Benoit Espinola

Statistical analysis of water data from an online EXO2 monitoring sonde

Modelling Chlorophyll-a using an EXO2 sonde

Helsinki Metropolia University of Applied Sciences

Bachelor of Engineering

Environmental Engineering, Water Resource Management

Thesis

Date 20 April 2017

Helsinki Metropolia University of Applied Sciences

Author(s) Title Number of Pages Date	Benoit Espinola Statistical analysis of water data from an online EXO2 monitoring sonde 47 pages + 26 appendices 20 April 2017
Degree	Bachelor of Engineering
Degree Programme	Environmental Engineering
Specialisation option	Water Resource Management
Instructor(s)	Veli-Matti Taavitsainen, Principal Lecturer and Thesis Supervisor Kaj Lindedahl, Principal Lecturer Antti Tohka, Principal Lecturer and Head of the programme Minttu Peuraniemi (project manager at Länsi-Uudenmaan Vesi ja Ympäristö ry for the Village Waters project).

This thesis aims to make statistical analysis and chlorophyll-a modelling based on the evolution over time of different chemical parameters in Lake Gennarbyträsket.
The project, which was done in collaboration with Länsi Uudenmaan Vesi ja Ympäristö ry, is part of the European Union's project Village Waters aiming to find and suggest better wastewater treatment solutions for villages and small settlements.

Chlorophyll-a is one of the indicators used to determine the trophic level of a lake. Understanding the roles of the different chemical parameters with the chlorophyll-a production process will help to develop a model that can be used for water quality monitoring system for municipalities. An EXO 2 water monitoring sonde has been deployed on site to measure in half an hour intervals pH, temperature, turbidity, dissolved oxygen, conductivity, a-chlorophyll and cyanobacteria levels.

Parameters evolve following different natural cycles, such as day and night or seasons. To be able to have enough data and to have significant results, the monitoring program needs to last for at least two sampling seasons. Unfortunately, the data available for this thesis partially covers an open water sampling season for chlorophyll-a. This thesis is therefore a preparation work for a possible long term research.

During this thesis, a model building methodology has been established, but this methodology requires validation using new data. The model seems to be providing good estimates for long period averages of chlorophyll-a despite its lack of precision for real time modelling.

The results of this thesis leads to believe on the existence of interactions between the different parameters measured. Nevertheless, there is clear need to collect more data, including local weather data, to be able to establish a more robust model of the production of chlorophyll-a. Additionally, further investigation is required to formulate proper conclusions.

Keywords	Statistical analysis, environmetrics, environmental modelling, Chlorophyll a, R, limnology

Contents

1 Introduction 1

2 Background 2

2.1 Länsi Uudenmaan Vesi ja Ympäristö ry 2
2.2 The EXO2 water station 8
2.3 Origin of the data 11
2.3.1 Water data 11
2.3.2 Weather data 12
2.4 Modelling and statistical analysis 12
2.5 Relevance of this thesis 13

3 Theoretical background 13

3.1 Lake nutrient levels and eutrophication 14
3.2 Chlorophyll-a 14
3.3 Factors that influence the levels of chlorophyll-a and its measurements 15

4 Methodology 18

4.1 Making data compatible with R 18
4.2 Time series 19
4.3 Correlation matrix 20
4.4 Variable standardization 20
4.4.1 Centering 21
4.4.2 Scaling 22
4.4.3 Effects of standardization on graphs 22
4.5 Smoothing 24
4.6 Wilkinson-Rogers notation 25
4.7 Multivariate data 27
4.8 Multiple Linear Regression 27
4.9 Training and testing sets 28
4.10 Cross validation 28
4.11 Variable selection 29
4.12 Principal Component Analysis 29
4.13 Process diagram 31

5 Results 31

5.1 Time series and smoothed series 31
5.1.1 pH 31
5.1.2 Conductivity 32
5.1.3 Temperature 33
5.1.4 Dissolved Oxygen 33
5.1.5 Chlorophyll-a and Turbidity 34
5.2 Evaluation of Lake Gennarbyträsket trophic level 35
5.3 Covariance matrix 38
5.4 Principal Component Analysis 40
5.5 Multiple Linear Regression 42

6 Discussion 44

6.1 Current setup of the EXO2 sonde 44
6.2 Needs for weather data 44
6.3 Needs for a working methodology with online sondes, field sampling and data treatment 45
6.4 Needs for nutrient monitoring 45
6.5 Needs to estimate lake retention time 46
6.6 Needs for additional data points 46
6.7 Further development 46
6.8 Online service possibilities 47
6.9 Programming good practices 47

7 Conclusion 47

8 Acknowledgements 48

Appendices

Appendix 1. Field data from LUVY for Lake Gennarbyträsket
Appendix 2. Maps
Appendix 3. Raw data
Appendix 4. The modelling process
Appendix 5. R code for examples
Appendix 6. Time series (pH)
Appendix 7. Time series (conductivity)
Appendix 8. Time series (temperature)
Appendix 9. Time series (turbidity)
Appendix 10. Time series (dissolved oxygen)
Appendix 11. Time series (chlorophyll-a)

Appendix 12. Comparison of the turbidity and chlorophyll-a (standardized using Equation 2)

Appendix 13. Comparison of the turbidity and chlorophyll-a (standardized using Equation 3)

Appendix 14. Smoothing

Appendix 15. Results for MLR1 (using standardization defined in Equation 2) – Run #1

Appendix 16. Results for MLR1 (using standardization defined in Equation 2) – Run #2

Appendix 17. Results for MLR1 (using standardization defined in Equation 2) – Run #3

Appendix 18. Results for MLR2 (using standardization defined in Equation 3) – Run #1

Appendix 19. Results for MLR2 (using standardization defined in Equation 3) – Run #2

Appendix 20. Results for MLR2 (using standardization defined in Equation 3) – Run #3

Appendix 21. Principal Component Analysis

Appendix 22. Source code (Lake quality assessment tool)

Appendix 23. Source code (MLR1 and smoothing)

Appendix 24. Source code (MLR2 and smoothing)

Appendix 25. Source code (PCA)

Appendix 26. Source code (Time Series)

Table of figures:

Figure 1. Operating area of Länsi-Uudenmaan vesi ja ympäristö ry (Länsi-Uudenmaan Vesi ja Ympäristö ry, 2013) 3

Figure 2. Hierarchy model of the thesis project. 4

Figure 3. View to Lake Gennarbyträsket. 4

Figure 4. Diver ready for the inspection dive. 5

Figure 5 Dive depth profile following the planned route for the sewage pipeline. 6

Figure 6. Lake Gennarbyträsket underwater. 7

Figure 7. Maintenance work on the EXO2 sonde (image credit: LUVY) 8

Figure 8. EXO2: a digital multiparameter sonde (image credit: LUVY) 9

Figure 9. EXO2 underwater in Lake Gennarbyträsket (image credit: LUVY). 11

Figure 10. Underwater view to the EXO2 sonde 16

Figure 11. Filamentous algae growing on an aquatic plant, the EXO 2 sonde can be seen in the background. 17

Figure 12. Lake pH example. 19

Figure 13. Plot of series A 23

Figure 14. Plot of series B 23

Figure 15. A and B not standardized 24

Figure 16. A and B standardized using scale 24

Figure 17. Example of smoothing data using loess 25

Figure 18. Example of hand shadow game 30
Figure 19. Bar plot output from the water quality evaluation tool. 36
Figure 20. Box plot output from the assessment tool 37
Figure 21. Chlorophyll-a levels over time from the assessment tool 38
Figure 22. Colour-coded plot output from the assessment tool 38
Figure 23. Original routing of the sewer pipe through Lake Gennarbyträsket (image credit: LUVY 1
Figure 24. Sampling locations 2
Figure 25. Process diagram for the data pre-treatment, smoothing and model construction. 1
Figure 26. pH time series 1
Figure 27. pH monthly (30 days) overlapping time series 2
Figure 28. pH weekly overlapping time series 3
Figure 29. Conductivity time series 1
Figure 30. Conductivity monthly (30 days) overlapping time series 2
Figure 31. Conductivity weekly overlapping time series 3
Figure 32. Temperature time series 1
Figure 33. Temperature monthly (30 days) overlapping time series 2
Figure 34. Temperature weekly overlapping time series 3
Figure 35. Turbidity time series 1
Figure 36. Turbidity monthly (30 days) overlapping time series 2
Figure 37. Turbidity weekly overlapping time series 3
Figure 38. Dissolved oxygen time series 1
Figure 39. Dissolved oxygen monthly (30 days) overlapping time series 2
Figure 40. Dissolved oxygen weekly overlapping time series 3
Figure 41. Chlorophyll-a time series 1
Figure 42. Chlorophyll-a monthly (30 days) overlapping time series 2
Figure 43. Chlorophyll-a weekly overlapping time series 3
Figure 44. Turbidity and Chlorophyll-a comparison (standardized using Equation 2) 1
Figure 45. Turbidity and chlorophyll comparison standardized using Equation 3 1
Figure 46. Chlorophyll-a smoothed curve 1
Figure 47, Chlorophyll-a strongly smoothed curve 2
Figure 48. Turbidity smoothed curve 3
Figure 49. Conductivity smoothed curve 4
Figure 50. pH smoothed curve 5
Figure 51. Dissolved Oxygen smoothed curve 6
Figure 52. Temperature smoothed curve 7

Figure 53. Comparison of turbidity and chlorophyll-a from smooth fits 8
Figure 54. Comparison of measured Chlorophyll-a to model estimations for MLR1, Run #1 2
Figure 55. Boxplot of the measured Chlorophyll-a and model estimations for MLR1, Run #1 2
Figure 56. Comparison of measured Chlorophyll-a to model estimations for MLR1, Run #2 2
Figure 57. Boxplot of the measured Chlorophyll-a and model estimations for MLR1, Run #2 2
Figure 58. Comparison of measured Chlorophyll-a to model estimations for MLR1, Run #3 2
Figure 59. Boxplot of the measured Chlorophyll-a and model estimations for MLR1, Run #3 2
Figure 60. Comparison of measured Chlorophyll-a to model estimations for MLR2, Run #1 2
Figure 61. Boxplot of the measured Chlorophyll-a and model estimations for MLR2, Run #1 2
Figure 62. Comparison of measured Chlorophyll-a to model estimations for MLR2, Run #2 2
Figure 63. Boxplot of the measured Chlorophyll-a and model estimations for MLR2, Run #2 2
Figure 64. Comparison of measured Chlorophyll-a to model estimations for MLR2, Run #3 2
Figure 65. Boxplot of the measured Chlorophyll-a and model estimations for MLR2, Run #3 2
Figure 66. PCA variance with respect to the principal components 1
Figure 67. PCA biplot of PC1 and PC2, hot temperatures are red and cold ones are blue, black line represents time 2
Figure 68. PCA biplot of PC3 and PC4, hot temperatures are red and cold ones are blue, black line represents time 3

Tables:

Table 1. EXO2 probes used in Lake Gennarbyträsket. 10
Table 2. Eutrophication level guideline according to the concentration of chlorophyll-a in surface freshwater bodies. 15
Table 3. Pearson Correlation Matrix 39

Table 4. Spearman Correlation Matrix 39
Table 5. Field measurements..... 1
Table 6. Laboratory analysis results..... 1
Table 7. head of the EXO2 raw data 1
Table 8. selection from the rain accumulation raw data..... 1
Table 9. Selection of weather raw data. 2

Equations:

Equation 1.. Photosynthesis reaction 14
Equation 2 20
Equation 3 21

1 Introduction

Water plays a major role in the environment and in the society, it supports life, it enables recreational activities, it provides pleasant surroundings and enables economic activities. Water can be used in numerous ways, such as fishing, scuba diving, drinking and irrigation. The presence of a clean lake allows fishing activities, increases property values and provides a pleasant environment. It is therefore essential to protect and monitor this resource. The Village Waters project aims to improve the wastewater treatment solutions in villages across the Baltic sea region to reduce their environmental impact. A monitoring program is essential to be implemented to measure the evolution of the impact of the wastewater treatment solutions from those villages. In Finland, the association Länsi Uudenmaan Vesi ja Ympäristö ry is responsible to monitor the water quality in the Gennarby village, built around Lake Gennarbyträsket. The association has chosen to deploy an EXO2 water sonde to monitor 7 parameters from the lake: pH, conductivity, turbidity, dissolved oxygen, temperature, chlorophyll-a (referred in the rest of this document as chlorophyll) and cyanobacteria. These parameters help to assess the water quality of the lake. The trophic level of a lake is a key indicator of the health of the lake ecosystem. Chlorophyll allows to estimate the lake's trophic level. Unfortunately, the chlorophyll-a sensor cost is among the highest in an EXO2 setup. Therefore, it can be economically interesting to find ways to assess the levels of chlorophyll based on cheaper commodity sensors.

The goal of this thesis is to perform statistical analysis of data recorded by the EXO2 water sonde, and to establish a methodology to create models to estimate levels of chlorophyll from pH, conductivity, turbidity, dissolved oxygen and temperature observations. R is used to perform the statistical analysis and to generate the models.

In the next section, the background will be presented. This will be followed by Section 3, the theoretical background covers concepts of limnology. The Section 4, presenting the methodology that has been used for this thesis. Section 5 and 6 present respectively the results and discussions. Finally, I conclude this thesis in Section 7.

2 Background

This thesis has been possible thanks to the collaboration between different entities. This section presents the context in which this thesis has been developed and basic information on the EXO 2 probe and its sensors. The final part of this section presents arguments supporting the relevance of this thesis.

2.1 Länsi Uudenmaan Vesi ja Ympäristö ry

Founded in 1975, Länsi Uudenmaan Vesi ja Ympäristö ry (referred in the rest of this document as LUVY) is an association specialized in research related to water such as sampling, analysis and monitoring, wastewater treatment consulting and microbial assessment (Länsi-Uudenmaan Vesi ja Ympäristö ry, 2016). LUVY provides services including consulting, research-based studies and laboratory analysis. LUVY's accredited testing laboratory is authorized by the FINAS - Finnish accreditation services (T147 accreditation requirement SFS-EN ISO/IEC 17025:2005). Based in Lohja (Finland), the association covers the western area of the Uusimaa region (Figure 1). LUVY also has offices in Tvärminne Zoological Station (near Hanko, Finland) and in Raasepori (Finland).

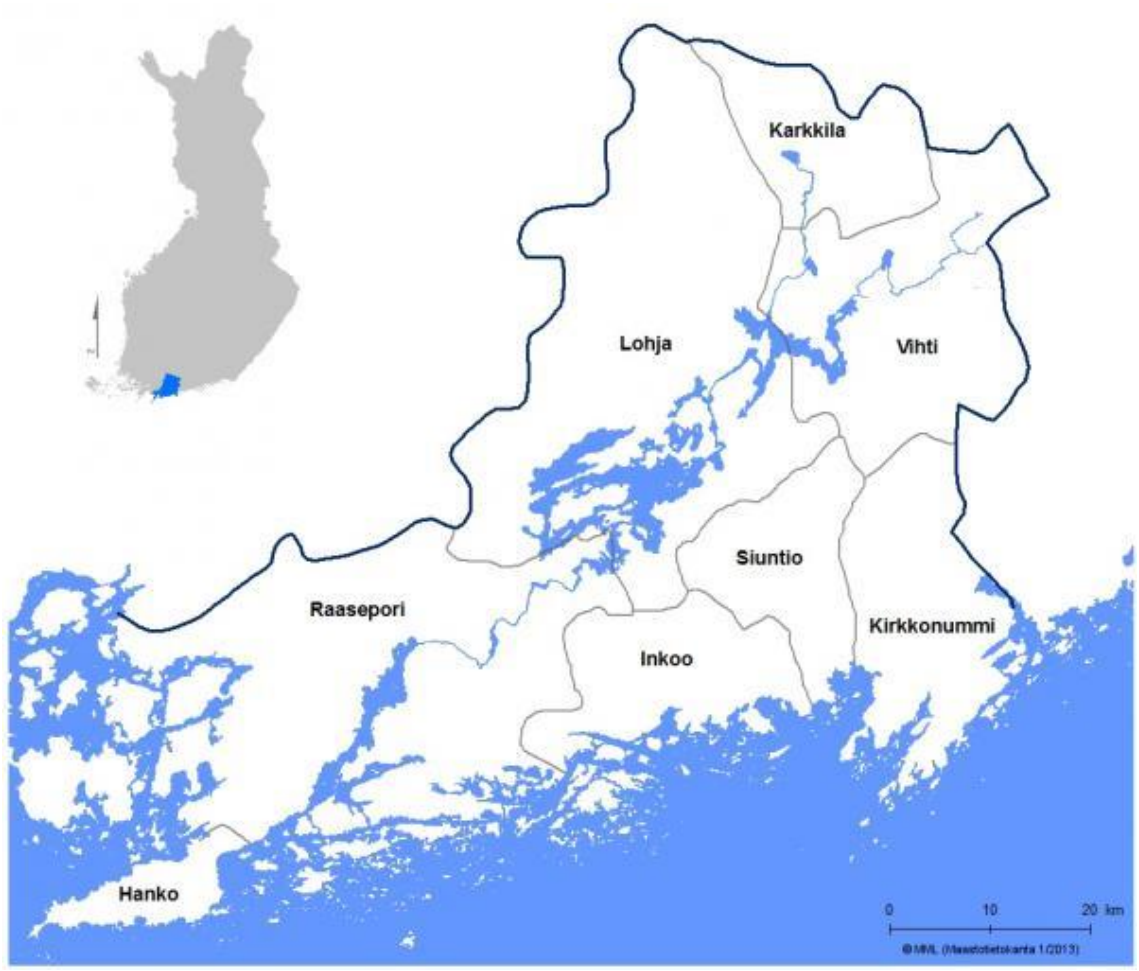

Figure 1. Operating area of Länsi-Uudenmaan vesi ja ympäristö ry (Länsi-Uudenmaan Vesi ja Ympäristö ry, 2013)

LUVY's research-based projects cover a wide range of research topics, such as water ecology, fish stocks, benthic animals, aquatic vegetation and wastewater purification. The Scattered wastewater project (LINKKI) is one of LUVY's ongoing projects. Launched in 2009, the project's goal is to assess waste water treatment systems which are not connected to the municipal wastewater network (Länsi-Uudenmaan Vesi ja Ympäristö ry, 2016). The LINKKI project is mainly funded by the participating municipalities and the Finnish Environmental Ministry.

The LINKKI project also participates to the European Union funded Village Waters project. The aim of Village Waters is to provide insights on the best possible solution for wastewater treatment for scattered settlements in villages around the Baltic Sea and to reduce nutrient load into the Baltic Sea. The project aims to find cost effective solutions that are financially viable (Village Waters, 2017a).

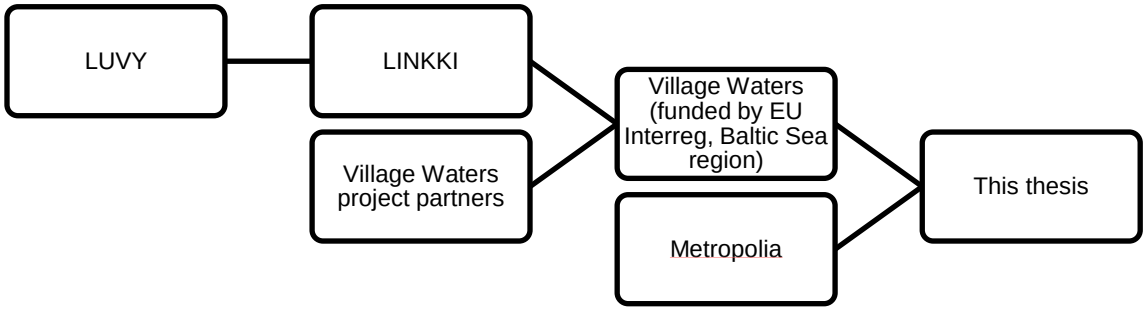

Figure 2. Hierarchy model of the thesis project.

This thesis is made in collaboration with LUVY, LINKKI and Village Waters, as seen in Figure 2.

LUVY and LINKKI selected the Gennarby to participate in the Village Waters project. More specifically, the area chosen includes approximately 10 properties around Lake Gennarbyträsket (Village Waters, 2017b). A view to Lake Gennarbyträsket is visible in Figure 3.

Figure 3. View to Lake Gennarbyträsket.

The properties around Lake Gennarbyträsket are over 100 years old, none is connected to a municipal sewage system, the wastewater treatment solutions are often outdated septic tanks and their drinking water comes from private wells. As a result, water quality varies. The village has a project to connect to the municipal wastewater and water system using pipelines. The goal is to improve both lake and drinking water quality. Additionally, power lines and optical cables will be installed underground for the houses. In order to optimize financial resources, the excavation work is done at the same time. Originally, a section of the sewage pipe was going to be installed in the bottom of the lake as described in Figure 33 (that can be found in Appendix 2). The construction work is not financed by Village Waters, it is financed by an organized cooperation between the villagers.

An inspection dive in the lake has been made before construction work started. The dive was performed on the 6th of October 2016, by Anu Suonpää (LUVY) and myself, it had a duration of 36min and a maximum depth of 7.9m (note that we were 1 to 1.5m away from the top of the sediment layer).

Figure 4. Diver ready for the inspection dive.

The dive profile can be seen in the Figure 5 bellow. According to the dive computer, the water temperature was about 10°C during the entire dive. We followed the planned route for the sewage pipe and documented it by recording a video. The length of the dive line

was about 300m. During the dive, we had a visibility of 6 meters by the surface and nearly null in the bottom. Around 20 meters away from the north-east shore, we have spotted old milk containers that reveal the history of the site. After further investigating with locals, we found out that there used to be a milk farm in the area. Plants were present on depth of less than 3 meters on both shores.

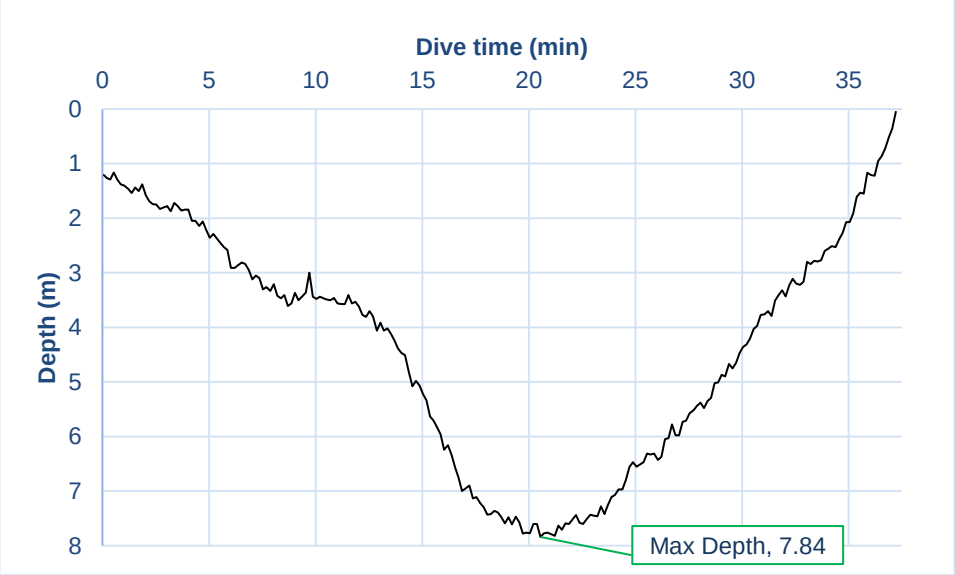

Figure 5 Dive depth profile following the planned route for the sewage pipeline.

