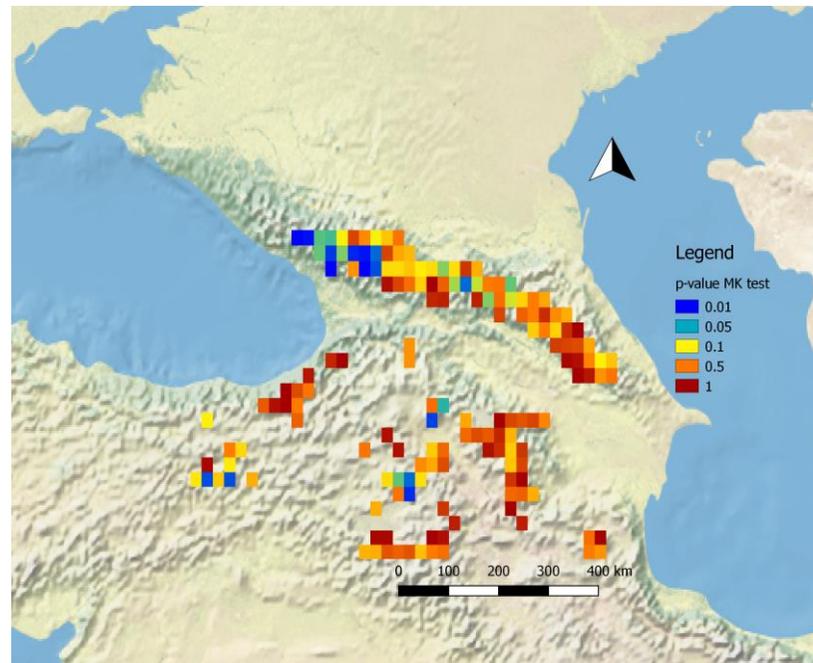
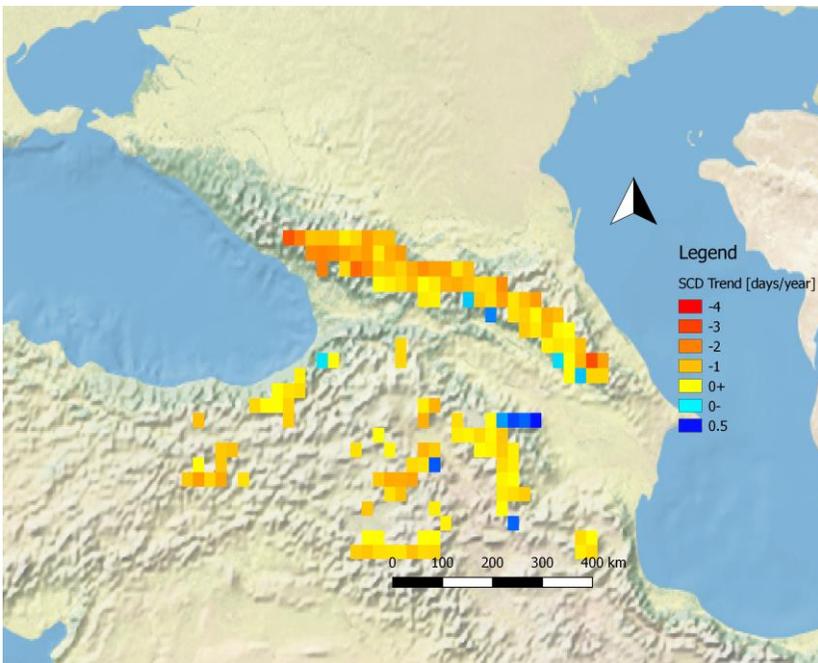