Plans for applying the pipeline have then changed, the entire length of the pipeline has now been buried on the land and does not pass through the lake.
In order to monitor the evolution of the water quality in Lake Gennarbyträsket, three samples were taken from each sampling point on the map found in Figure 34 (on Appendix 2). Additionally, LUVY has installed an EXO2 water sonde to have an online continuous monitoring of water parameters in Lake Gennarbyträsket, which will be discussed in detail in this thesis.

The results of the field samples can be found in Table 5 and Table 6 (available in Appendix 1). The original idea of this thesis was to model levels of phosphorus. Unfortunately, the data cannot be used for statistical analysis as there are not enough replicates and not enough data points with significant variation. These samples have been used by LUVY for other purposes beyond the scope of this thesis.
It has required by LUVY to obfuscate of the exact location of the EXO2, this information will therefore not be shared in this document. The probe is about a meter away from the bottom and is attached to a wooden floating pier that is about 3 meters long.

Figure 6. Lake Gennarbyträsket underwater.

Part of my duties in LUVY, were biweekly maintenance visits, as showed in Figure 7 , ensuring that the probe was in good conditions and that there was no fouling built up on the sensors (and in the rest of the probe).

Figure 7. Maintenance work on the EXO2 sonde (image credit: LUVY)

These visits were also the opportunity to change the battery when this was required.

2.2 The EXO2 water station

The EXO2 is a digital multiparameter sonde with applications for water quality monitoring (YSI Inc./Xylem Inc., 2017). It is developed and produced by YSI Inc., part of the Xylem group. Thanks to a wide choice of probes, it is possible to follow different parameters, as we can see in Figure 8.

Figure 8. EXO2: a digital multiparameter sonde (image credit: LUVY)

The EXO2 has an availability of 7 ports where probes and accessories can be attached. The central port can be used to attach a scraper which slows down the fouling process on the sensors.

LUVY has chosen to equip the EXO2 with the probes described in Table 1 (YSI Inc. / WTW GmbH / Xylem Inc., 2017), together with a central scraper. For project-related reasons, one port remains unused.

Table 1. EXO2 probes used in Lake Gennarbyträsket.

Probe	Parameter	Range	Accuracy	Resolution
Conductivity Temperature	Conductivity	0 to 200 mS/cm	0 to 100: ±0.5% of reading or 0.001 mS/cm, whichever is greater (w.i.g.); 100 to 200: ±1% of reading	0.0001 to 0.01 mS/cm (range dependent)
	Temperature	-5 to 35°C 35 to 50°C	±0.01°C ±0.05°C	0.001 °C
Dissolved Oxygen	Dissolved Oxygen	0 to 50 mg/L	0 to 20 mg/L: ±0.1 mg/L or 1% of reading, w.i.g.; 20 to 50 mg/L: ±5% of reading (Relative to calibration gases)	0.01 mg/L
pH	pH	0 to 14 units	±0.1 pH units within ±10˚C of calibration temp; ±0.2 pH units for entire temp range (within the environmental pH range of pH 4 to pH 10).	0.01 units
Total Algae	Blue-green Algae, Phycocyanin	0 to 100 µg BGA-PC/L	Linearity: R^2 >0.999 for serial dilution of Rhodamine WT solution from 0 to 100 µg BGA-PC/mL equivalents	0.01 µg BGA-PC/L
	Chlorophyll a	0 to 400 µg Chl a/L	Linearity: R^2 >0.999 for serial dilution of Rhodamine WT solution from 0 to 400 µg Chl a/L equivalents	0.01 µg Chl a/L
Turbidity	Turbidity	0 to 4000 FNU	0 to 999 FNU: 0.3 FNU or ±2% of reading, w.i.g.; 1000 to 4000 FNU: ±5% of reading (values are automatically calculated from conductivity according to algorithms found in Standard Methods for the Examination of Water and Wastewater (Ed. 1989))	0 to 999 FNU = 0.01 FNU; 1000 to 4000 FNU = 0.1 FNU

The cost of our setup is around 20 000€ (including the different probes, options and calibration). The cheapest probe is the pH, costing around 600€; and the most expensive is the Total Algae, costing around 3450€ (representing over 17% of the total cost for the system). Luode Consulting Oy (referred in the rest of this document as Luode), is the contractor that is responsible for the sonde calibration and onsite deployment. They also provide services such as backend maintenance and support.

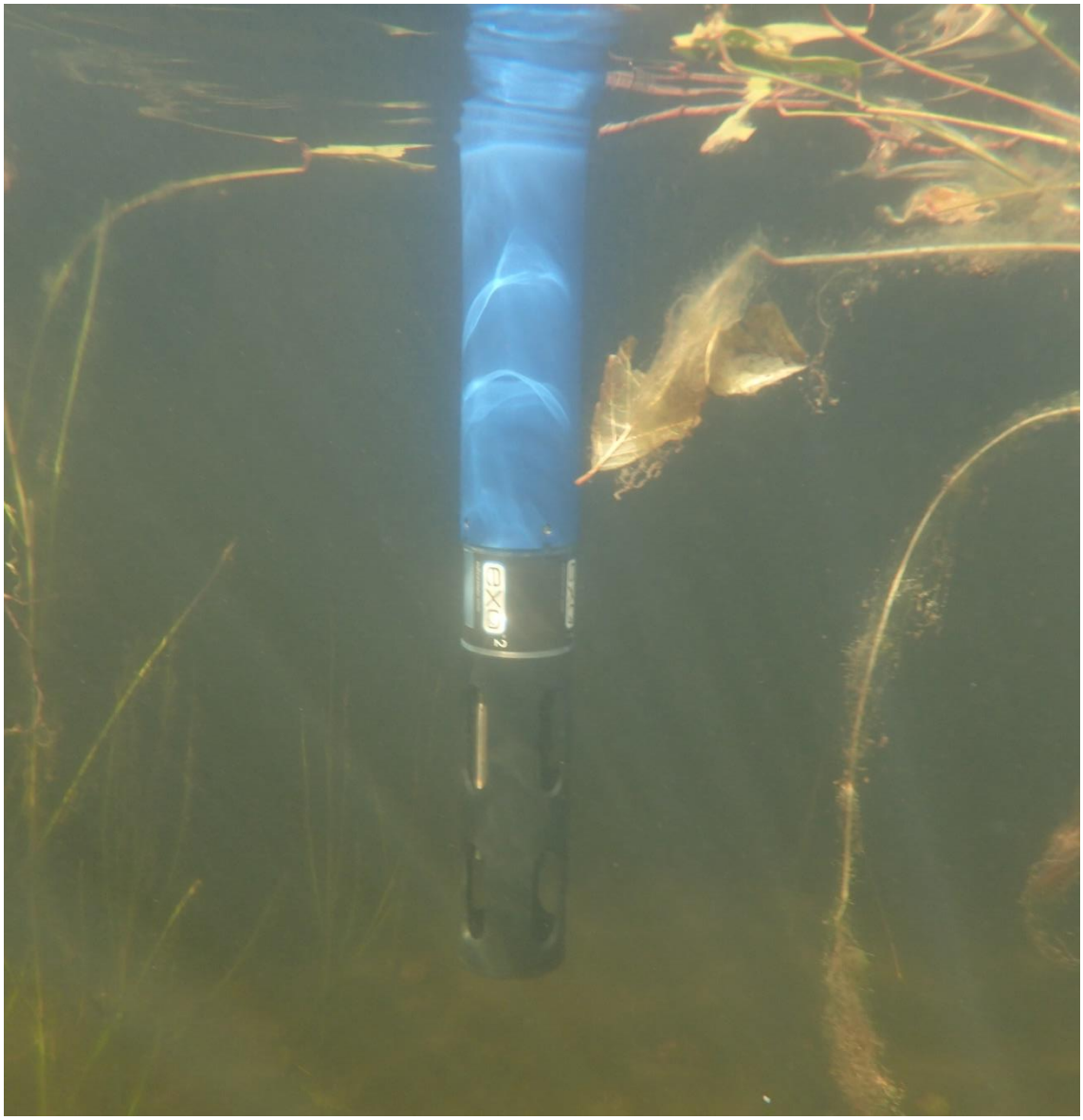

Figure 9. EXO2 underwater in Lake Gennarbyträsket (image credit: LUVY).

The probe was connected to a datalogger (acting as a backup) and a GSM modem that was sending the measurements to a server controlled by Luode. The whole system was powered by a 12V, 33Ah battery, with an autonomy of about a month.

2.3 Origin of the data

The data has been gracefully given by LUVY and by the Finnish Meteorological Institute (FMI). The data was provided as CSV files from LUVY and as excel files from FMI.

2.3.1 Water data

The data comes from the measurements taken by the EXO2 sonde in Lake Gennarbyträsket. The sampling period went from 21.07.2016 13:30, when the sonde has been deployed by Luode to 07.11.2016 12:00, when Lake Gennarbyträsket started to have an ice cover. A sample of the raw data can be found on Table 7 in the appendices.
Following the advice given by Luode, it is assumed that the standard deviation of the measurements to be the accuracy for each probe described in the Table 1. An estimation of the standard deviation has not been made by Luode during the calibration process.
Additional background data about the general morphology and characteristics of the lake has been given by LUVY.

2.3.2 Weather data

Weather data is a courtesy given by the Finnish Meteorological Institute (FMI). They are the observations from the 4 closest weather stations (located in Tvärminne, Kemiönsaari, Salo and Lohja). The dataset includes observations on temperature, dew point, relative humidity, wind direction average wind speed (in the last 10 min), maximum wind gust (in the last 10 min) and rain accumulation (from the last hour). A sample of the raw of each dataset can be found on Table 8 and Table 9 in the appendices.
Unfortunately, the weather data was not significant to use during this thesis. This will be further developed in the discussion section (page 44).

2.4 Modelling and statistical analysis

Modelling is the process of using mathematical tools to analyse, describe and/or predict the behaviour of a phenomenon. The first step of modelling is abstraction: defining variables in a mathematical way. This is typically done by setting up metrics for different parameters which can be followed (for instance pH, temperature and turbidity). When building up a model, reality is simplified and details are left out. There are two main groups of models: empirical and mechanistic. The first model type, empirical, simply describes the behaviour of a data set. Whereas the second type, mechanistic, is based on fundamental science, such as Einstein's Theory of Relativity (Berthouex & Brown, 2002).

Statistical analysis is the process of analysing a set of data using statistical tools to achieve a higher level of understanding on the dataset. This is done by interpreting the statistics calculated from the dataset in question (SAS Institute Inc., 2016). For instance,

analysing the performance of students from the same class in an exam. One can calculate the average and the median to evaluate the student's performance in the test with respect to the other students.

This thesis focuses on empirical modelling and statistical analysis of the data collected in Lake Gennarbyträsket. The assumptions are that chlorophyll depends on pH, turbidity, dissolved oxygen, conductivity and temperature; that errors are random; and that pH, turbidity, dissolved oxygen, conductivity and temperature are independent.

2.5 Relevance of this thesis

As discussed in The EXO2 water station, the Total Algae, costing around 3450€ (representing over 17% of the total cost for the system). This is a significant cost in a setup. Finding a model that can describe algae and cyanobacteria growth from parameters using cheaper probes allows to reduce costs and releases an additional port that can be used for monitoring other parameters.
Algae and cyanobacteria monitoring are of great interest for both research and for the public. In a research point of view, this can be used to assess eutrophication and water quality. The public, on the other hand, is interested in algae bloom as this might affect leisure activities such as swimming and scuba diving or water use for sauna (very popular in Finland). Finally, the model might have predictive capacities that can emit early warnings for authorities, researchers and for the public.
It is therefore profitable to search for possibilities to establish such a model.
LUVY uses chlorophyll data to assess the trophic level of lakes. The usual resolution of such a data is 3 to 15 data points per open water season. Additionally, these data points are limited by working hours and by the weather. Using the EXO2, the resolution becomes 1 data point every half an hour, independent of working hours and weather. The increase in resolution can also lead to a more precise estimation of the lake's trophic level. This thesis is also aiming to producing a tool for LUVY to use to assess the lake's trophic level from the data generated by the EXO2 algae sensor.

3 Theoretical background

Limnology is the study of lake biology, and of its physical and chemical characteristics. This thesis' focus is closely related to limnology studies. It is therefore essential to get familiarized with basic concepts of limnology, which are presented in this section.

3.1 Lake nutrient levels and eutrophication

The trophic level of a water body is the amount of nutrients present in this water body. Different water bodies have different natural trophic levels. This is especially the case of freshwater lakes. The natural trophic level depends on the lake's and its catchment's area's characteristics. Some lakes might be naturally eutrophic, for instance lakes in a clay soil area (which is typical in the South of Finland). Lake Gennarbyträsket is currently oligotrophic (having low levels of nutrients). Eutrophication (from the ancient Greek: well nourished) is the consequence of an excessive increase in nutrients concentrations in a water body. Consequently, algae biomass increases (and therefore levels of chlorophyll), which can cause blooms (Suonpää, 2015).

When the nitrogen-based nutrients are consumed (usually by the end of the summer), algae cannot continue to reproduce in the same rate. Cyanobacteria, which can capture atmospheric nitrogen, start dominating the phytoplankton biomass (composed by planktonic algae and cyanobacteria). Cyanobacteria might release toxins into the water body that can be harmful for human and animals as it is decomposed at the end of its lifecycle. The final decomposition of the phytoplankton is usually made by bacteria in the bottom of the lake. This process is aerobic, and therefore consumes dissolved oxygen. For lakes in an advanced eutrophication state over a long period of time, oxygen deprivation might occur, leading to the leaching (releasing of nutrients from the sediment layer) in an anoxic environment. In addition, the low oxygen levels might lead to the disappearance of benthic fauna in the area. The entire lake ecosystem is affected which can alter the distributions of species (Wetzel, 1983).

3.2 Chlorophyll-a

Chlorophyll is a light-absorbing pigment that can be found on every photosynthetic organism, including cyanobacteria (Miller, 2004). Activated by light, chlorophyll acts as a catalyst to a reaction between water and carbon dioxide that produces glucose an oxygen.

Equation 1.. Photosynthesis reaction

$$6CO_2 + 6H_2O \xrightarrow{Chlorophyll+light} C_6H_{12}O_6 + 6O_2$$

Chlorophyll levels in lake water are good indicators to estimate the lake's eutrophication levels as these are strongly correlated (Oravainen, 1999). Chlorophyll is usually estimated by taking water samples within the first 2 meters of the water column. The concentration of chlorophyll is weather dependent, as wind and rain might interfere in the distribution of chlorophyll in the water column. Therefore, it is common practice to take several samples during the open water (ice-free) season. The measurements are then averaged to estimate the average chlorophyll level in the lake for the entire open water season. To assess the eutrophication levels, both the average and the general evolution of the chlorophyll levels need to be studied. This method is prone to error as weather can skew the measurements (if the sampler goes on days that are rainy or windy, the measured concentration of chlorophyll will be less important as the lake water is being mixed). HELCOM recommends 15 observations to be made to evaluate the chlorophyll levels in the water (HELCOM, 2017).

Table 2. Eutrophication level guideline according to the concentration of chlorophyll-a in surface freshwater bodies.

Water quality	Excellent	Good	Satisfactory	Passable	Poor
Eutrophication level	I	II	III	IV	V
Chlorophyll-a (µg/l)	<4	<10	<20	≤50	>50

The Table 2 is a guideline proposed by SYKE to assess the eutrophication level of surface freshwater bodies (Mitikka, 2015). The chlorophyll concentration is an average of the observations from the open water season samples. Another approach can be taken to evaluate the lake's eutrophication level: analysing both the average and the individual measurements. The assessment considers the different levels of eutrophication as an average but also the frequency distribution with weather data. This not only gives a better estimate of the real eutrophication state of the lake, but also helps to understand the lake's behaviour over the open water season (Suonpää, 2017).

3.3 Factors that influence the levels of chlorophyll-a and its measurements

External factors might affect the measured levels of chlorophyll in two ways (Wetzel, 1983). The first is related to the distribution of chlorophyll in the water body, in our case Lake Gennarbyträsket. This is related to the mixing of the water and can be triggered by rain and wind. Additionally, turbulent flow and retention times influence in the mixing of

the water. Chlorophyll concentration is typically more elevated on the first meter of the water column than in its bottom. When the water is mixed, the chlorophyll is no longer concentrated on the surface layer. Consequently, measuring mixed water at the surface layer will result in lower concentrations.

Some species of phytoplankton can form colonies by aggregation. The size of these aggregates can influence the measurement of chlorophyll. This is mainly due to the number of cells present in front of the sensor and their retention time. The reading is then larger than the real concentration of chlorophyll in the lake in question. According to LUVY, there are phytoplankton species present in Lake Gennarbyträsket which are known to form aggregates. For instance, LUVY's study reveals that the phytoplankton was dominated by the species number and by the biomass by Chroococcales, which is an order of cyanobacteria, known to form aggregates. These aggregates can be visible at bare eye when a water sample is taken in a glass container against the light.

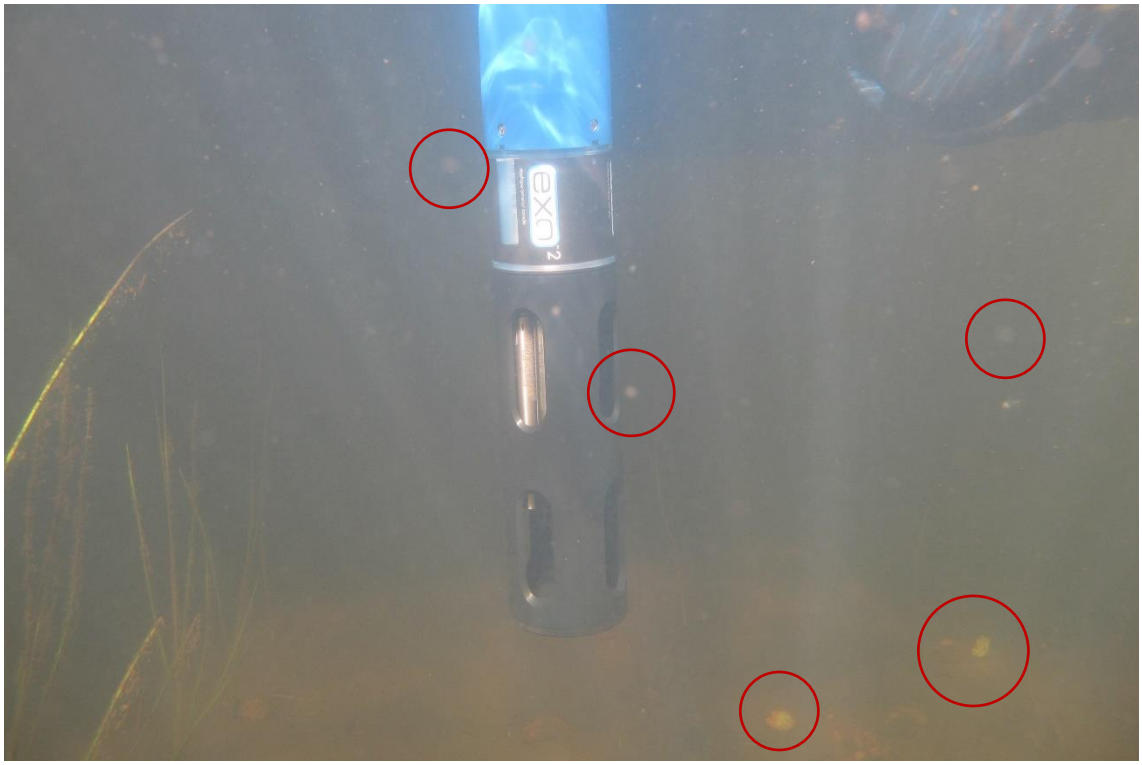

Figure 10. Underwater view to the EXO2 sonde

Particles in the water can also affect the measurements if their surface react to the light in the same way chlorophyll reacts (in the sensor's point of view). On Figure 10, we can see some suspended particles (blurry dots circled in red in the picture), some of these could be algae aggregates or particles that could interfere in the chlorophyll and turbidity measurements.

Additionally, the sensor is about one meter away from the bottom (distance varies according to the lake surface level), turbulence induced by aquatic life, pier and water movements could lead to a disturbance of the sediment layer's surface which could temporarily increase the suspend particles, leading to higher measurements than the lake's average levels. Finally, there is filamentous algae visible on the plants near the water monitoring station which could be released in the water and potentially lead to peaks in the readings of chlorophyll and turbidity levels. These are visible on Figure 11.

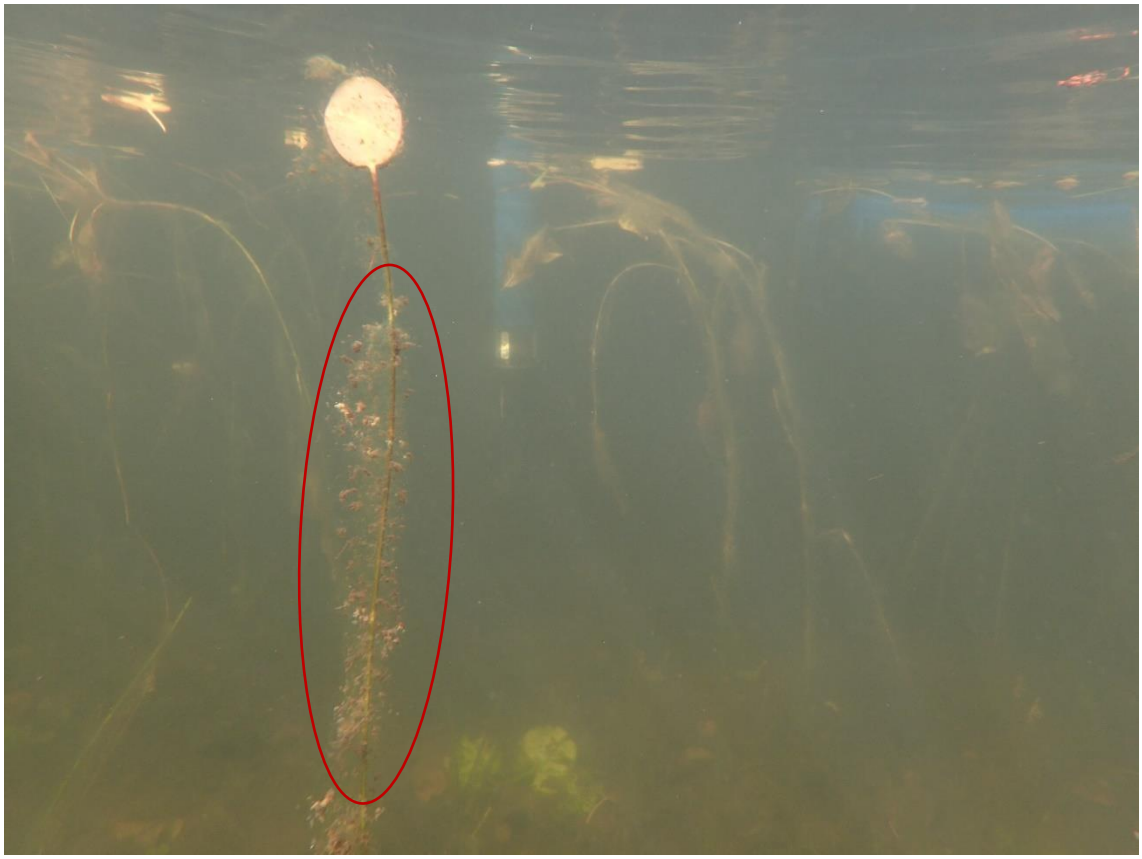

Figure 11. Filamentous algae growing on an aquatic plant, the EXO 2 sonde can be seen in the background.

The second factors are related to the growth of phytoplankton. They are mainly nutrients, dissolved carbon dioxide, water temperature and solar irradiance. Nutrients are divided into three groups: silica, nitrogen and phosphorus. Each nutrient group can be a limiting factor for the growth of phytoplankton. Weather factors can also influence in phytoplankton growth, for instance, the amount of light available decreases as cloudiness increases, leading to an inhibition of phytoplankton growth. Phytoplankton growth is known to have diurnal changes. Phytoplankton grows more in the morning, less in midday and more actively in the evening and at night the light availability limits the growth (Wetzel, 1983).

Seasonal changes also affect chlorophyll measurements. Summer has longer and typically warmer days, exposing the phytoplankton to more light and warmer temperatures, which promotes growth; while winter has short and typically colder days, inhibiting growth. The fall is also typically marked by storm episodes, which mix the water column, and spring has typically ice melting, increasing the inflow of water that can carry extra nutrient loads.

Environmental factors might also influence in the growth of phytoplankton. Fertilizers used in agriculture and emissions from transportation can increase the levels of nutrients when these are carried by the runoff. Additionally, wastewater that is not properly treated can also be a source of nutrients.

4 Methodology

This section will present the different tools and approaches used during this thesis. It will first present some data loading and pre-treatment methods. Followed by statistical analysis and modelling methods.

4.1 Making data compatible with R

The first step to be performed on the raw data is to make sure its structure is compatible with R. The data comes from different sources with different formatting and this could be problematic when using R. The ideal is to have the whole dataset in the same format. There is a wide variety of format available, I chose to use CSV following the RFC 4180 standard (Internet Engineering Task Force, 2005) as default format. This standardization process allows me to have consistency over the data and to facilitate the work with the data.

The time format in the datasets is formatted differently and is incompatible with R. The next step is to make the time R-friendly using strptime. This function returns a timestamp in the POSIXlt format, that is the number of seconds since the beginning of 1970 in the UTC time zone (R Reference, 2001). This means that 0 in POSIX represents the 1st of January 1970 at 00:00:00 in the UTC time zone. This is not practical as 0 in POSIX time represents the 1st of January 1970 at 02:00:00 in the Finnish time zone, and the first sampling date is the 21st of July 2016 at 13:30 in the Finnish time zone with Daylight Saving Time, which is in POSTIX 1469097000. To simplify date calculations, we set the origin to the first sampling date. This is done by subtracting 1469097000 to the POSTIX time. The result is the number of seconds since the first sample has been taken. Finally,

we calculate the amount of days elapsed since that original date (by dividing the number of seconds by 86400, which is the number of seconds in a day). The result is a decimal figure that represents how many days have elapsed since the first sample. Note that the sampling rate is 30min for the water station and that the origin of our time is not the starting point of a day. The time information is then added to the data frame so that we can use it later.

4.2 Time series

Time series are composed of data points that are ordered with respect of time. They reveal the behaviour of the data over time. It is important to first plot the timed data as a time series to get the feeling of how the data behaves over time (Mac Berthouex & Brown, 2002). Plotting the time series might also reveal outliers in our data, these might be seen as jumps for individual values. Nevertheless, one needs to be careful when assuming a jump is an outlier. A jump can be considered as an outlier if it exceeds the expected variation of the data. Let us set up a scenario as an example. Let's suppose that the pH of a lake is measured every minute. Let's suppose that plotting the data we get the Figure 12 with values fluctuating around 7.6, except for one: 13.8. It is very unlikely that the pH of a lake can go from 7.6 to 13.8 and back to 7.6 in a couple of minutes. The measurement indicating 13.8 is most likely an outlier. The outlier can also be a lower value, such as 1.6 in the graph of the example.

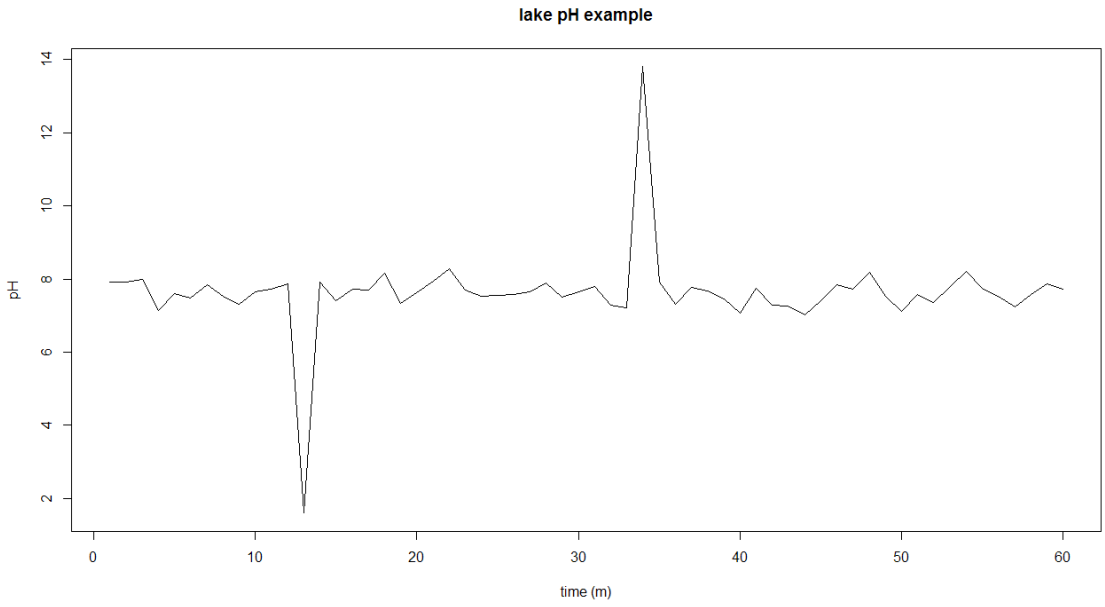

Figure 12. Lake pH example.

Overlapping time series can reveals periodical behaviour. It also helps to determine the periodicity of a phenomena.

4.3 Correlation matrix

Correlation is an estimation of the linear relationships of a group of variables. A correlation matrix is a representation method to show pair-wise correlations in a group of variables. There are different approaches to estimate correlation coefficients Depending on the approach, the output can be statistically more or less robust to outliers. The Pearson correlation coefficient is most used method to estimate correlation. This method requires normally distributed data and is very sensitive to outliers and therefore not so robust. A more robust approach would be to use the Spearman correlation, which first calculates the ranks of each variable and then the Pearson correlation coefficient of these ranks. This approach estimates how the two variables follow each-other in continuous intervals. Because of the random nature of the data and taking into consideration all the possible interferences described in the previous chapter, a more robust approach is preferable. This thesis is therefore using the Spearman correlation to estimate relationships in the data.

4.4 Variable standardization

Variable standardization is applied to all the independent variables before processing with regression models to make them unite-less and comparable (Albers, 2017 (to be published)). This standardization process also eases computer calculations (Sokal & Rohlf, 1995). There are different approaches for variable standardization. In R, there is a function called scaling that can be used for variable standardization.
The formula R uses for its scale function is:

Equation 2

$$X_i = \frac{x_i - \bar{x}}{\sigma}$$

This equation corresponds to a standardization using the Z-scores (Comprehensive R Archive Network, 2017).

- x is the set of measurements
- X_i is the standardized, dimensionless, value (corresponding to the Z-score) of the ith value of x
- x_i is the ith value in physical unit of x

- $\bar{x}$ is the mean of all measurements of x
- σ is the standard deviation of all the measurements of x

This R function allows us to choose to centre and scale the data.

A more robust standardization method uses the median and the median absolute deviation (MAD).

Equation 3

$$X_i = \frac{x_i - \tilde{x}}{MAD(x)}$$

Where:

- x is the set of measurements,
- X_i is the standardized, dimensionless, value (corresponding to the Z-score) of the i[th] value of x,
- x_i is the i[th] value in physical unit of x,
- $\tilde{x}$ is the median of all measurements of x
- $MAD(x)$ is the median absolute derivation of all the measurements of x

In R, this can be achieved with the formula:

```
# x is a dataframe
X <- scale(x,
           center = apply(x, 2, median),
           scale = apply(x, 2, mad)
)

# x is a vector, for instance c(1,2,3,4,5,6,7,8,9,2,2)
X <- scale(x,
           center = median(x),
           scale = mad(x)
)
```

This approach is advised for data containing possible outliers (Varmuza & Filzmoser, 2009b).

4.4.1 Centering

In the approach used in Equation 2, centring is done by subtracting the mean of the set from each value. This sets the mean of the independent variables to be 0, making the intercept of the model to be the expected value when the independent variables are set

to their mean. The median becomes 0 after centering when using the approach in Equation 3. Both scenarios make sure that the intercept represents a realistic scenario, within the boundaries of each independent variable (assuming they are independent to each-other).

4.4.2 Scaling

Scaling following the approach of Equation 2 is done by dividing the (centred) values by the standard deviation of the original set. Because of scaling, the standard deviation of the scaled variables is 1 and the units are cancelled, leaving the output unite-less. The approach of Equation 3, sets the MAD to be 1. In both cases, this is done to make data with different magnitudes comparable.

4.4.3 Effects of standardization on graphs

Standardization does not change the shape of the curve/surface (Mortenson, 1985). Observing scaled data therefore give us insights on how the data behaves (its trends). When comparing the graphs of two standardized variables, we can see if they are correlated or not. Graphs that look alike are correlated.
Let us have an example. Two data series A and B.
Plotting each series individually, we get a feeling that Figure 13. Plot of series A and Figure 14. Plot of series B. Plotting both series in the same plot (Figure 15), we realise that they have different scales. We can clearly see the behaviour of B but A looks like a straight line (which is not true). With closer attention on the individual plots, the reason get clear. The data from A vary within the interval [-1, 1] while the data from B varies in the interval $[-10^6, 10^6]$. Standardizing the data of both datasets and plotting it again (Figure 16) makes it easier to compare, it is now clear that A and B are strongly correlated (as they behave mostly similarly).

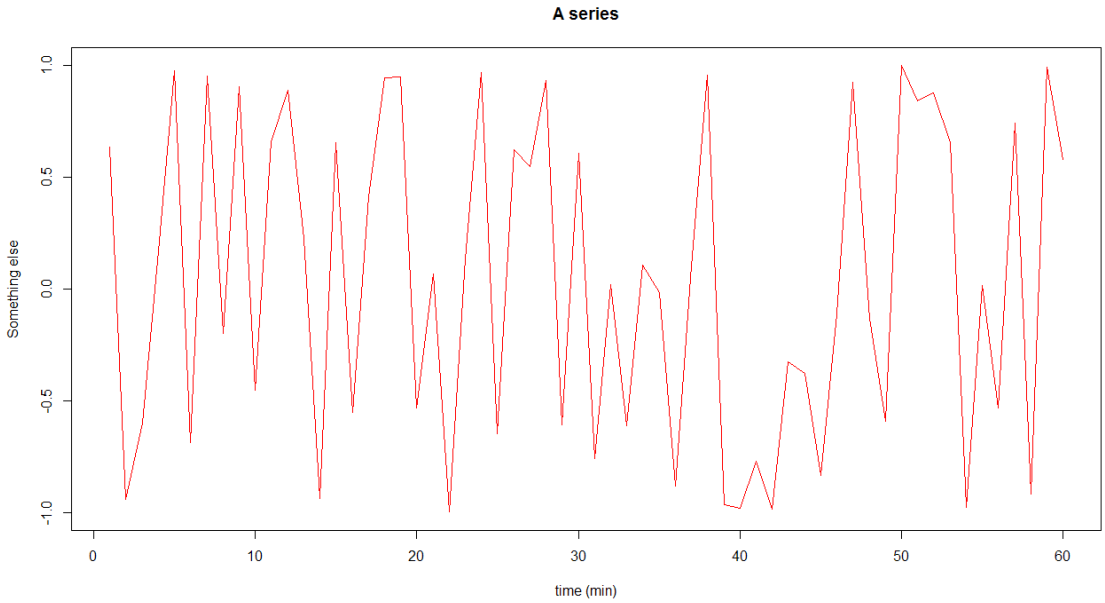

Figure 13. Plot of series A

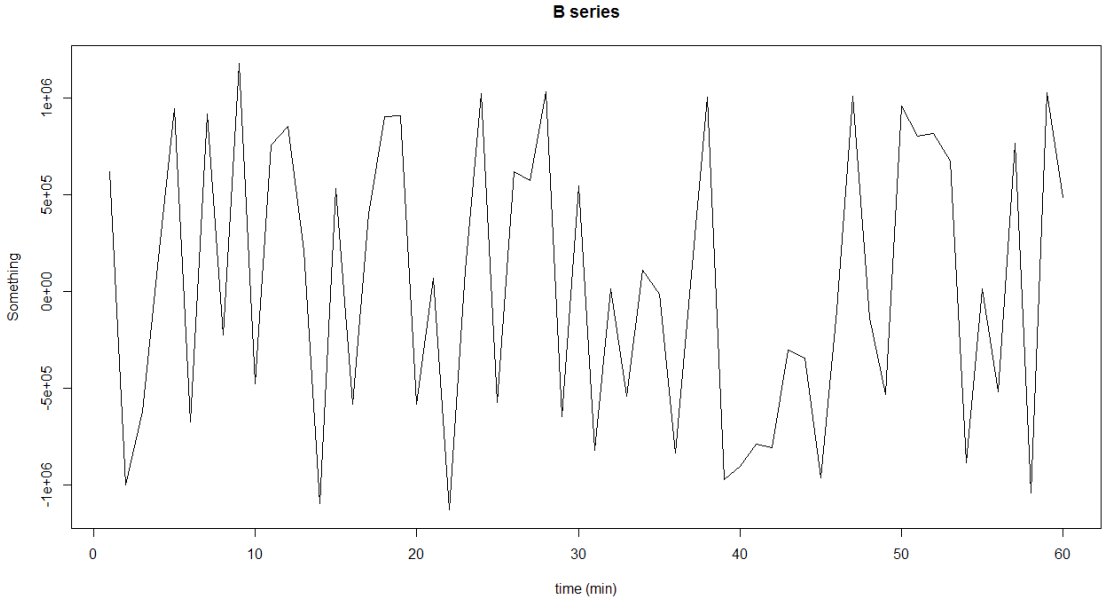

Figure 14. Plot of series B

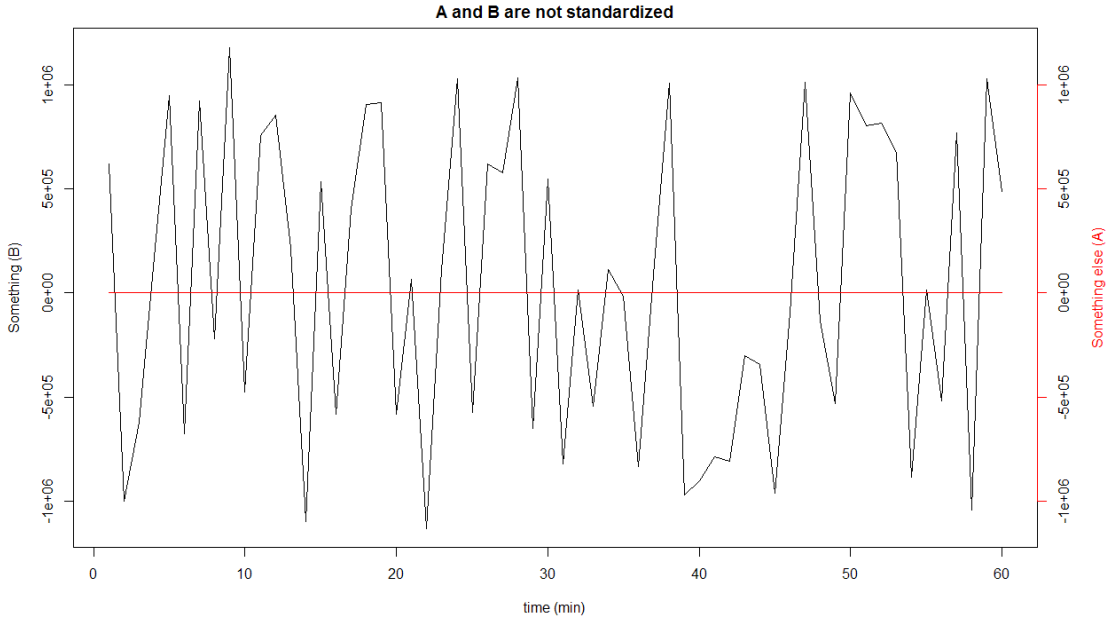

Figure 15. A and B not standardized

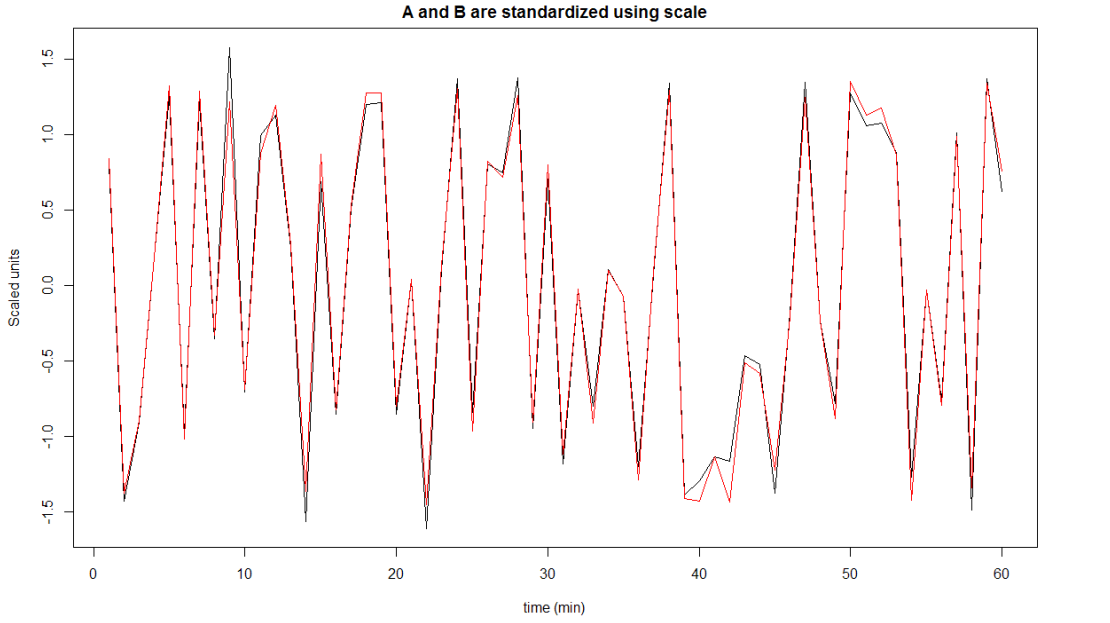

Figure 16. A and B standardized using scale

For verification, making a correlation test between the dataset A and B estimates the correlation coefficient of 0.995 (the closer to 1, the stronger is the correlation). But these are off the scope of this thesis.

4.5 Smoothing

Smoothing is done to reduce the local variations in the data while keeping the overall tendencies. When smoothing variations are flattened, making the data look smoother, with less peaks. This also reduces the impact of outliers, as these are diluted in the overall tendency of the data. Note that smoothing operations introduce lag to the data.

Local polynomial regression fitting (loess) is a local regression method that makes local fits of a polynomial surface (or line) based on the weighted distance to their neighbours, the neighbour region and the weights are defined in the loess function parameters (Sawitzki, 2009a).

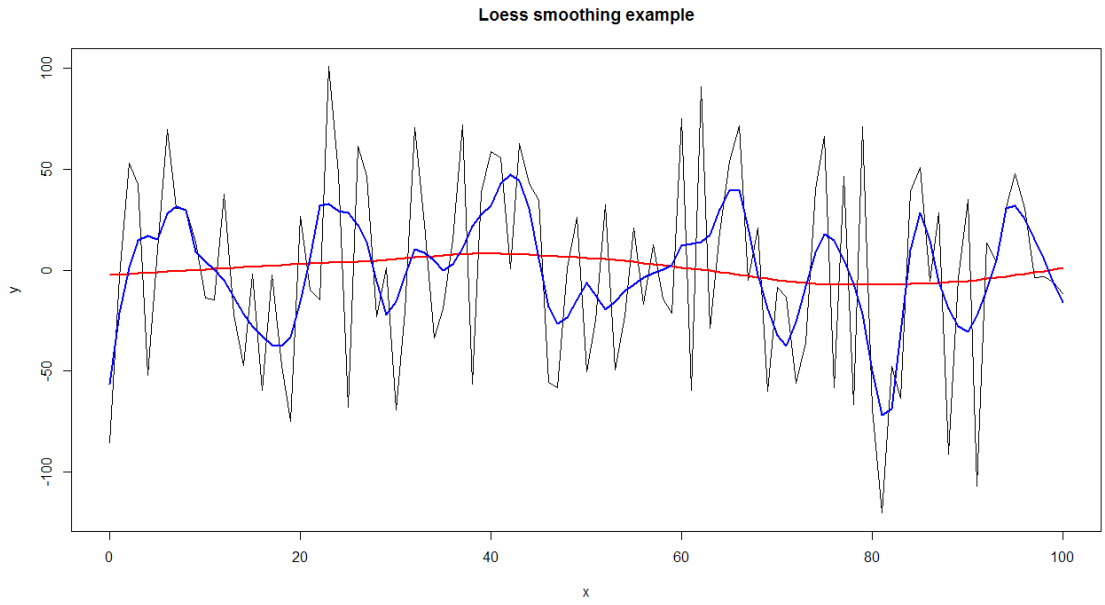

Figure 17. Example of smoothing data using loess

The example in shows data with noise (black line) and two smooth curves using loess (red and blue). The smoothing level is dependent on the objective of the smoothing operation. If it is to reduce noise, the blue curve describes the data behaviour better than the red curve. If the variations are not to be taken into consideration, the red curve might lead to a better smoothing.
This thesis uses loess to smooth the data.