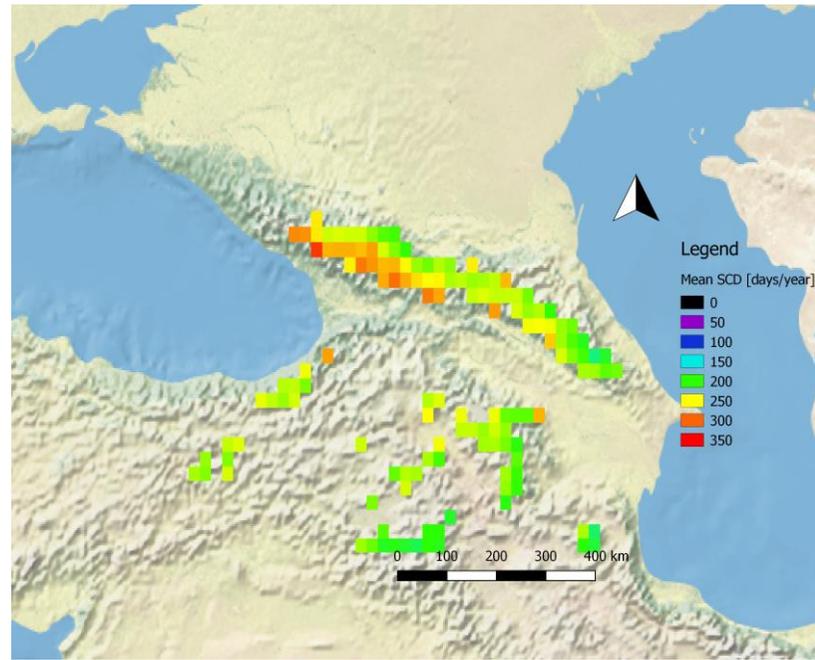
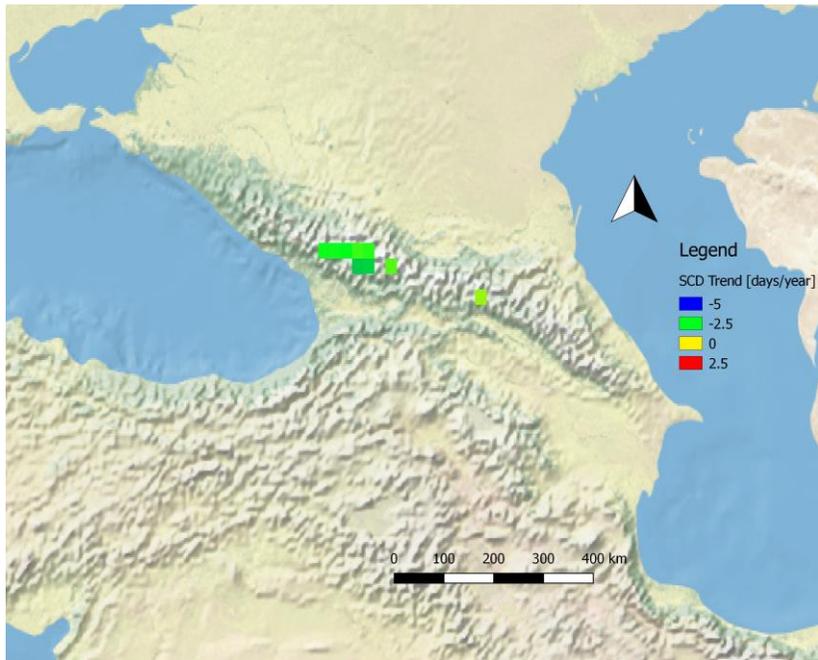
Elevation: 3200 m

Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

Bottom-Right: p-values



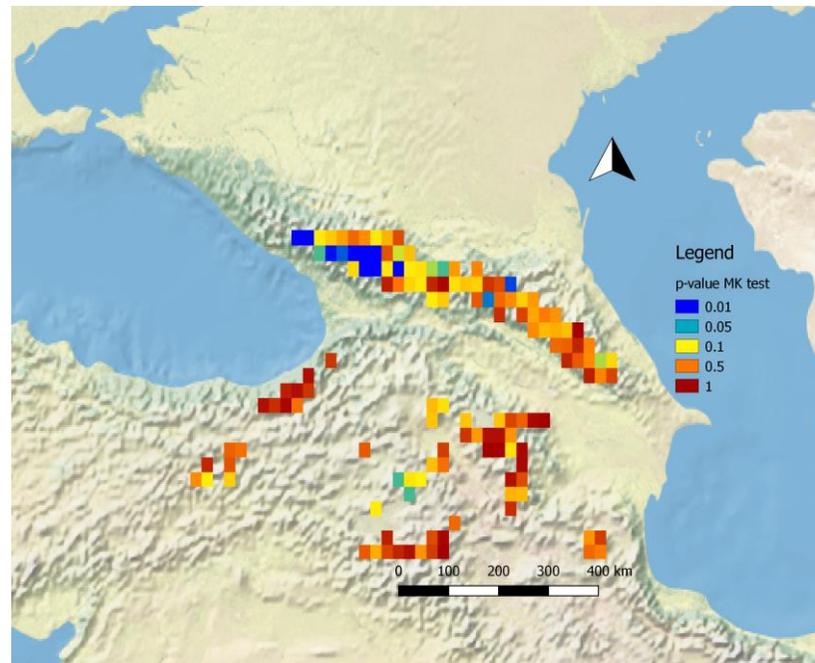
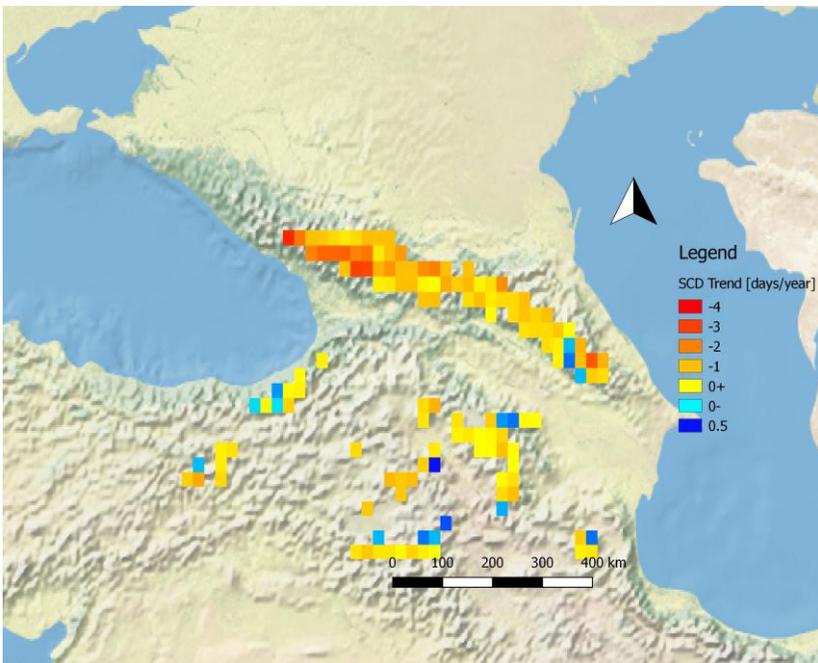