4.6 Wilkinson-Rogers notation

R uses the Wilkinson Rogers notation for model formulas (Sawitzki, 2009b). This notation omits by default the intercept and the residual error. The notation also omits variable coefficients.

Here are the main Wilkinson Rogers notation rules:
The '~' sign means "the left side depends on the right side".
Example:
Y ~ X (Wilkinson Rogers notation) is equivalent to $Y = b_0 + b_1 X + e$
Here, X is the independent variable and Y is the dependent variable.
If there is more than one dependent variable, we can separate them using a coma:
For instance: Ya, Yb, Yc ~X

To exclude the intercept from the formula we can either write a 0 or -1.
Example:
Y~X-1 (Wilkinson Rogers notation) is equivalent to $Y = b_1 X + e$

Terms are added by using the '+' sign.
Example:
Y ~Xa + Xb – 1 Wilkinson Rogers notation) is equivalent to $Y = b_1Xa + b_2Xb + e$

The '*' sign is not a multiplication, but rather a short hand notation for the sum of the variables plus the term-wise interactions.
Example:
Y ~Xa * Xb – 1 Wilkinson Rogers notation) is equivalent to
$Y = b_1Xa + b_2Xb + b_{1,2}Xa*Xb + e$

To exclude a term we use the '-' sign.
Example:
Y ~Xa * Xb – Xb Wilkinson Rogers notation) is equivalent to
$Y = b_0 + b_1Xa + b_{1,2}Xa*Xb + e$

Use ':' (colon symbol) to represent term-wise interactions.
Example:
Y ~ Xa : Xb – 1 Wilkinson Rogers notation) is equivalent to $Y = b_{1,2}Xa*Xb + e$

To specify that the expression should be assessed as it is we use I(expression).

Example:

Y ~I(X^2) Wilkinson Rogers notation) is equivalent to $Y = b_0 + b_1X^2 + e$

4.7 Multivariate data

Multivariate data is a set of data containing multiple columns and rows (Varmuza & Filzmoser, 2009c). The columns typically represent different objects classes, for instance pH, turbidity, conductivity and so on. And each row represents a set of values. The columns become mathematical dimension, and a point in the space of the dataset can be defined by coordinates having the point's row values in the table. Each column is a variate, that is a quantity that is represented by values, because there are multiple columns, the data is classified as multivariate.

Example:

	pH	Turbidity (NTU)	Conductivity (mS/cm)	Colour
Point 1	7.3	3.6	53	White
Point 2	12.6	174.9	128	Red

The colour can be coded as follow White = -1 and Red =1.

Point1 can be defined as follow: [7.3 3.6 53 -1].
And point2 can be defined as [12.6 174.9 128 1]

Visual representations of multivariate data with more than 3 dimensions is very challenging. Instead, mathematical tools and analogies are used to represent multivariate data.

4.8 Multiple Linear Regression

Linear regression consists on fitting linear models in a cloud of points while minimizing the distance (error) between the points and the line (Varmuza & Filzmoser, 2009a).
Simple linear regression the response of the model depends on a single independent variable and an intercept:

$$y \sim b_0 + b_1 x + e$$

b_0 is the intercept and b_1 is a regression coefficient, both are unknown.
x is the independent variable and y is the response.
e is the residual

A dataset measurements of x and y becomes a system of equations with two unknowns. The solution of this system of equations is the model that describes an approximation of y with respect to x.

The general form for regressions is:

$$y \sim f(x) + e$$

Where $f(x)$ can be linear or nonlinear.

Multiple regression lays in the same principles, except that there is more than one independent variable.

The formula for multiple regression analysis is then:

$$y \sim b_0 + \sum_{i=1}^{m} b_i x_i + e$$

Pair-wise interactions and quadratic terms can be added to the model:

$$y \sim b_0 + b_1 x_1 + \cdots + b_m x_m + b_{1,2} x_1 x_2 + \cdots + b_{m,m-1} x_m x_{m-1} + b_{1,1} x_1^2 + \cdots + b_{m,m} x_m^2 + e$$

Where b_0 is the intercept, $b_1 \ldots b_{m,m}$ are the regression coefficients, $x_1 \ldots x_m$ are the independent variables, m is the number of independent variables, e is the residual and y is the response. The Wilkison Rogers notation is : $y \sim x_1 * x_2 * \ldots * x_m + I(x_1)^2 + \cdots + I(x_m)^2$

This thesis focuses on multiple linear regression.

4.9 Training and testing sets

The dataset is divided in two parts, one for training and one for testing the model. The division has been made by splitting the data into two datasets: one containing 10% of the full data, points chosen randomly, used to test the model; another containing the rest of the data (90%), used to train the model.

This is one strategy to implement assisted learning (Varmuza & Filzmoser, 2009a).

4.10 Cross validation

Correlated variables influence the behaviour of a model's p-values. Because of this, it is necessary to choose another strategy to select the variables that will be used in the model. Because of that it is important to use cross validation to assess the model fit (Taavitsainen, 2017).

Cross validation consists on slicing the data in segments of equal sizes that are going to be used for model training and validation (Varmuza & Filzmoser, 2009a). This guarantees that the model complexity is taken into consideration during training. When using cross validation, multiple models are tested, the optimal model is then selected.

4.11 Variable selection

Variable selection consists on reducing the number of variables to obtain a model that has a good fit while optimizing prediction qualities (Varmuza & Filzmoser, 2009a). Variable selection decreases the fitness of the model (R^2), but increases the level of significance of each variable. Variable selection also helps to reduce the needs in computing power as models get more concise. Additionally, model complexity is reduced making the model easier to interpret.

The objective of variable selection is to keep only variables with high significance levels (coefficient p-values < 0.1).

For this thesis, the variables were first selected using a cross validation forward selection followed by manual backwards elimination.

4.12 Principal Component Analysis

The basic principle behind Principal Component Analysis (PCA) is dimension reduction of multivariate data using projections (Varmuza & Filzmoser, 2009c). Data is projected from the data space onto lines, reducing the number of dimensions. The strategy behind PCA is to find directions that maximize the variance of the points in the projection line. Each direction is called Principal Component (PC) and there are as many PCs as there are variables. The components are ordered from the one representing the most variation to the one representing the least variation of the data. The components are also orthogonal to each-other. The idea is to first find the Principal Component 1 (PC1) to then find the Principal Component 2 (PC2), the direction which is orthogonal to PC1 where the projection of the data represents the most variance. We repeat the process until all the possible directions are covered. This yields to principal components denoted as PCn, where n is the number of the principal component. We can imagine the PCs to be camera angles where we can gasp part of the shape of the data. The direction that the camera is set is called rotation. Each PC is associated to a rotation.

One could make an analogy of PCA with hand shadow games (see Figure 18). The subject's hand has a 3-dimensional shape which is projected onto a wall. With the correct alignment between the hand and the light source, the shadow reveals a shape. The audience is then able to associate the shadow with a well-known image. If the subject changes the angle of the hands, the shape becomes meaningless. To perform this ex-

periment one needs an object (for instance hands), a projector (light source) and a projection surface (for instance a wall). In PCA one needs an object (the data “plotted” in the dataset’s space), a direction (the rotations from each PC) and something to receive the projection (the PC themselves).

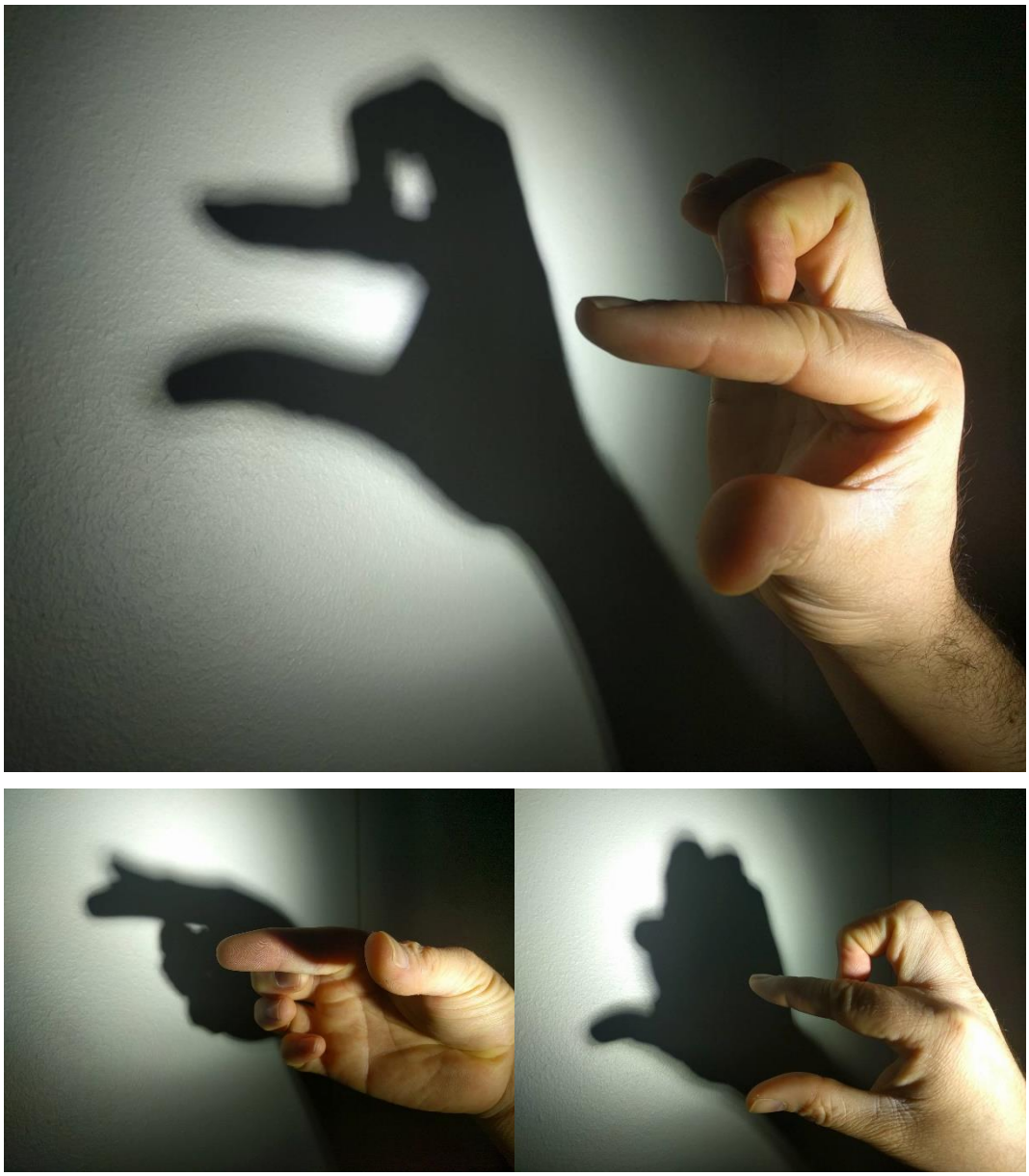

Figure 18. Example of hand shadow game

This process enables analysis of the data from different perspectives. Each perspective might explain some characteristics of the data. Each PC represents a part of the total variation of the data, and each variable has stronger or weaker influences on the data’s variation in that component.

4.13 Process diagram

A process diagram for the production of the overlapping time series and Multiple Linear Regressions can be found in Figure 35, available in the appendices.

5 Results

The results of this thesis are divided in 5 sub-sections. The first one presents the results and interpretations from the time series and smoothed series. Then the tool built for evaluating the trophic level of lakes will be presented, this is accompanied with the estimation of the trophic level of Lake Gennarbyträsket. The findings from the covariance matrices are presented in the sub-section, followed by the principal component analysis results. Finally, the multiple linear regressions will have their outcome presented.

5.1 Time series and smoothed series

In this section, we will be analysing the data using the graphs from Figure 36 to Figure 63 in the appendix. A smaller version of the relevant plots has been added to the text.

5.1.1 pH

During the first 70 days, the pH oscillates around 7.4, the peak amplitude of the oscillation is about 0.6 units.
Between day 70 and day 80, the pH and the oscillation peak amplitude decrease. From day 80, onwards, the peak amplitude is rather small, around 0.2 and the pH stabilizes around 7.0. This is most likely related to the seasonal change, as fall starts around day 75. At fall the temperature and sun light decrease, this leads to a decrease of the biological activity in the lake water. The reduced biological activity limits the variation of dissolved carbon dioxide, which is one of the driving factors of lake water pH (Wetzel, 1983).
The weekly overlapping time series clearly shows a difference on the behaviour of pH during the summer compared to its behaviour during the fall. During the summer, the pH evolves together with the daylight, around noon the pH is at its highest values and during the night time it is at its lowest. During the day, CO2 is utilized in photosynthesis which increases pH levels, and at night, CO2 increases due to respiration and pH decreases (Wetzel, 1983).

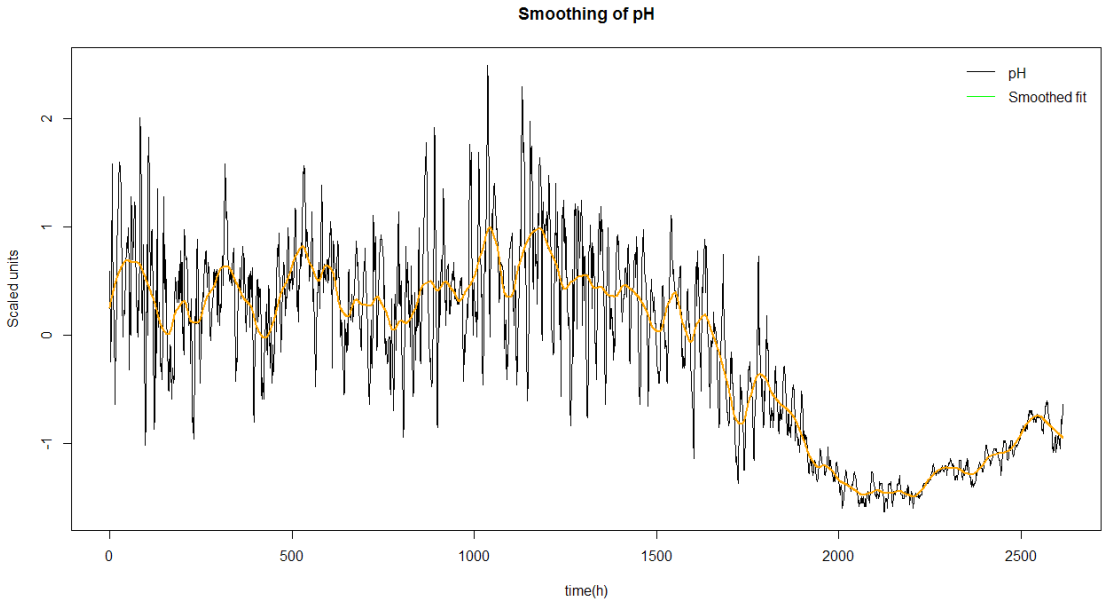

Figure 19. Smoothing of pH (more detailed view available in the Appendix)

At the bottom of the graph we can see the behaviour of pH during fall. As stated before, it is rather flat with minimal variation.

5.1.2 Conductivity

The conductivity time series shows little variation in the observations in the summer season. The values fluctuate around 59.5µS/cm with a peak amplitude of about 1µS/cm. The conductivity clearly changes at day 80 to stabilize around 61.5µS/cm.

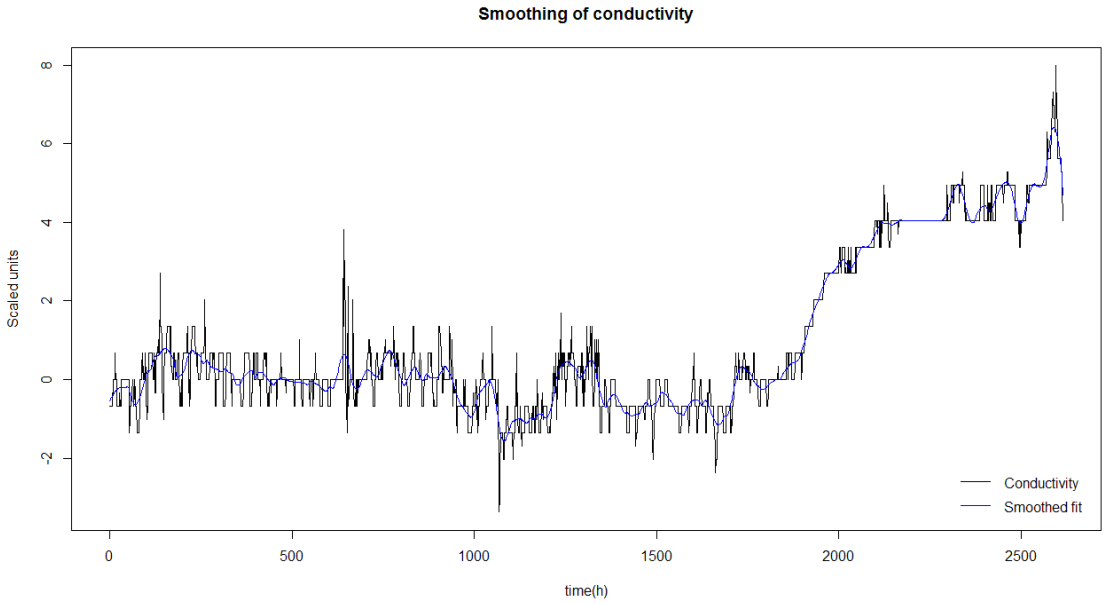

Figure 20. Smoothing of conductivity (more detailed view available in the Appendix)

These changes in conductivity are most likely not significant to affect the water quality, nevertheless there must be a reason for the seasonal changes that have been observed.

Conductivity is linked to temperature and salt dissolution constant. When the temperature decreases, the solubility of the salts also decreases (University of California, Davis, 2016).

5.1.3 Temperature

There is a clear outlier in the temperature reading around the day 12.5 (which might correspond to a maintenance operation of the sonde). Water temperature follows a daily variation. The amplitude of the variation seems to decrease when the water temperature is below 10°C.

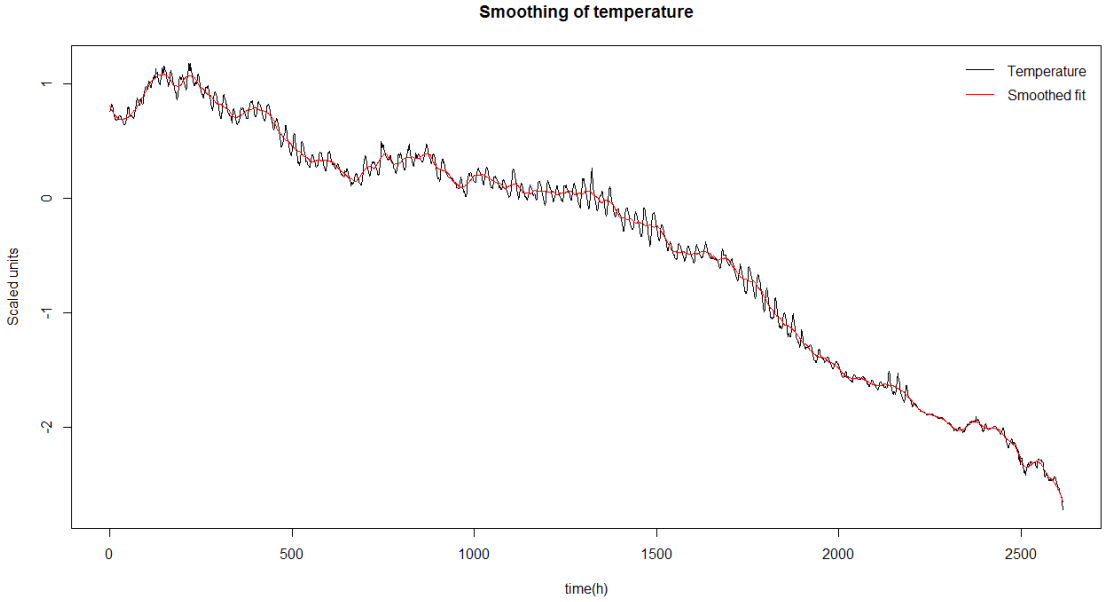

Figure 21. Smoothing of temperature (more detailed view available in the Appendix)

Temperature is related to other parameters described in the following sections.

5.1.4 Dissolved Oxygen

Dissolved oxygen also varies over the day. This is most likely due to the biological activity in the lake. By day 80 the behaviour seems to change and the average level of dissolved oxygen seems to increase.

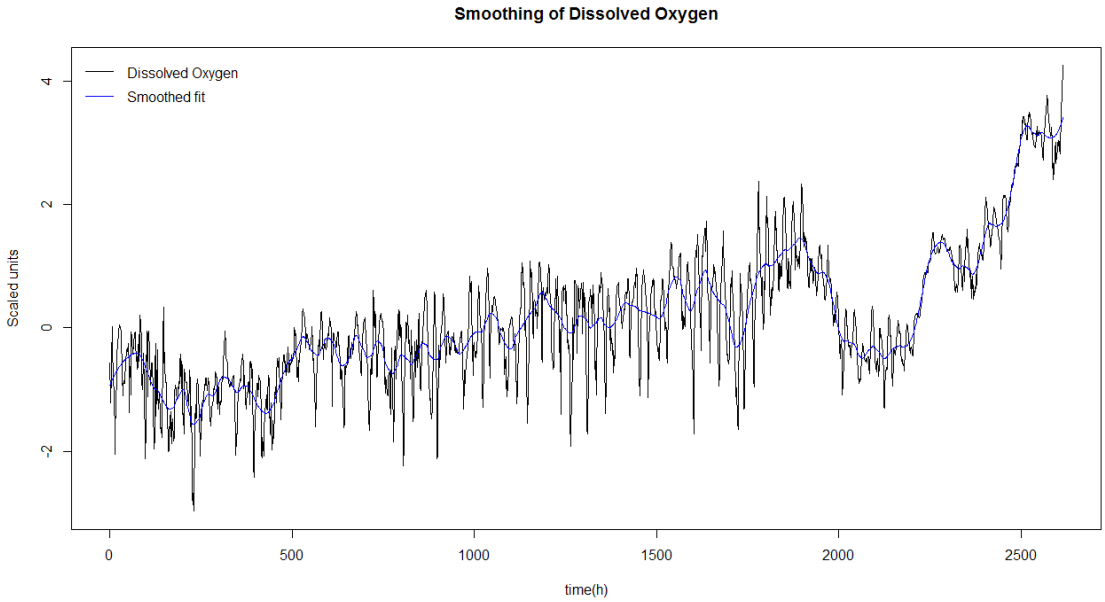

Figure 22. Smoothing of dissolved oxygen (more detailed view available in the Appendix)

This is related to a decrease in biological activity coupled with an increase in the oxygen storage capacity of the lake, colder temperatures facilitate the dissolution of oxygen in water (University of California, Davis, 2016; Wetzel, 1983).