Elevation: 3300 m

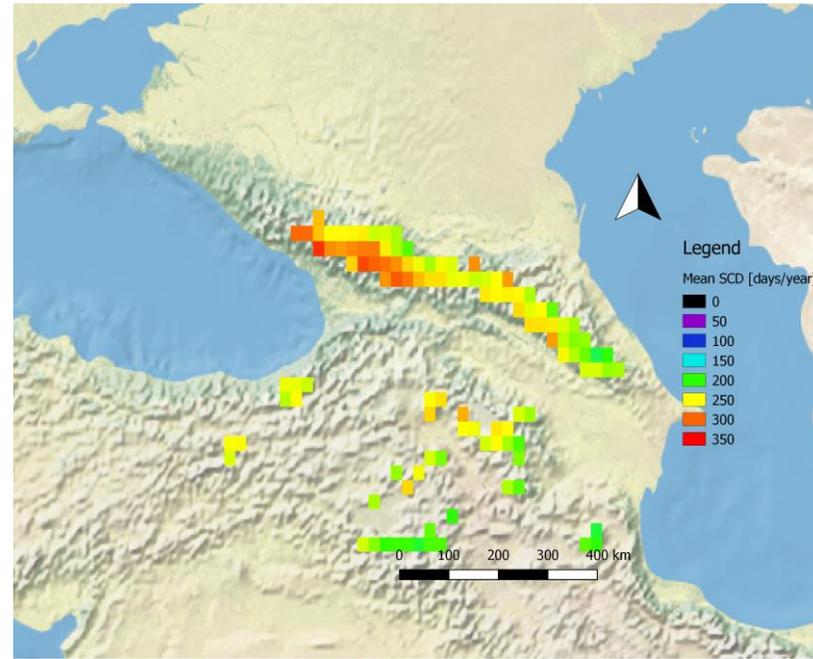
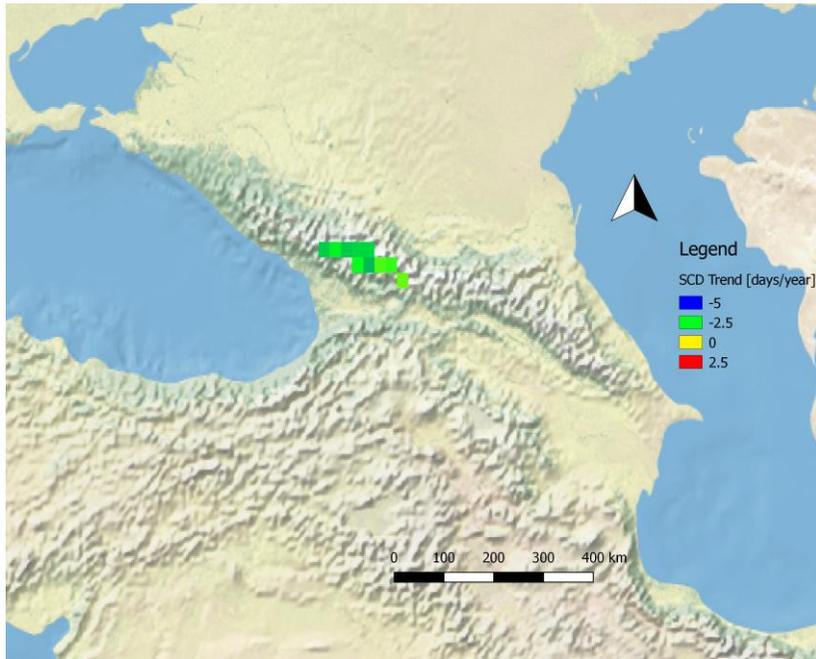
Top-Left: significant trends

Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends

Bottom-Right: p-values



Elevation: 3400 m



Top-Left: significant trends
Top-Right: mean SCD
Bottom-Left: All trends, including not significant trends
Bottom-Right: p-values