5.1.5 Chlorophyll-a and Turbidity

Chlorophyll and turbidity seem to have a similar behaviour. Both have peaks that might be outliers. The reason for these peaks is most likely related to the size of the aggregates certain algae species might form.

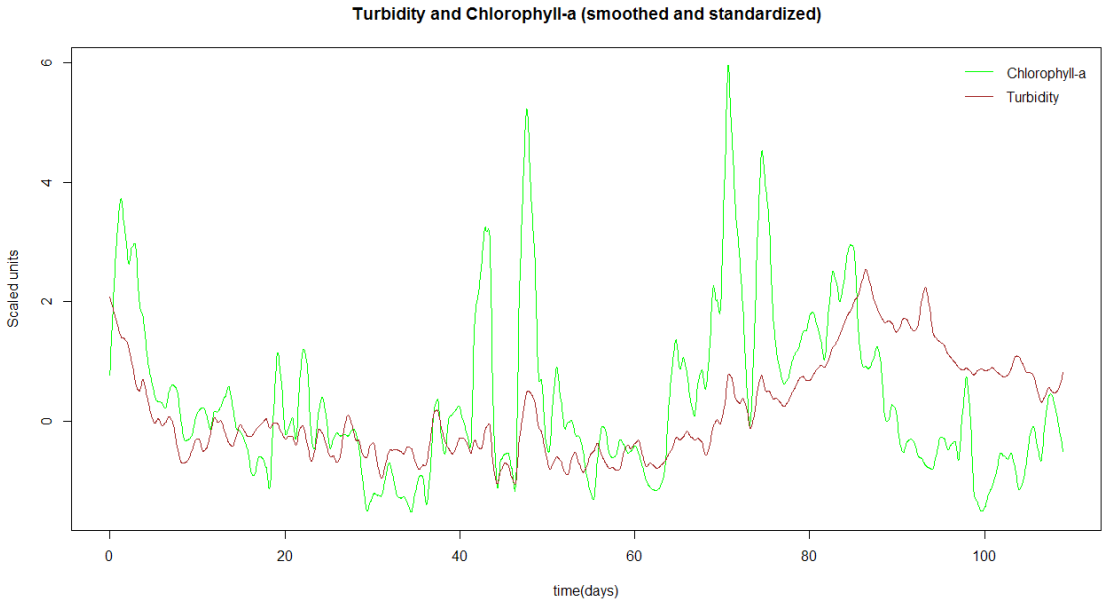

Figure 23. Turbidity and Chlorophyll-a smoothed (more detailed view available in the Appendix)

When comparing the chlorophyll and the turbidity (standardized and smoothed), we can suspect that these might be strongly correlated.

Again, seasonal change induces in a change in the behaviour of the chlorophyll-a (Wetzel, 1983). At fall the turbidity is no longer strongly correlated with the chlorophyll. Summertime the turbidity value most likely represents the biogenic turbidity caused but phytoplankton biomass (seen in chlorophyll). During the fall the biological activity decreases. Then turbidity values might be related to water mixing and weather patterns. During the fall, it is typical to have storm episodes that might increase the particle load from the surface runoff (Oravainen, 1999; Wetzel, 1983).

5.2 Evaluation of Lake Gennarbyträsket trophic level

A tool to evaluate the trophic levels in lakes has been developed in R following the guidelines provided by LUVY during an interview with a water specialist and in accordance to SYKE's trophic levels definition presented in Table 2.

The tool takes chlorophyll data as an input (from raw data or model predictions) and calculates the percentage of the time in which the lake was within a specific trophic level.

The evaluations are done for the full data collection period. The tool yields a graph with useful information, a series of boxplots that show the distribution of the data for each level, a plot displaying the time series and a plot displaying the same time series where the points are colour-coded according to the SYKE classification.

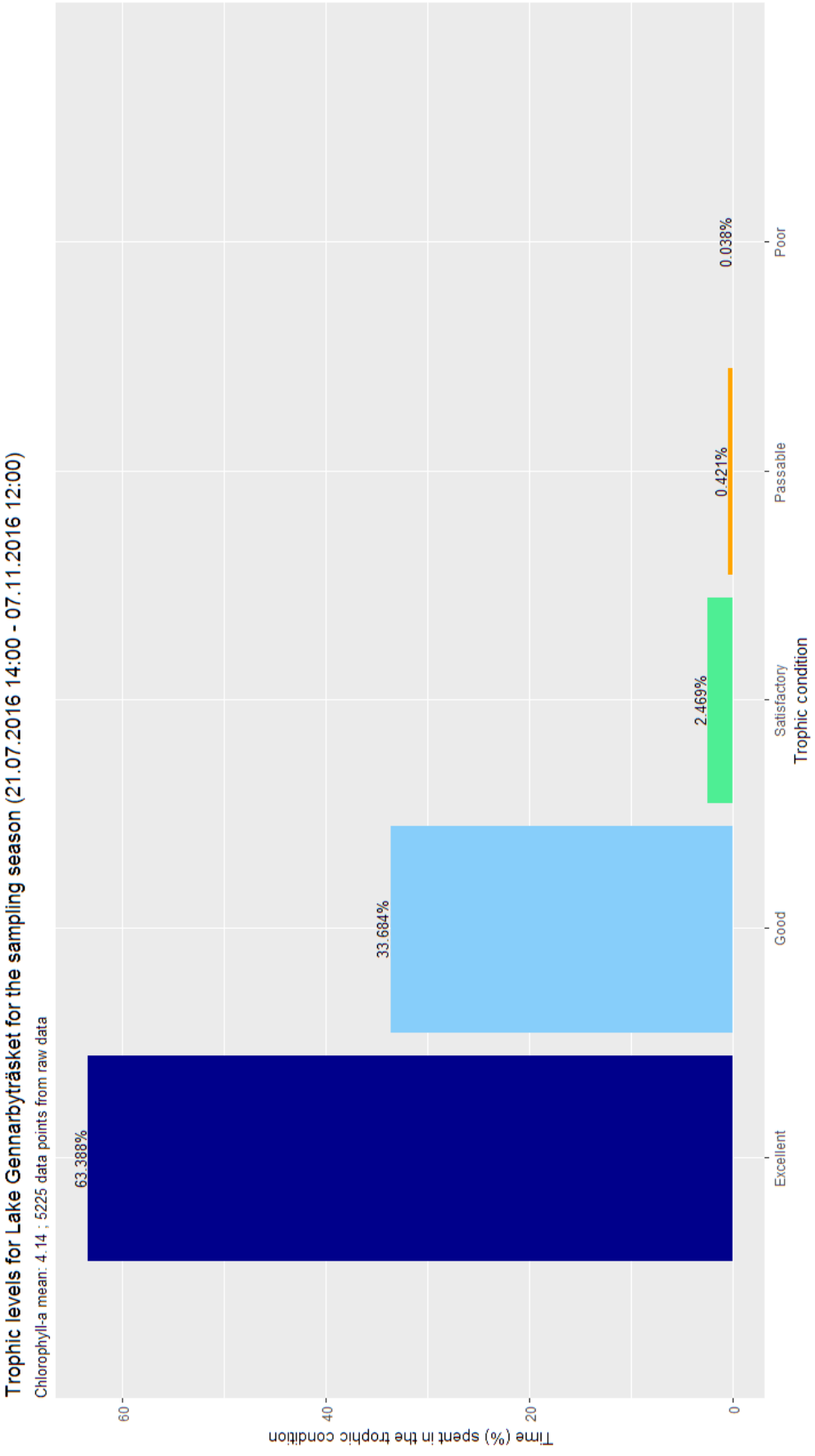

Figure 24. Bar plot output from the water quality evaluation tool.

The output of the tool using the data from Lake Gennarbyträsket can be seen in Figure 24. Based on this graph it can be said that the lake's trophic quality level from the 21.07.2016 14:00 to 07.10.2016 12:00 has been excellent 63% of the time, good 34% of the time and satisfactory and below satisfactory for 3% of the time. Therefore, the lake's water quality with respect to the trophic levels can be considered very good over the monitoring period.

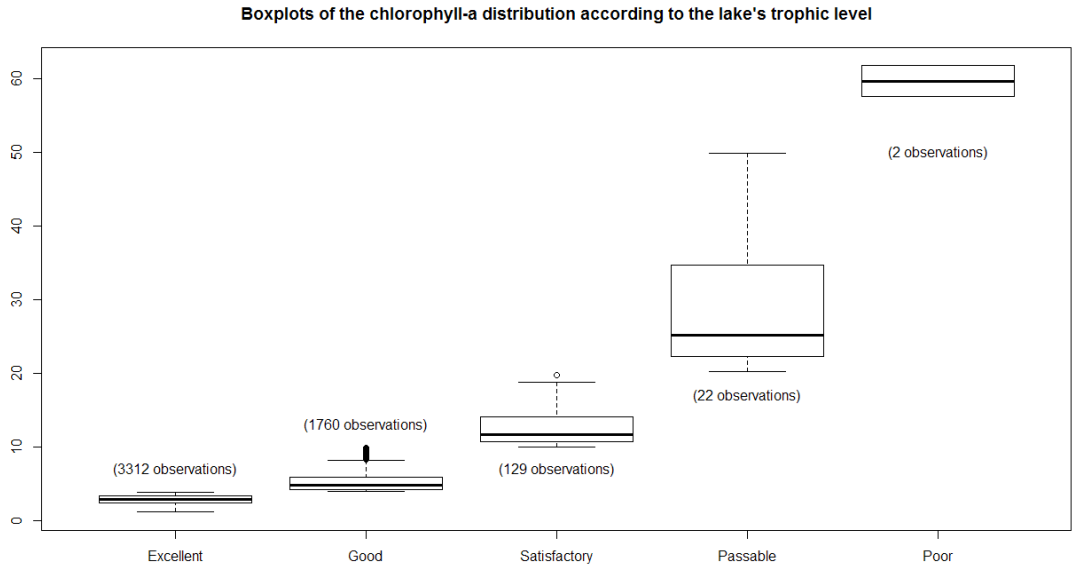

Figure 25. Box plot output from the assessment tool

On Figure 25, we can see how the data varies for the different water quality classifications. We can see from this graph that the data has tendency to be in the lower end of each classification group. This can give a more detailed reading of the scale proposed by SYKE. One could see that the lake is for instance in the mostly in the lower end of the satisfactory level whereas in the SYKE classification the level would just be satisfactory. Note that the size of the boxes is not related to the number of observations that they describe. Instead they are related to the minimum and maximum values of each classification. To avoid any confusion a text describing the number of observations has been added to the plot.

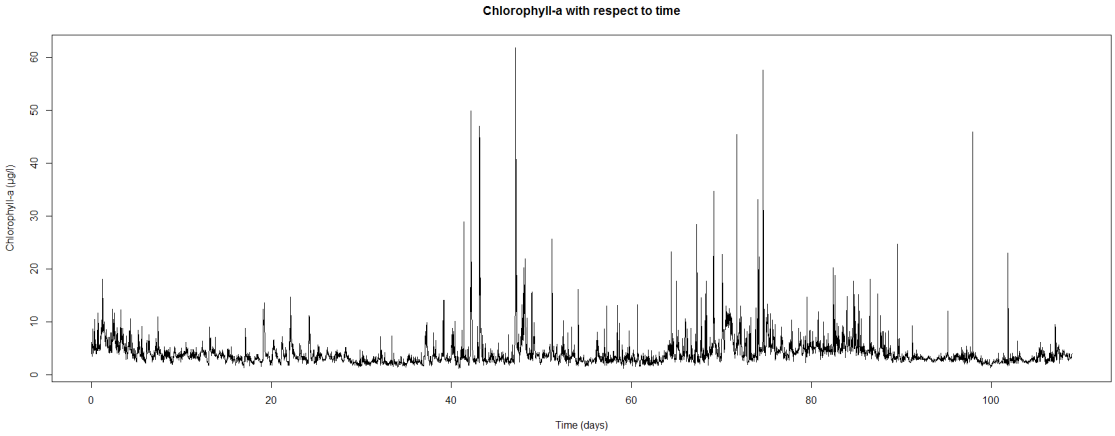

Figure 26. Chlorophyll-a levels over time from the assessment tool

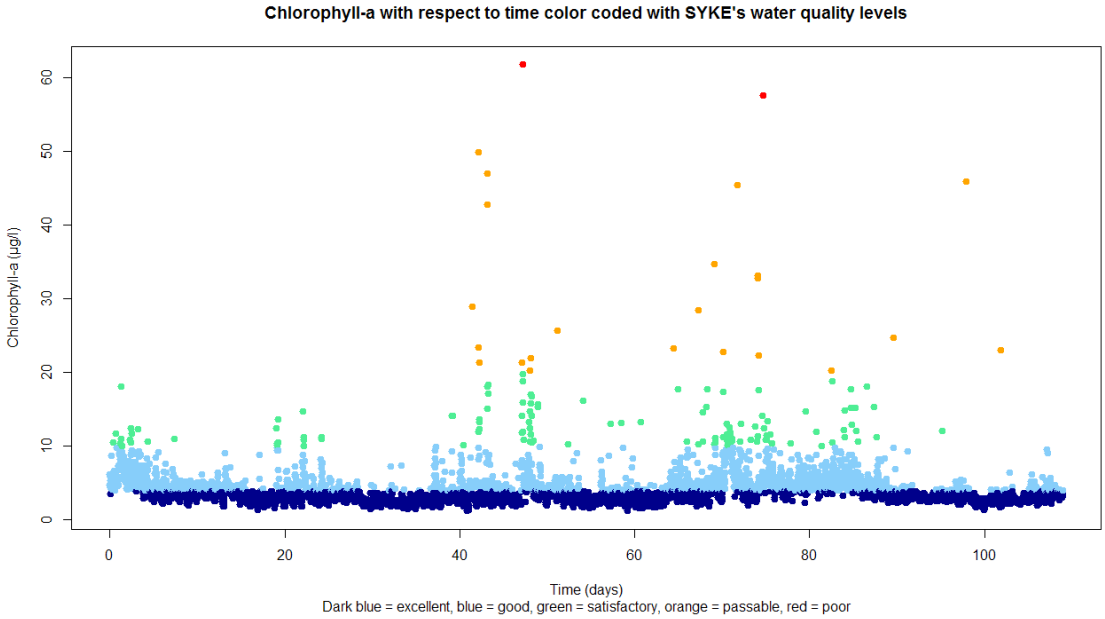

Figure 27. Colour-coded plot output from the assessment tool

The Figure 26 and Figure 27 show the chlorophyll level with respect to time. The Figure 27 lets us believe that most of the passable and poor points of the data might be outliers, as the variation of chlorophyll cannot happen this fast from a production point of view (Wetzel, 1983). This variation might be explained by the factors discussed in section 3.3 Factors that influence the levels of chlorophyll-a and its measurements.

5.3 Covariance matrix

Two covariance matrices have been estimated, one calculating the Pearson correlation coefficients and the second one using the Spearman methodology.

Table 3. Pearson Correlation Matrix

	temperature	conductivity	pH	turbidity	DO	a.chlorophyll
temperature	1	-0.808	0.78	-0.558	-0.718	0.007
conductivity	-0.808	1	-0.756	0.556	0.456	-0.131
pH	0.78	-0.756	1	-0.58	-0.21	0.089
turbidity	-0.558	0.556	-0.58	1	0.233	0.333
DO	-0.718	0.456	-0.21	0.233	1	0.044
a.chlorophyll	0.007	-0.131	0.089	0.333	0.044	1

Table 4. Spearman Correlation Matrix

	temperature	conductivity	pH	turbidity	DO	a.chlorophyll
temperature	1	-0.506	0.699	-0.486	-0.71	-0.011
conductivity	-0.506	1	-0.694	0.466	0.16	-0.103
pH	0.699	-0.694	1	-0.603	-0.136	-0.019
turbidity	-0.486	0.466	-0.603	1	0.261	0.37
DO	-0.71	0.16	-0.136	0.261	1	0.027
a.chlorophyll	-0.011	-0.103	-0.019	0.37	0.027	1

Absolute values of correlation coefficients below 0.3 show that the correlation between the variables is poor or non-existing (Varmuza & Filzmoser, 2009c). Surprisingly, with a correlation coefficient bellow 0.3 in both correlation matrices, the correlation between the chlorophyll and the temperature is practically non-existing. Limnology sources describe temperature as one of the key drivers of the phytoplankton growth, which is the main source of chlorophyll in the water column. Chlorophyll seems to correlate mainly with turbidity, which is understandable as chlorophyll is one of the components of turbidity, therefore if the concentration of chlorophyll increases, the turbidity increases. Nevertheless, an increase in turbidity does not necessarily mean that the chlorophyll level has increased.

The analysis of this table might help to understand some of the mechanisms in the lake, which could be useful to understand and explain variations in chlorophyll.

Dissolved oxygen and temperature are also clearly correlated. This can be explained by the effects of temperature on gas solubility. Lower temperatures lead to an increased solubility for gases in a liquid (University of California, Davis, 2016). Carbon dioxide is highly soluble in water. When the temperature decreases, the concentration of CO2 increases. The dissolved carbon dioxide contributes with the decrease in pH (acidification), which is why temperature and pH are correlated.

On the other hand, the solubility of solids is increased as the temperature of the liquid increases. As the conductivity is caused by dissolved salts, a lower temperature will result in the precipitation of the dissolved elements, and therefore in a decrease in conductivity.

The relationship between turbidity and temperature is not clear. Nevertheless, these might correlate because of seasonal changes. As temperature decreases, by the end of summer, stormy event frequency might increase. Runoff intake, water mixing and turbulence induced by the movement of the pier which the sensor is attached to increases with has consequences on the turbidity of the lake. Nevertheless, this is an intuitive approximation and requires further investigation to establish a formal relationship.

These results support the analysis done for the time series.

5.4 Principal Component Analysis

The PCA plot in Figure 28 shows clustering according to the water temperature. Additionally, in the left part of the plot (between -2 and -0.5 from PC1) we can see the time line go around the same area, which indicates that there might be a consistent behaviour of the lake in that time frame. The arm moving towards the right of the graph (from -0.5 onwards on PC1), probably shows the seasonal change from summer to winter.

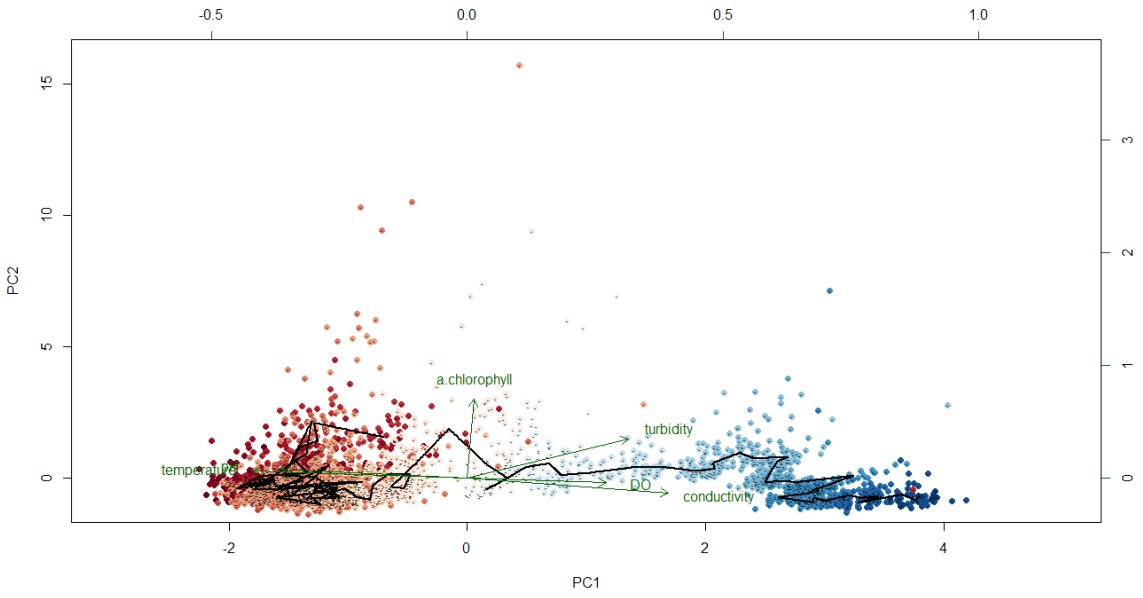

Figure 28. PC1 and PC2 (more detailed view available in the Appendix)

The eigenvectors indicate positive correlations between dissolved oxygen and conductivity; temperature and pH; and negative correlation between the first two with the last two. These correlations are in accordance to the correlation matrix in Table 3 and Table 4. Chlorophyll appears to be independent from pH, temperature, dissolved oxygen and conductivity.

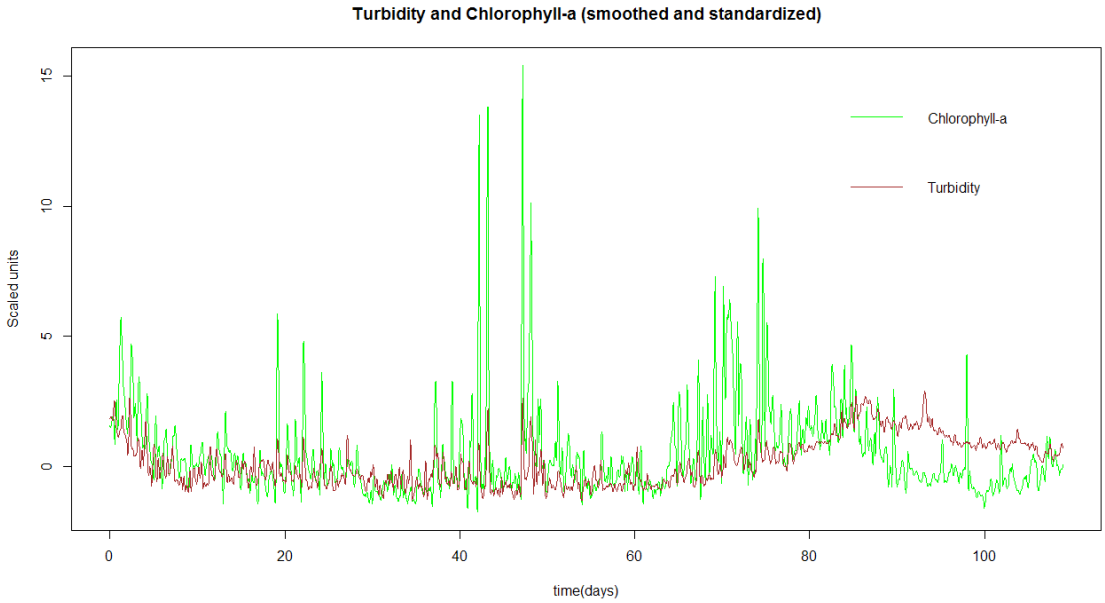

Figure 29. Chlorophyll-a and turbidity smoothed (more detailed view available in the Appendix)

The chlorophyll and turbidity complex relationship can also be seen on Figure 29. Both seem to correlate during the warmer period. As temperatures go down their behaviour is no longer explicitly correlated. This probably happens because of the autumn storms that participates to the turbidity levels but do not affect the production of chlorophyll.

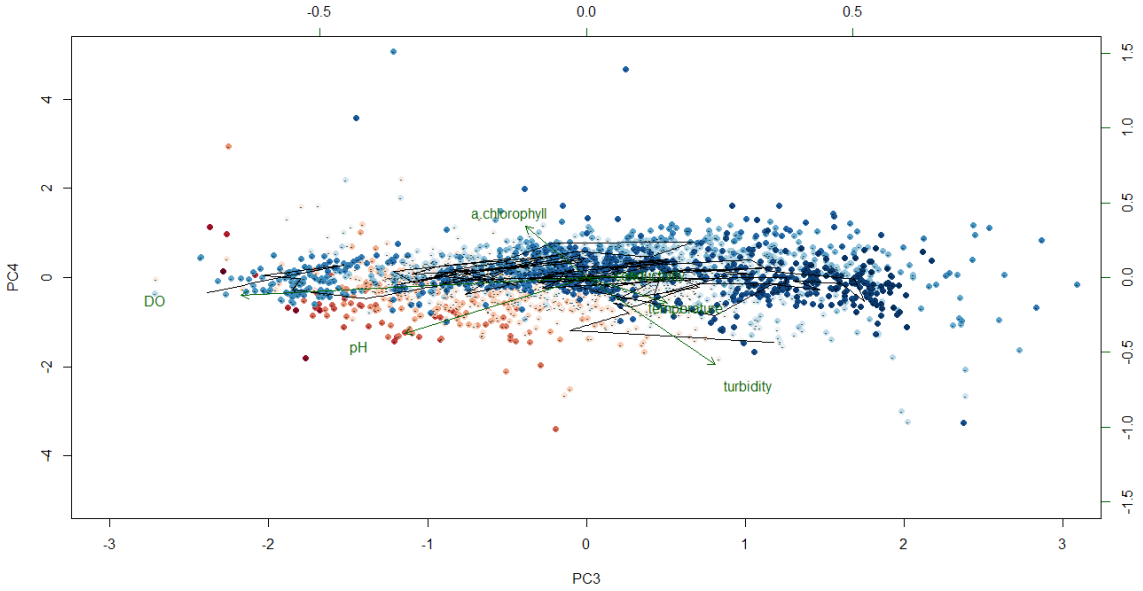

Figure 30. PC3 and PC4 (more detailed view available in the Appendix)

Additionally, looking at the Figure 30, we can see that turbidity and chlorophyll are negatively correlated. This might reflect the turbidity negative effect on chlorophyll production, as the increased turbidity limits the penetrability of light into the lake water. With less light, phytoplankton cannot reproduce as fast, therefore the level of chlorophyll decreases.

5.5 Multiple Linear Regression

Two methodologies have used to created using Multiple Linear Regression models: MLR1 and MLR2. The difference is in the formula used to standardize the data. MLR1 uses Equation 2, while MLR2 uses Equation 3, supposed to be more robust and better for data with outliers.

Both methods follow the process diagram presented in Figure 35 (available in the appendices). First the data goes through pre-processing (standardization). Then two sets are created. The first one is to be used as a verification set, containing 20% of the original data. The rest of the original data is assigned to the training, which is then used to train the model using a cross validation with forward variable selection strategy. The best model is then returned for a second stage of variable selection. The most insignificant terms are removed on a one by one basis until all the remaining terms are significant. Once this process is completed, the verification set is used to assess the quality of the model.

Three runs have been made from each methodology to guarantee the reproducibility of the results.

Both MRL1 and MRL2 seem to be giving models that yield an R^2 around 0.45. The Residual Standard Error (RSE) is always within the same magnitude order of the root mean square of the residuals (RMS). This would lead to believe that the model does not fully represent the variation of the data, but nevertheless it provides a decent fit to the data. This might be misleading as the verification set is composed by a sample of points from the original data. These points might follow the variations of the original data which makes this methodology potentially bias. The ideal would be to have random subsets of sequential data.

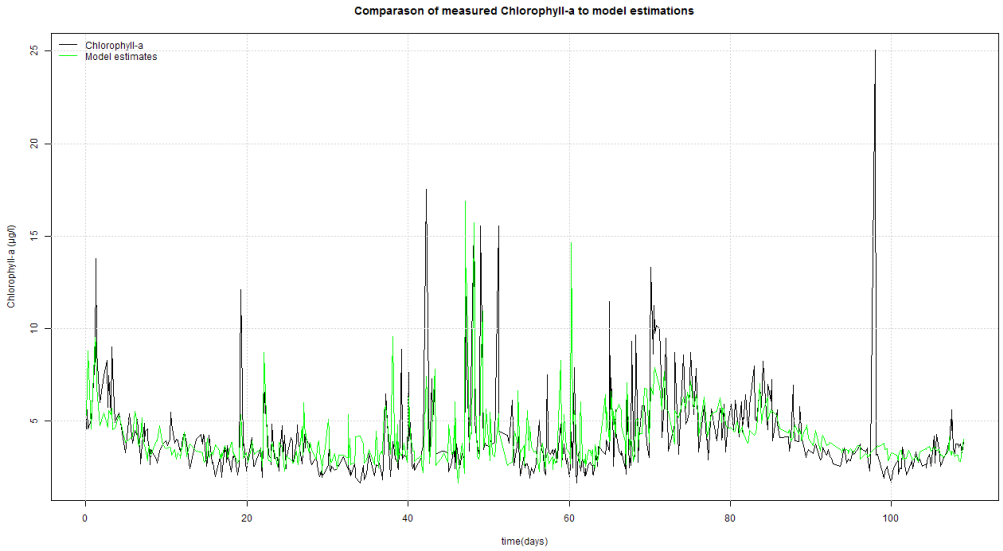

Figure 31. Modelled chlorophyll-a vs measured chlorophyll-a for MLR1 (more detailed view available in the Appendix)

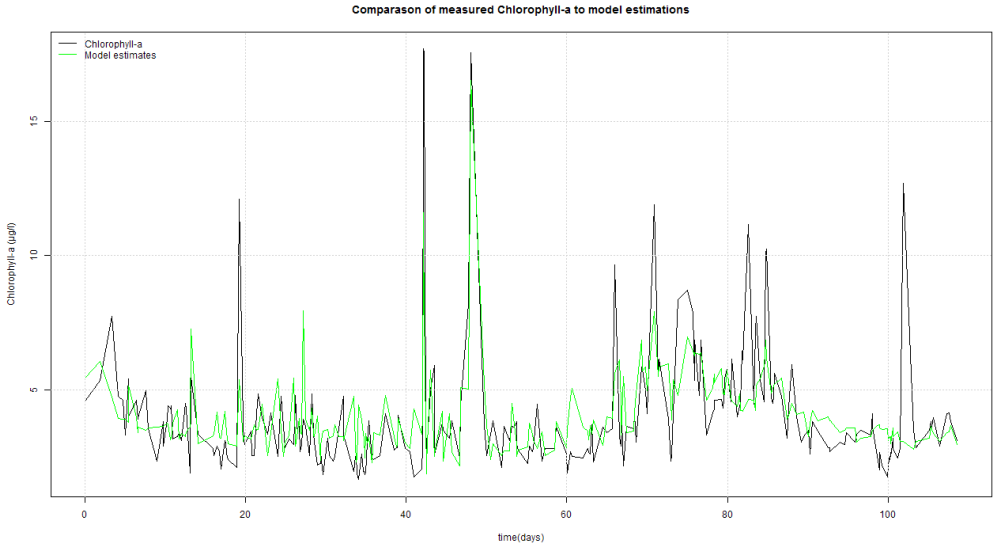

Figure 32. Modelled chlorophyll-a vs measured chlorophyll-a for MLR2 (more detailed view available in the Appendix)

Looking at the graphs (in Figure 31 and Figure 32) comparing the prediction (in green) to the measurements (in black) we can see differences in the regions where there are peaks. The prediction sometimes also produces peaks that do not exist in the measured data.

Nevertheless, the three runs seem to represent fairly to the overall tendency of the chlorophyll. These models could potentially be used to estimate daily averages (or lower resolution), but most likely not hourly levels of chlorophyll. The average of the predictions from the model (using the test set) are quite close to the average of the test set. The relative error between these two is always quite small (way below 5%) for both MLR1 and MLR2 in each run. This is potentially a good tool to estimate the average chlorophyll in a lake from the independent variables used during this thesis. Nevertheless, this needs to be confirmed with new data, that will be collected during the next summer (in 2017).

6 Discussion

The discussion chapter presents some of the issues and thoughts gathered during the making of this thesis. Additionally, it also gives some ideas for further development. Firstly, the current setup of the EXO2 sonde is discussed, followed by the different needs that could help to improve the significance and quality of the data generated by the probe. Finally, possibilities that are created by the deployment of such a solution are presented followed by thoughts on good practices in writing programming code.

6.1 Current setup of the EXO2 sonde

The setup is now collecting one data point every half an hour. There are no replicates made to reduce the effect of outliers. It could be wise to make at least one replicate for each measurement. Additionally, it can be valuable to make a measurement every 10 minute. This could allow to detect fast changes in chlorophyll, for instance due to water mixing. The extra data points can also improve the model generation and testing.

6.2 Needs for weather data

There was a very limited access to weather data. There is no weather station close to the monitoring area. The closest is at an approximated distance of 23.5km, which is over 10km, the best resolution available from the Finnish Meteorological Institute (Finnish Meteorological Institute, 2017).

As weather variables also influence on algal and cyanobacterial growth, it is very important to include such observations in the modelling process. Unfortunately, during the research we did not have time and resources to get access to weather data with enough significance and precision.

The ideal scenario would be to deploy at least one weather station that measures wind speed and direction, air temperature, solar irradiance and precipitation. Such observations would be valuable for a model that considers weather factors to predict or estimate levels of chlorophyll.

6.3 Needs for a working methodology with online sondes, field sampling and data treatment

To successfully complete a monitoring campaign, there is a need to develop a methodology that covers digital and manual sampling that considers database requirements and that enables to use the data in statistical studies. Field sampling should be taken near the sonde, and should match at least some of the parameters measured by the sensors. If possible, there should be at least one manual sample that measures all the parameters followed by the EXO2. In the case of this thesis, water samples were taken near the sensor but none of the samples covers all the parameters, making it impossible to incorporate them in the statistical analysis process.

Additionally, the format of the information should be standardized. This is particularly valid for choices such as date and time format, file format, separators and parameter units. This eases the data pre-processing and allows the re-use of the R scripts with new data.

6.4 Needs for nutrient monitoring

The original objective of the thesis was to model the nutrient levels depending on the same data that has been used for this thesis. Unfortunately, resources did not allow to establish a monitoring program for nutrient loads. The lack of data suitable for modelling purposes made this project impossible.

Because of the significant influence of nutrients in phytoplankton growth, it is essential to estimate the nutrient load present in the lake.

Additionally, an estimation of the nutrient load capacity can be made taking into consideration the environment in the lake's catchment area (agricultural and road activities).

6.5 Needs to estimate lake retention time

The lake retention time is an estimation of the time that water stays in the lake. For Lake Gennarbyträsket, the precise lake retention time is currently unknown. Nevertheless, based on a single observation (the only sample available was from 1989, it is assumed that this observation was done in the same time) using SYKE's VEMALA model (Huttunen, 2017), the lake retention time is estimated to be over 1000 days.

6.6 Needs for additional data points

Ideally, the data should be continuously collected for an extent of at least 2 to 3 years (including winter). This could be used to study the change of parameter behaviours over the years. Additionally, the new data can verify the model and eventually participate in the model training to improve model significance.
Measurements should be taken in series of replicates. This could help to assess possible outliers and make the data more robust.
There is a second dataset available with measurements from Lake Hiidenvesi with an EXO water station, but there are unfortunately no turbidity measurements. Therefore, this dataset cannot be used for this thesis.

6.7 Further development

The models could be further tested with the data that is going to be collected in 2017. This would allow to assess whether these models can be used to estimate daily and yearly averages of chlorophyll based on pH, turbidity, conductivity, temperature and dissolved oxygen.
Unfortunately, time and resources limited the scope of statistical tools that could be applied to the data to extract potential information from it. Experimenting with these tools could be done to have different viewpoints on the data. Additionally, it could make more sense to build the models using training and testing sets that are composed of randomly selected sequential series of data points from the original data. This might reduce the bias of the models as the actual methodology uses random points, which might implicitly inherit the overall variance of the original data set.
Finally, experiments could be designed to best assess the reliability of the EXO2 data and to optimize the modelling process of the data. Such experiments could be for instance monitoring the production of chlorophyll in a controlled environment.

6.8 Online service possibilities

The setup deployed in Lake Gennarbyträsket allows to collect data in a digital form. This data can be directly sent to a server to be stored in a database (which is already the case). This creates an opportunity for designing new services around the collected information. Such services could be, for instance, a web portal where citizens can follow the water quality in a monitoring site with an online sonde, such as the EXO2.
The online data could also be used for collaborative research. Different institute could deploy sensor arrays in water bodies and collect data that could be used for research purposes.

6.9 Programming good practices

Following programming good practices could potentially save a lot of time both in development and debugging. As the goal of programming in R is to perform mathematical calculations, a good programming practice would be to develop code following the functional programming paradigm. Unfortunately, this has not been done when working in this thesis, which lead to lengthy, repetitive and complicated source code. This paradigm has come to my knowledge by the end of the thesis, when time was no longer available and most of the programming was done. Functional programming could increase the readability of the code, reduce debugging time, increases code portability and improve code stability. It is therefore strongly advised to implement such practices for future source code.

7 Conclusion

The data from the EXO2 sonde provide us with information on the lake state. The modelling of chlorophyll has revealed more challenging than expected. No clear relation has been established between chlorophyll and pH, conductivity, turbidity, dissolved oxygen and temperature. The two multiple linear regressions modelling method outputs do not fully explain the hourly evolution of the data. Nevertheless, they might provide a good estimation of the average levels of chlorophyll, useful to determine the trophic state of the lake. This needs to be verified with new data that will unfortunately be available only from the summer onwards, too late for this thesis.

The statistical analysis of the data might reveal underlying mechanisms in the lake, such as the relationship with dissolved oxygen and temperature. The use of statistical methods to study the behaviour of the lake could lead to a deeper understanding of the relationships between the different elements of the lake. The resolution that the EXO2 offers to such an analysis is exceptional. It reveals patterns over time and allows an analysis that goes beyond the traditional sampling technique, bias because of the working time hours and limitations with the weather conditions and limited in number of observations over time.

8 Acknowledgements

This thesis has been a rich learning experience. I had the opportunity to freely experiment with the data while applying my learnings from the Environmetrics course. This first contact with modelling and statistical analysis has further developed my understanding of the possibilities and limitations from such an approach. It has also given me the inspiration to further continue my studies into a Masters of Sciences in Oceanography at the University of Southampton (United Kingdom).

I would like to first thank Veli-Matti Taavitsainen, my thesis supervisor and main lecturer in mathematics, for transmitting such a valuable knowledge and for his advices, guidance and help during the making of this thesis. It has been a very valuable experience for me.

I would also like to thank Antti Tohka, Jenni Merjankari and Kaj Lindedahl for making this thesis project possible within the provided timeframe and circumstances. Thanks also to Minna Paananen-Porkka, our language advisor, for all the insights on proper manners for writing a thesis and scientific papers. Additionally, I would like to thank Länsi-Uudenmaan Vesi ja Ympäristö ry and Village Waters for making it possible for me to work with an EXO 2 sonde and for the data that has been shared, without this input this thesis could not possibly exist. I would like to thank Luode Consulting Oy for their technical support with respect to the EXO 2 sonde and its probes. I would also like to thank the Finnish meteorological institute for the weather data they have gracefully provided.

I would also like to take this occasion to demonstrate my gratitude to Anu Suonpää, who has patiently and generously supported me in every possible way during the entire period of my studies and in particular with this thesis. Without her this thesis would not have been possible.

Julien Mineraud, Marko Kallio, Johanna Markkanen and Isadora Bicalho are also to be thanked for their support and guidance during the making of this thesis. Thanks also to everybody involved in a way or another with this thesis.

Additionally, I would like to thank the R project participants and community for their generous work and dedication. This thesis would not be possible without the existence of the R project and all the packages that have been used. The R project is an incredible tool that democratizes computational approaches of statistics.

Finally, I would like to thank my parents, Arlette and Humberto Espinola, for their unconditional support during all these years.

I would like to dedicate this thesis to Anu, my spouse, and my son, who will be born later during this summer. You have provided me with the inspiration and energy to conclude this thesis.

References

Albers, M. J., 2017 (to be published). *Introduction to Quantitative Data Analysis in the Behavioral and Social Sciences.* 1st ed. Hoboken: John Wiley & Sons.

Berthouex, P. M. & Brown, L. C., 2002. *Statistics for Environmental Engineers.* 2nd ed. Boca Raton: Lewis Publishers.

Comprehensive R Archive Network, 2017. *R: Scaling and Centering of Matrix-like Objects.* [Online]
Available at: https://stat.ethz.ch/R-manual/R-devel/library/base/html/scale.html
[Accessed 9 April 2017].

Finnish Meteorological Institute, 2017. *Climate Statistics.* [Online]
Available at: http://en.ilmatieteenlaitos.fi/climate-statistics
[Accessed 3 March 2017].

HELCOM, 2017. *Monitoring requirements - HELCOM.* [Online]
Available at: http://www.helcom.fi/baltic-sea-trends/indicators/chlorophyll-a/monitoring-requirements/
[Accessed 1 April 2017].

Huttunen, M., 2017. *VEMALA – a water quality and nutrient load model system for Finnish watersheds.* [Online]
Available at: http://www.syke.fi/en-US/Research__Development/Sustainable_management_of_the_Baltic_Sea_and_freshwater_resources/Models_and_tools/Models_for_river_basin_management_planning/A_water_quality_and_nutrient_load_model_system_for_Finnish_watersheds_VEMALA
[Accessed 9 April 2017].

Internet Engineering Task Force, 2005. *RCF 4180.* [Online]
Available at: https://tools.ietf.org/html/rfc4180
[Accessed 3 March 2017].

Länsi-Uudenmaan Vesi ja Ympäristö ry, 2013. *Toiminta-alue.* [Online]
Available at: http://www.luvy.fi/easydata/customers/luvy/files/yhdistys/toiminta-alue/toimialueen_kartta_suomi.jpg
[Haettu 19 February 2017].

Länsi-Uudenmaan Vesi ja Ympäristö ry, 2016. *The Association for Water and Environment of Western Uusimaa.* [Online]
Available at: http://www.luvy.fi/en
[Accessed 19 February 2017].

Mac Berthouex, P. & Brown, L. C., 2002. *Statistics for Environmental Engineers.* 2nd ed. Boca Raton: Lewis Publishers.

Miller, C. B., 2004. *Biological oceanography.* 1st ed. Oxford, UK: Blackwell Science Ltd..

Mitikka, S., 2015. *Ympäristö.fi.* [Online]
Available at: http://www.ymparisto.fi/download/noname/%7BC1C37484-04C6-43CE-95BF-E30D2BB78A29%7D/78231
[Accessed 1 April 2017].

Mortenson, M. E., 1985. *Geometric Modelling.* 1st ed. New York: John Wiley & Sons.
Oravainen, R., 1999. *Vesistötulosten Tulkinta - opasvihkonen.* [Online]
Available at: http://kvvy.fi/wp-content/uploads/2015/10/opasvihkonen.pdf
[Accessed 1 April 2017].

R Reference, 2001. *Date-Time Classes.* [Online]
Available at: https://stat.ethz.ch/R-manual/R-devel/library/base/html/DateTimeClasses.html
[Accessed 27 March 2017].

SAS Institute Inc., 2016. *Statistical Analysis - What is it?.* [Online]
Available at: https://www.sas.com/en_us/insights/analytics/statistical-analysis.html
[Accessed 3 April 2017].

Sawitzki, 2009a. *Computational Statistics, An Introduction to R.* 1st ed. Boca Raton: CRC Press.

Sawitzki, G., 2009b. *Computational Statistics, An introduction to R.* 1st ed. Boca Raton: CRC Press.

Sokal, R. R. & Rohlf, F. J., 1995. *Biometry.* 3rd ed. New York: W. H. Freeman and Company.

Suonpää, A., 2015. *Bongaa Kalaa.* [Online]
Available at: http://www.bongaakala.fi/kalat-ja-jarvikunnostus
[Accessed 3 April 2017].

Suonpää, A., 2017. *Consultation on the water quality measuring and data interpretation protocols and practices in LUVY.* [Interview] (1 4 2017).

Taavitsainen, V.-M., 2017. *Cross validation and variable selection strategy for modelling.* [Interview] (7 April 2017).

University of California, Davis, 2016. *Effects of Temperature and Pressure on Solubility.* [Online]
Available at: https://chem.libretexts.org/Textbook_Maps/General_Chemistry_Textbook_Maps/Map%3A_Chemistry_(Averill_and_Eldredge)/13%3A_Solutions/13.4%3A_Effects_of_Temperature_and_Pressure_on_Solubility
[Accessed 15 April 2017].

Varmuza, K. & Filzmoser, P., 2009a. Callibration. In: C. Press, ed. *Table Of Content Section.* Boca Raton: CRC Press, pp. 103-194.

Varmuza, K. & Filzmoser, P., 2009b. Centering and Scaling. In: C. Press, ed. *Introduction to Multivariate Statistical Analysis in Chemometrics.* 1st ed. Baton Rouge: CRC Press, pp. 35-36.

Varmuza, K. & Filzmoser, P., 2009c. *Introduction to Multivariate Statistical Analysis in Chemometrics.* 1st ed. Boca Raton: CRC Press.

Village Waters, 2017a. *About Village Waters.* [Online]
Available at: https://www.villagewaters.eu/About_VillageWaters_757
[Accessed 21 February 2017].

Village Waters, 2017b. *Gennarby, Finland.* [Online]
Available at: https://villagewaters.eu/Gennarby_Finland_777
[Accessed 21 February 2017].

Wetzel, R. G., 1983. *Limnology.* 2nd ed. New York: CBS College Publishing.
YSI Inc. / WTW GmbH / Xylem Inc., 2017. *EXO Sensors.* [Online]
Available at: http://www.exowater.com/sensors
[Accessed 25 February 217].

YSI Inc./Xylem Inc., 2017. *YSI EXO2 water quality sonde is a new YSI multiparameter instrument that collects data with six user-replaceable sensors. | ysi.com.* [Online]
Available at: https://www.ysi.com/EXO2
[Accessed 25 February 2017].

Field data from LUVY for the Gennarbyträsket lake

Table 5. Field measurements.

Date	Place	Name	Time	Temp	Appearence	Smell	Flow	Tot depth
Month/day/year			hh:mm (UTC+3)	°C			m3/s	m
4/19/2016	1	Gennarbyträsket tulo	11:15	5	YEB	H	0.003	
4/19/2016	2	Gennarbyträsket NE	9:25	6.8	CB	L		
4/19/2016	3	Gennarbyträsket N	9:40	6.7	CB	L		
4/19/2016	4	Gennarbyträsket NW	10:00	6.7	CB	H		
4/19/2016	5	Gennarbyträsket lasku	10:40	6.7	CB	H	0.0075	
4/19/2016	6	Gennarbyträsket syvänne	10:20	5.4	CB	H		10.6
7/21/2016	1	Gennarbyträsket tulo	9:48	13.4	WF	H	0.001	
7/21/2016	2	Gennarbyträsket NE	9:58	20.5	CB	H		
7/21/2016	3	Gennarbyträsket N	9:05	20.4	CB	H		
7/21/2016	4	Gennarbyträsket NW	8:47	20.6	CB	H		
7/21/2016	4	Gennarbyträsket NW		P				
7/21/2016	5	Gennarbyträsket lasku	10:08	20.2	CB	H	0.003	
7/21/2016	6	Gennarbyträsket syvänne	9:26	7.1	CB	LRV		10
8/18/2016	1	Gennarbyträsket tulo	11:30	13.3	CB	H	0.0005	
8/18/2016	2	Gennarbyträsket NE	11:45	18	CB	H		
8/18/2016	3	Gennarbyträsket N	11:55	18.1	CB	H		
8/18/2016	4	Gennarbyträsket NW	10:55	17.7	CB	H		
8/18/2016	5	Gennarbyträsket lasku	12:32	14.5	CB	H	0.0002	
8/18/2016	6	Gennarbyträsket syvänne	11:15	6.8	CB	SRV		11

Table 6. Laboratory analysis results.

Date	Place	Depth	*solids.GFC	*O2	*pH	conductivity	*BOD7	Tot.N	*NH4-N	Tot.P	*a-chlorofy	*Ecoli	Enterokok.	*koliler
Month/day/year		m	mg/l	mg/l		mS/m	mg/l	µg/l	µg/l	µg/l	µg/l	pmy/100 ml	pmy/100 ml	pmy/100 ml
4/19/2016	1	0.1	6.1			5.8	<1,5	1100	15	40		0	1	43
4/19/2016	2	0.5	1.6				<1,5	600	15	15		0	0	3
4/19/2016	3	0.5	1.5			5.9	<1,5	580	15	14		0	1	1
4/19/2016	4	0.5	1.7			5.9	<1,5	590	15	15		0	0	2
4/19/2016	5	0.1	1.5			5.9	<1,5	620	16	15		3	2	8
4/19/2016	6	bottom-1m		8.9	7.1			620	27	16		1	0	1
7/21/2016	1	0.1	4.7			7.2	<1,5	1300	28	28		37	88	980
7/21/2016	2	0.5	1.4				<1,5	350	5.8	11		0	1	91
7/21/2016	3	0.5	1.5			6.1	<1,5	350	5.6	11		3	4	120
7/21/2016	4	0.5	2			6.1	<1,5	360	6	13		1	4	150
7/21/2016	4	0-1m kokooma									5.7			
7/21/2016	5	0.1	1.4			6.2	<1,5	380	21	25		15	12	1100
7/21/2016	6	pohja-1m		0.7	6.7			970	540	48		2	0	66
8/18/2016	1	0.1	5.3			9	<1,5	470	43	40		4	34	2400
8/18/2016	2	0.5	1.4				<1,5	380	5.5	12		0	3	1000
8/18/2016	3	0.5	1.7			6.4	<1,5	390	5.7	15		2	3	190
8/18/2016	4	0.5	<1			6.2	<1,5	330	5.9	11		3	4	180
8/18/2016	5	0.1	2			10	1.6	660	87	230		490	150	>2400
8/18/2016	6	bottom-1m		0.2	6.9			1400	990	110		2	0	79

Maps

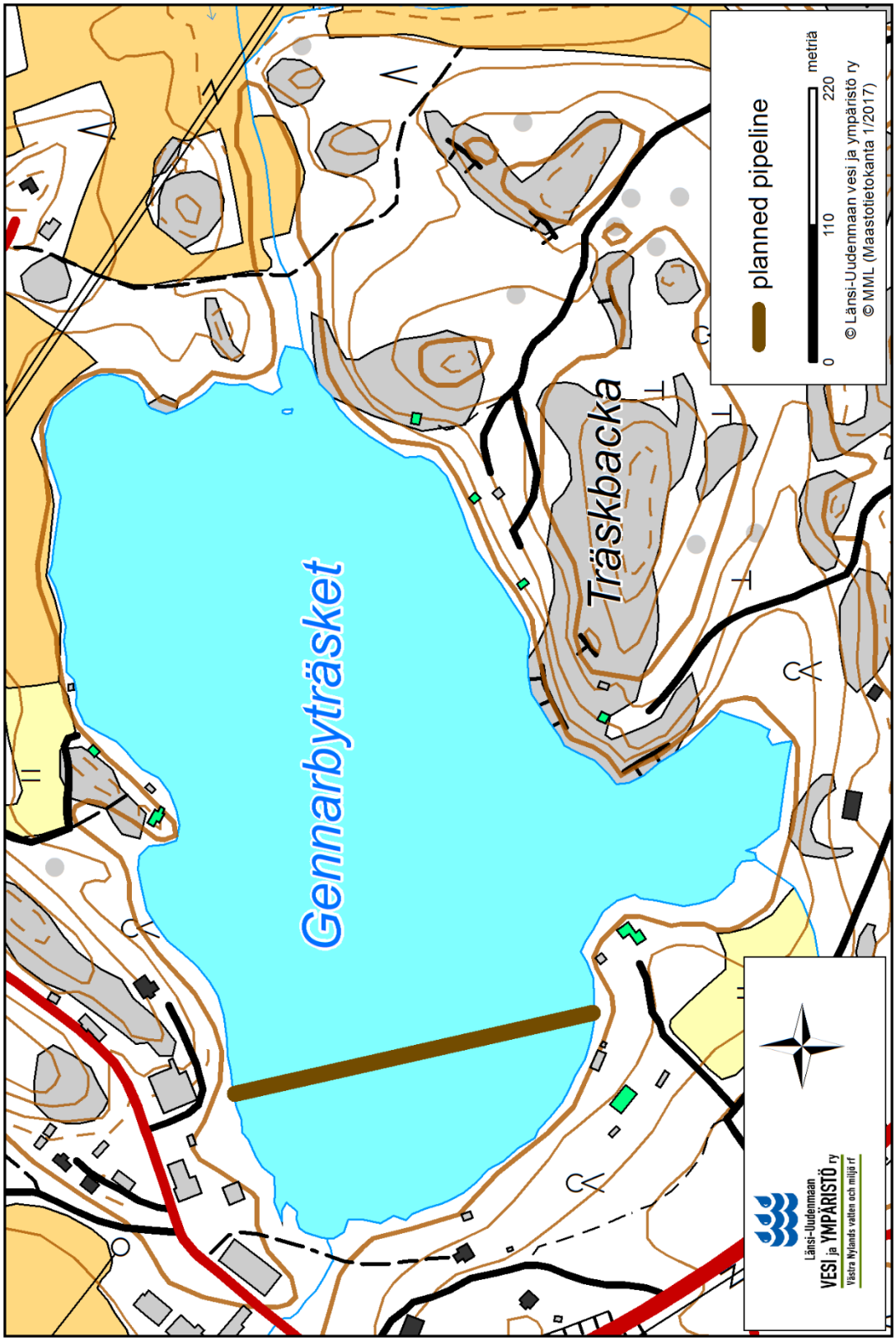

Figure 33. Original routing of the sewer pipe through Lake Gennarbyträsket (image credit: LUVY

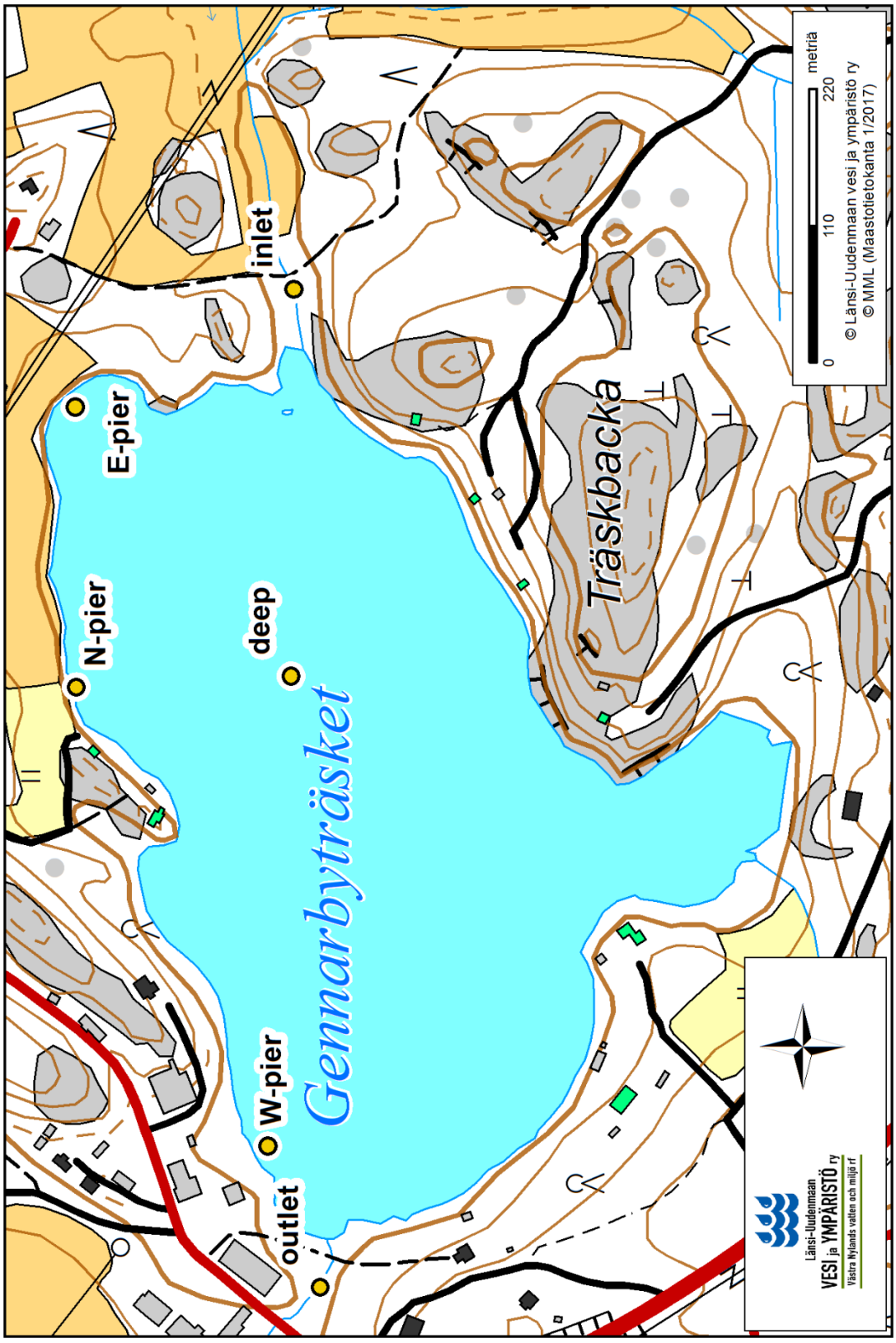

Figure 34. Sampling locations

Raw data

Table 7. head of the EXO2 raw data

Date and time	Lämpötila [°C]	Johtokyky [µS/cm]	pH [ ]	Sameus [FTU]	Happi [mg/l]	A-klorofylli [µg/l]	Sinilevien osuus [µg/l]	Akku [V]
21.07.2016 12:38	20,97	59,5	7,54	0,7	9,10	5,0	0,1	12,95
21.07.2016 13:00					9,05			12,97
21.07.2016 13:30	21,04	59,8	7,53	0,7	9,05	5,4	0,1	12,97
21.07.2016 14:00	21,08	59,5	7,49	0,7	9,00	6,2	0,1	12,99
21.07.2016 14:30	21,09	59,5	7,31	0,7	8,76	4,8	0,1	12,99
21.07.2016 15:00	21,12	59,5	7,28	0,7	8,70	4,4	0,1	13,01
21.07.2016 15:30	21,13	59,5	7,29	0,9	8,72	3,5	0,1	12,99
21.07.2016 16:00	21,19	59,5	7,30	0,6	8,79	5,8	0,1	13,01
21.07.2016 16:30	21,18	59,5	7,25	0,8	8,65	4,7	0,1	13,01
21.07.2016 17:00	21,21	59,5	7,26	0,7	8,70	5,4	0,1	13,01
21.07.2016 17:30	21,26	59,5	7,35	0,8	8,91	8,7	0,2	13,01
21.07.2016 18:00	21,34	59,5	7,37	0,7	8,91	5,7	0,1	13,01
21.07.2016 18:30	21,41	59,5	7,33	0,8	8,88	6,1	0,1	12,99
21.07.2016 19:00	21,41	59,5	7,62	0,8	9,20	5,1	0,1	12,98
21.07.2016 19:30	21,40	59,5	7,81	0,6	9,33	5,1	0,1	12,98
21.07.2016 20:00	21,35	59,5	7,73	0,6	9,27	4,1	0,1	12,98
21.07.2016 20:30	21,31	59,5	7,63	0,7	9,14	6,3	0,1	12,98
21.07.2016 21:00	21,28	59,8	7,56	0,7	9,09	5,7	0,1	12,97
21.07.2016 21:30	21,26	59,5	7,46	0,6	8,94	4,8	0,1	12,95
21.07.2016 22:00	21,21	59,8	7,41	0,8	8,88	6,1	0,1	12,94
21.07.2016 22:30	21,16	59,8	7,41	0,7	8,88	10,5	0,2	12,93

Table 8. selection from the rain accumulation raw data

Row #	Lyhyt_nimi	Year	Month	Day	Hour	Minute	Rain accumulation
1	HANKO TVÄRMINNE	2016	7	1	0	0	-1
2	HANKO TVÄRMINNE	2016	7	1	1	0	-1
3	HANKO TVÄRMINNE	2016	7	1	2	0	-1
4	HANKO TVÄRMINNE	2016	7	1	3	0	-1
12	HANKO TVÄRMINNE	2016	7	1	11	0	-1
13	HANKO TVÄRMINNE	2016	7	1	12	0	-1
14	HANKO TVÄRMINNE	2016	7	1	13	0	-1
15	HANKO TVÄRMINNE	2016	7	1	14	0	-1
16	HANKO TVÄRMINNE	2016	7	1	15	0	-1
17	HANKO TVÄRMINNE	2016	7	1	16	0	-1
18	HANKO TVÄRMINNE	2016	7	1	17	0	-1
19	HANKO TVÄRMINNE	2016	7	1	18	0	-1
20	HANKO TVÄRMINNE	2016	7	1	19	0	-1
21	HANKO TVÄRMINNE	2016	7	1	20	0	-1
22	HANKO TVÄRMINNE	2016	7	1	21	0	-1
23	HANKO TVÄRMINNE	2016	7	1	22	0	-1
24	HANKO TVÄRMINNE	2016	7	1	23	0	-1
25	HANKO TVÄRMINNE	2016	7	2	0	0	-1
26	HANKO TVÄRMINNE	2016	7	2	1	0	-1
27	HANKO TVÄRMINNE	2016	7	2	2	0	-1
28	HANKO TVÄRMINNE	2016	7	2	3	0	-1
40	HANKO TVÄRMINNE	2016	7	2	15	0	-1
41	HANKO TVÄRMINNE	2016	7	2	16	0	-1
42	HANKO TVÄRMINNE	2016	7	2	17	0	-1
43	HANKO TVÄRMINNE	2016	7	2	18	0	-1

Table 9. Selection of weather raw data.

Row #	Lpnn	Station Name	Year	Month	Day	Hour	Minute	Temperature [C]	Dewpoint [C]	Relative Humidity	Wind direction (degrees)	Average wind speed (last 10 min) [m/s]	Maximum wind gust (last 10 min) [m/s]
1	120	KEMIÖNSAARI KEMIÖ	2016	7	1	0	0	16	15	94	173	3.3	4.4
2	120	KEMIÖNSAARI KEMIÖ	2016	7	1	0	10	15.9	14.9	94	169	3.2	4.6
3	120	KEMIÖNSAARI KEMIÖ	2016	7	1	0	20	15.9	14.9	94	163	3.4	4.5
4	120	KEMIÖNSAARI KEMIÖ	2016	7	1	0	30	15.8	14.9	94	166	3.5	4.8
5	120	KEMIÖNSAARI KEMIÖ	2016	7	1	0	40	15.6	14.7	94	170	3	4.5
6	120	KEMIÖNSAARI KEMIÖ	2016	7	1	0	50	15.3	14.5	95	175	2.9	3.6
7	120	KEMIÖNSAARI KEMIÖ	2016	7	1	1	0	15.1	14.4	96	174	2.6	4
8	120	KEMIÖNSAARI KEMIÖ	2016	7	1	1	10	15.1	14.4	96	177	2.6	3.4
9	120	KEMIÖNSAARI KEMIÖ	2016	7	1	1	20	14.8	14.1	96	187	2.5	3.5
10	120	KEMIÖNSAARI KEMIÖ	2016	7	1	1	30	14.6	14	96	180	2.2	2.8
11	120	KEMIÖNSAARI KEMIÖ	2016	7	1	1	40	14.8	14.3	97	173	2.1	3
12	120	KEMIÖNSAARI KEMIÖ	2016	7	1	1	50	15	14.5	96	175	2.5	3.4
13	120	KEMIÖNSAARI KEMIÖ	2016	7	1	2	0	15.1	14.5	96	179	2.8	3.5
14	120	KEMIÖNSAARI KEMIÖ	2016	7	1	2	10	15.3	14.7	96	181	2.8	3.8
19096	202	HANKO TVÄRMINNE	2016	7	3	16	40	17.5	10.3	63			
19097	202	HANKO TVÄRMINNE	2016	7	3	16	50	17.4	10.5	64			
19098	202	HANKO TVÄRMINNE	2016	7	3	17	0	17.5	10.7	64			
19099	202	HANKO TVÄRMINNE	2016	7	3	17	10	17.1	9.4	60			
19100	202	HANKO TVÄRMINNE	2016	7	3	17	20	16.9	9.1	60			
37763	205	SALO KÄRKKÄ	2016	7	3	9	30	17.5	13.2	76			
37764	205	SALO KÄRKKÄ	2016	7	3	9	40	17.5	12.3	72			
37765	205	SALO KÄRKKÄ	2016	7	3	9	50	17.5	12.4	72			
37766	205	SALO KÄRKKÄ	2016	7	3	10	0	17.3	12.5	73			
37767	205	SALO KÄRKKÄ	2016	7	3	10	10	18.2	13.3	73			
37768	205	SALO KÄRKKÄ	2016	7	3	10	20	18.1	13.2	73			
37769	205	SALO KÄRKKÄ	2016	7	3	10	30	17.6	12.6	72			
37770	205	SALO KÄRKKÄ	2016	7	3	10	40	17.9	13.5	75			
37771	205	SALO KÄRKKÄ	2016	7	3	10	50	18.1	13	72			
37772	205	SALO KÄRKKÄ	2016	7	3	11	0	18.1	12.3	69			
74837	307	LOHJA PORLA	2016	11	7	23	0	-6.6	-8.4	87			
74838	307	LOHJA PORLA	2016	11	7	23	10	-6.7	-8.5	87			
74839	307	LOHJA PORLA	2016	11	7	23	20	-6.7	-8.5	87			
74840	307	LOHJA PORLA	2016	11	7	23	30	-6.7	-8.4	88			
74841	307	LOHJA PORLA	2016	11	7	23	40	-6.8	-8.5	88			
74842	307	LOHJA PORLA	2016	11	7	23	50	-6.8	-8.7	87			
74843	307	LOHJA PORLA	2016	11	8	0	0	-6.9	-8.6	88			

The modelling process

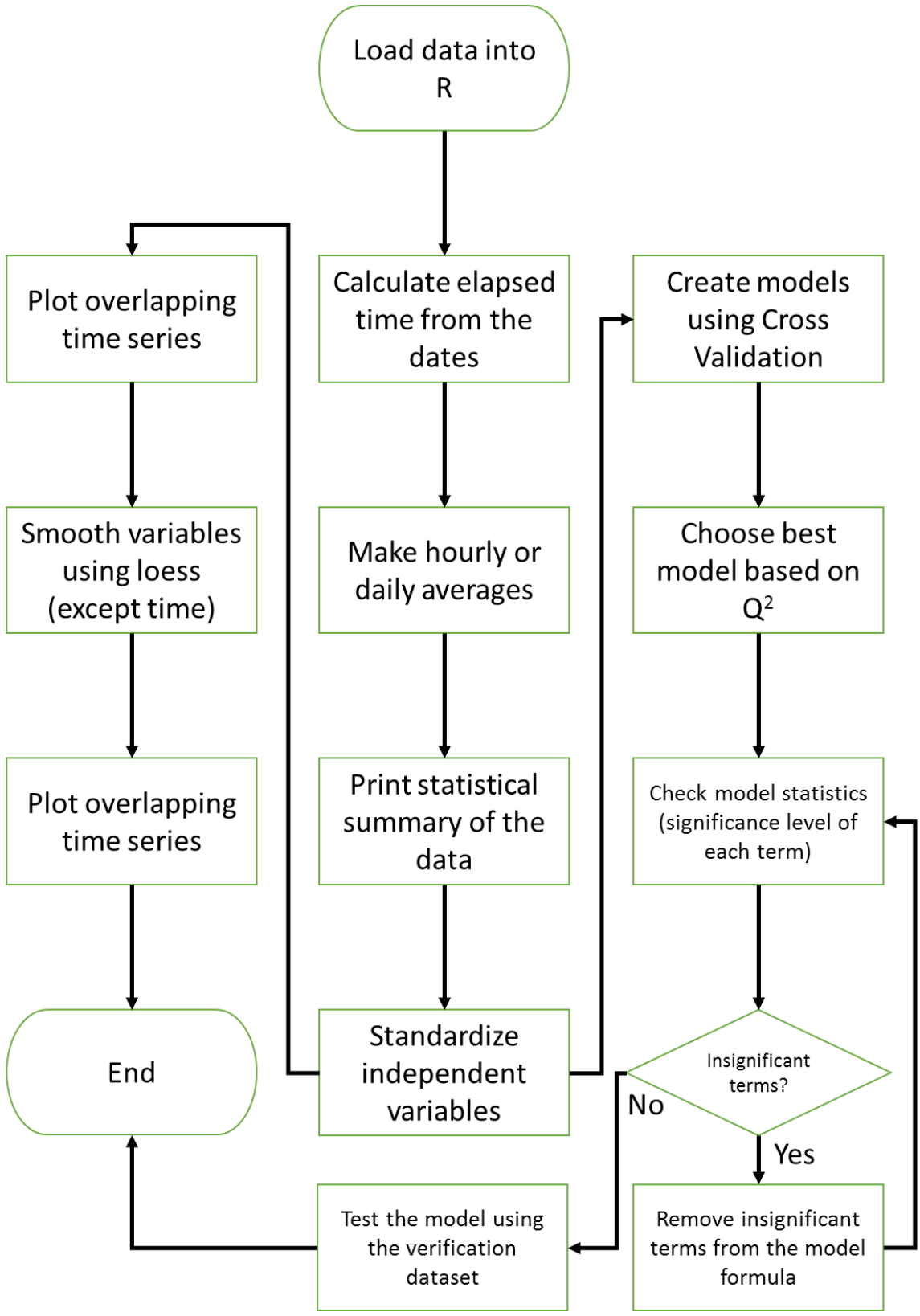

Figure 35. Process diagram for the data pre-treatment, smoothing and model construction.

R code for examples

```
# Graph example for paragraph 4.2 - Time series.
temp <- rnorm(60,7.6,0.23)
temp[34] <- 13.8
temp[13] <- 1.6
plot(temp, type = 'l', xlab='time (m)', ylab='pH', main='lake pH examp
le')

# Example for paragraph 22 - Effects of standardization on graphs.
temp <- rnorm(60,10,5)
A <- sin(temp)
B <- rnorm(60,1,0.1)
B <- (A * B)*1e6
plot(B, xlab = 'time (min)', ylab='Something', main='B series', type='
l', col='black')
plot(A, xlab = 'time (min)', ylab='Something else', main='A series', t
ype='l', col='red')
# A and B are similar but have different scales
par(mar = c(5,5,2,5))
plot(B, type= 'l', xlab = "time (min)", ylab="Something (B)", ylim=c(m
in(B),max(B)), main='A and B are not standardized')
par(new = T)
plot(A, type= 'l', axes=F, ylim=c(min(B),max(B)),col='red', xlab=NA, y
lab=NA)
axis(side = 4, ylab="bar", col = 'red')
mtext(side = 4, line = 3, 'Something else (A)', col='red')
# the series A is looks like a straight line
A <- scale(A,T,T)
B <- scale (B,T,T)
par(mar = c(5,5,2,5))
plot(B, type= 'l', xlab = "time (min)", ylab="Scaled units", ylim=c(mi
n(A,B),max(A,B)), main='A and B are standardized using scale')
lines(A, type= 'l', col= 'red')

#Loess example
cor.test(A,B)

x <- c(0:100)
noise <- rnorm(n = 101, mean = 0, sd = 50)
y <- sin(x*pi/100)
y <- y+noise

plot(x,y,type='l', main='Loess smoothing example')

DF <- data.frame(x=x, y=y)
model <- loess(y~x, DF)
lines(x, predict(model),col='red', lwd=2)

model <- loess(y~x, DF, span=.1)
lines(x, predict(model),col='blue', lwd=2)
```

Time series (pH)

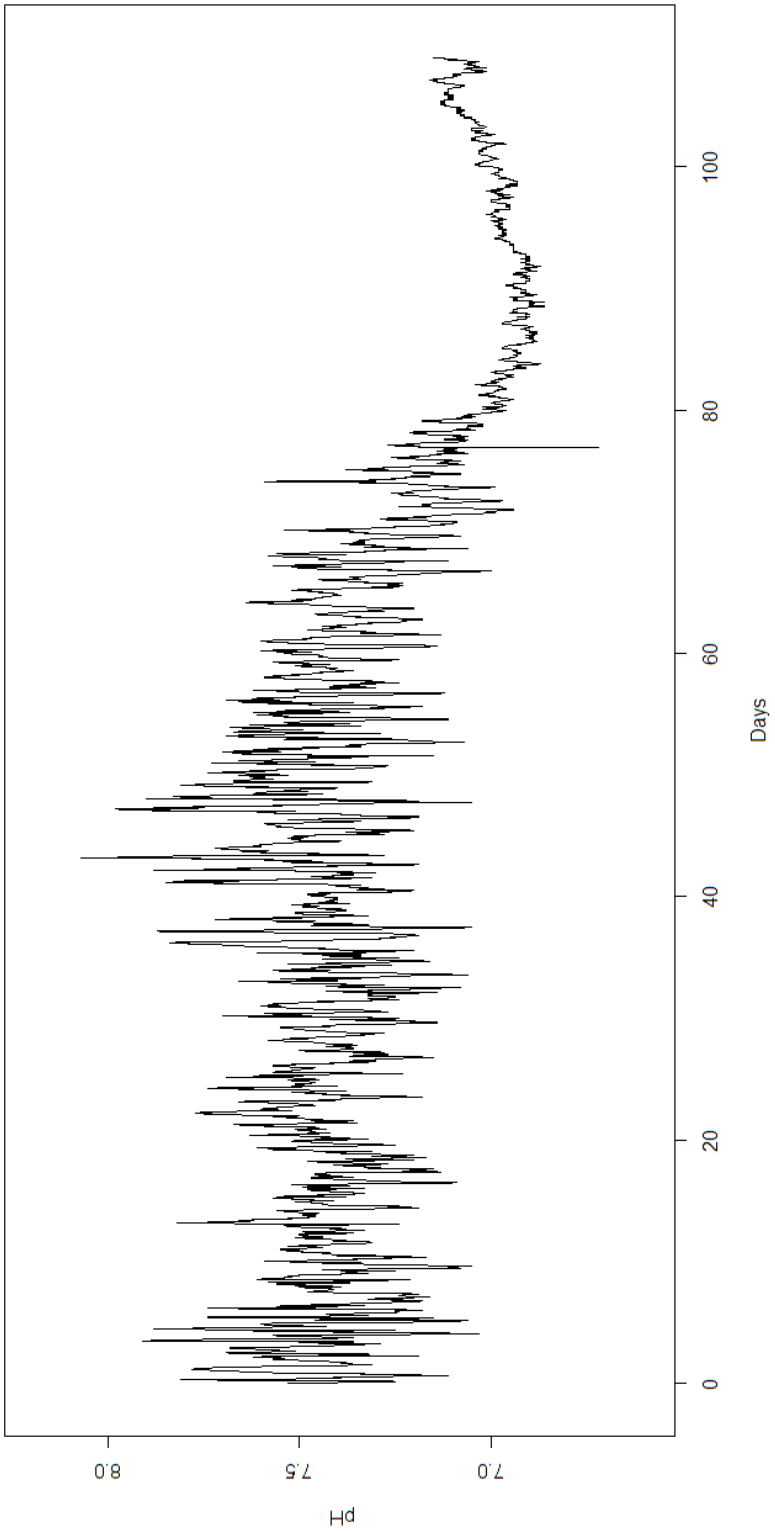

Figure 36. pH time series

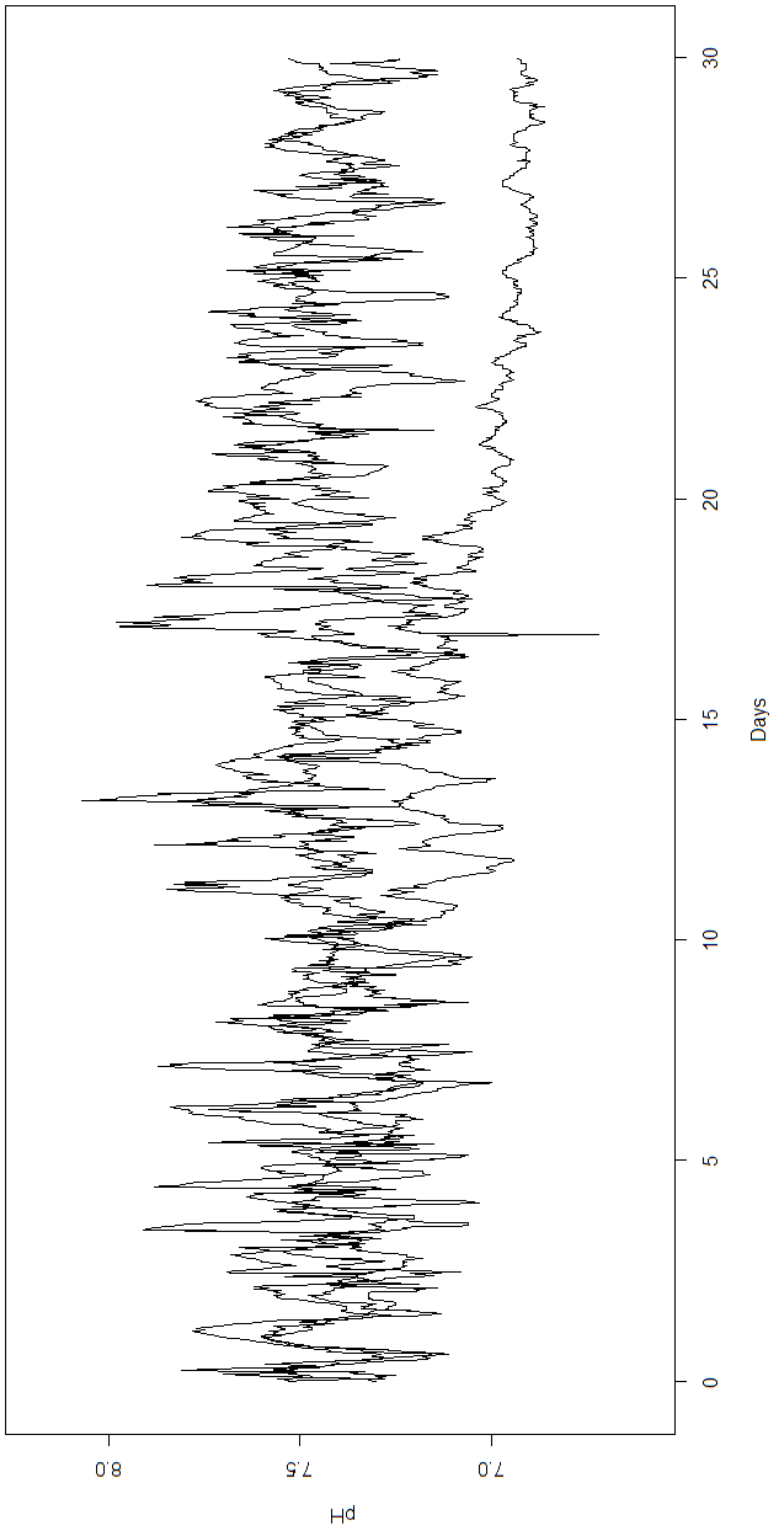

Figure 37. pH monthly (30 days) overlapping time series

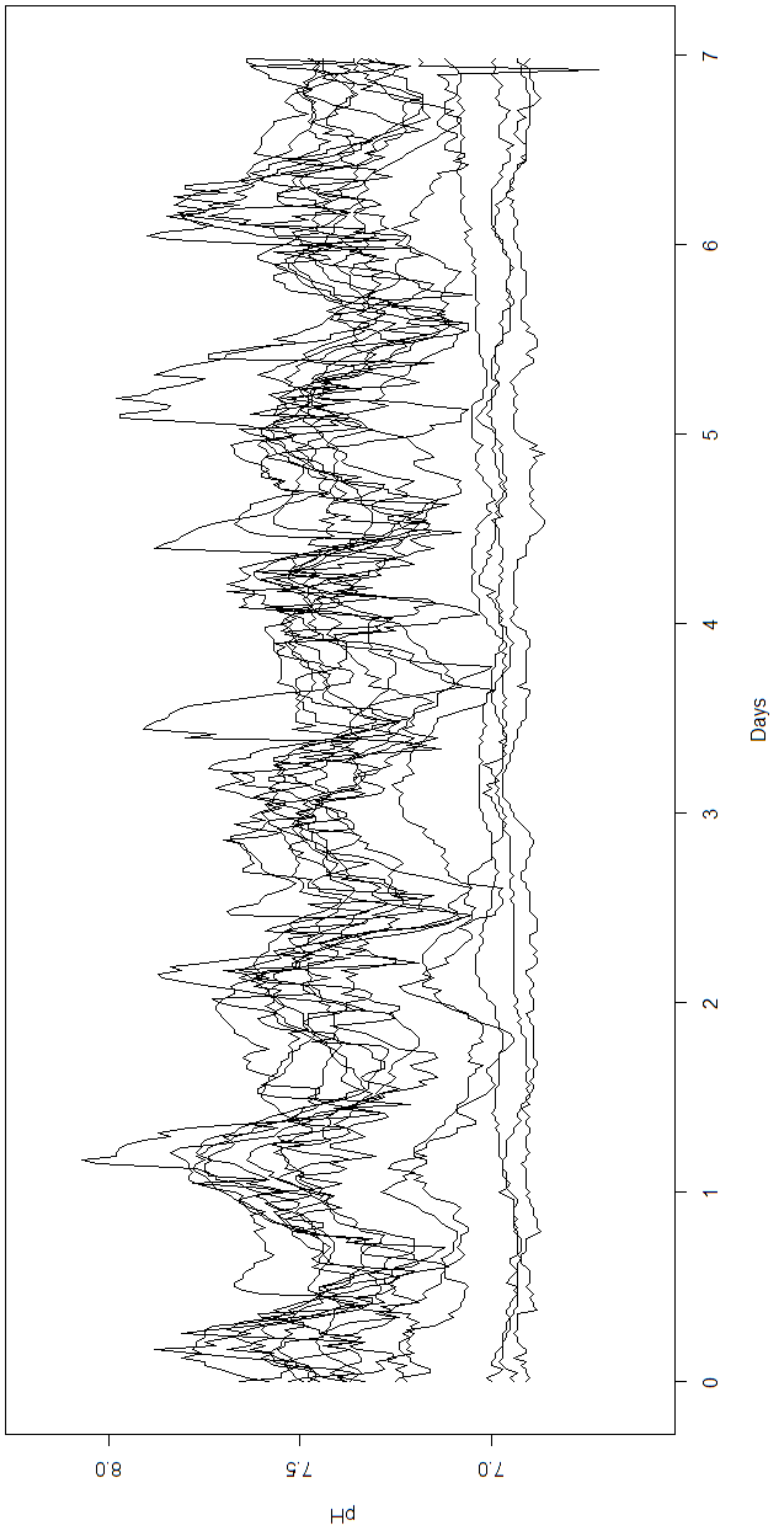

Figure 38. pH weekly overlapping time series

Time series (conductivity)

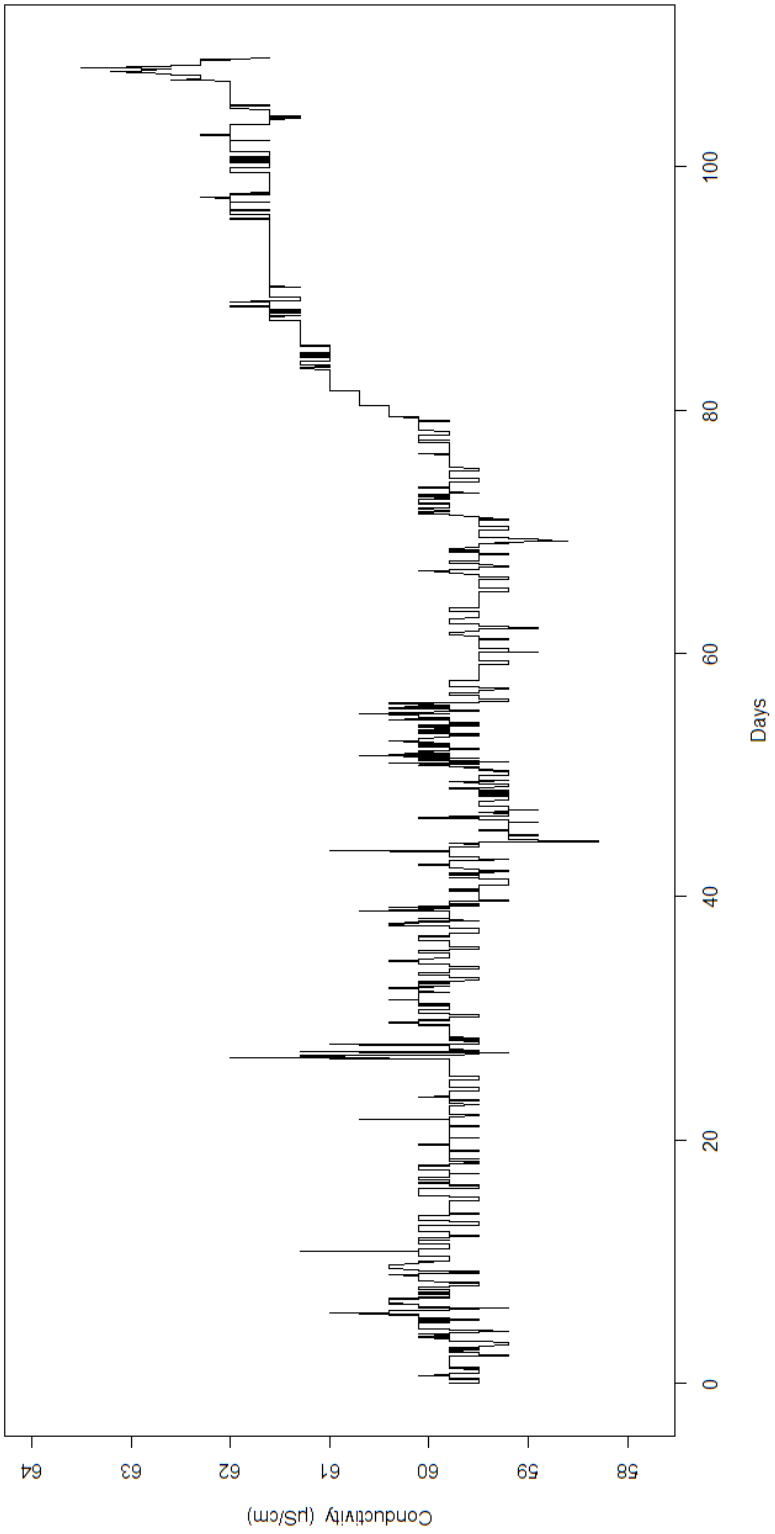

Figure 39. Conductivity time series

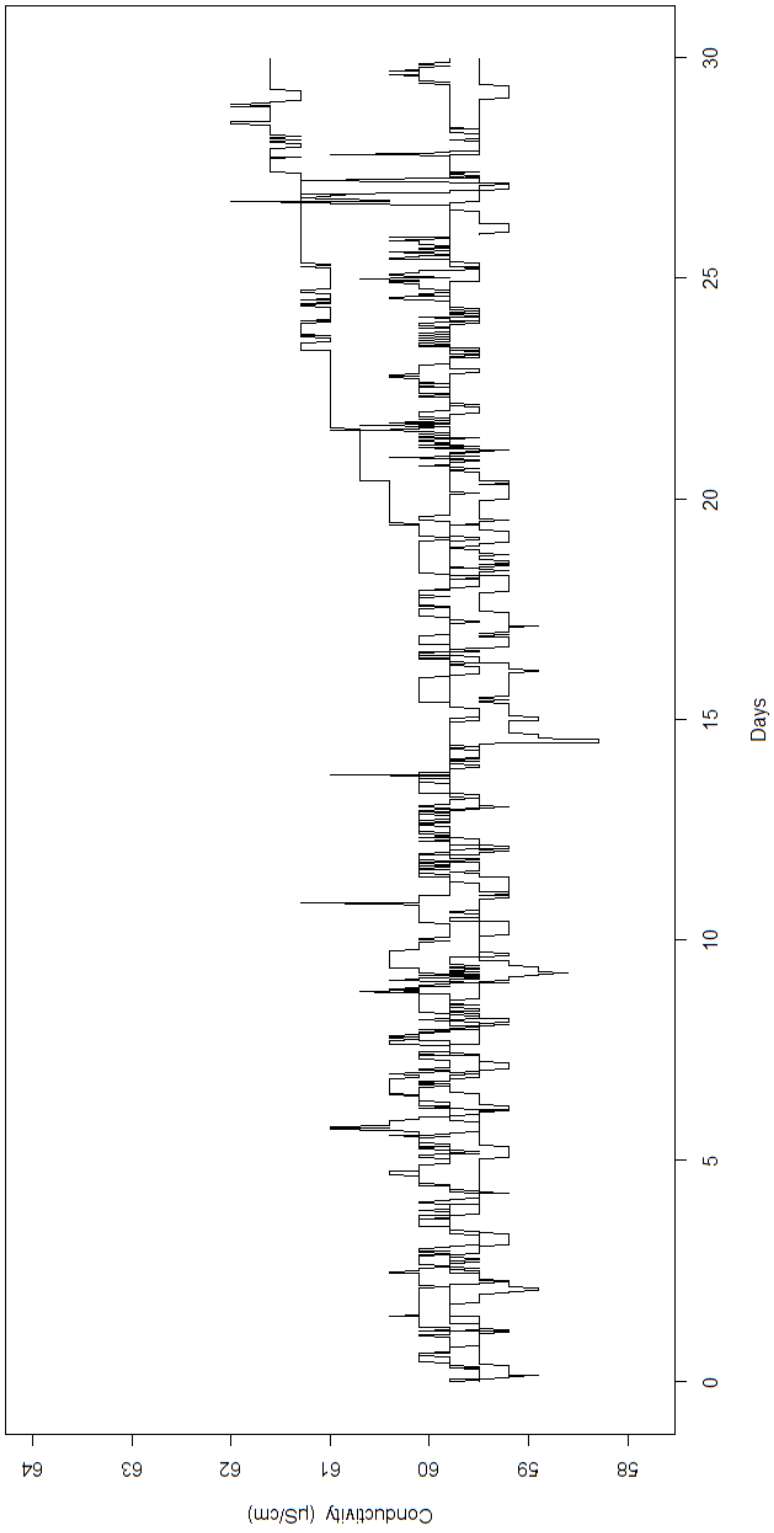

Figure 40. Conductivity monthly (30 days) overlapping time series

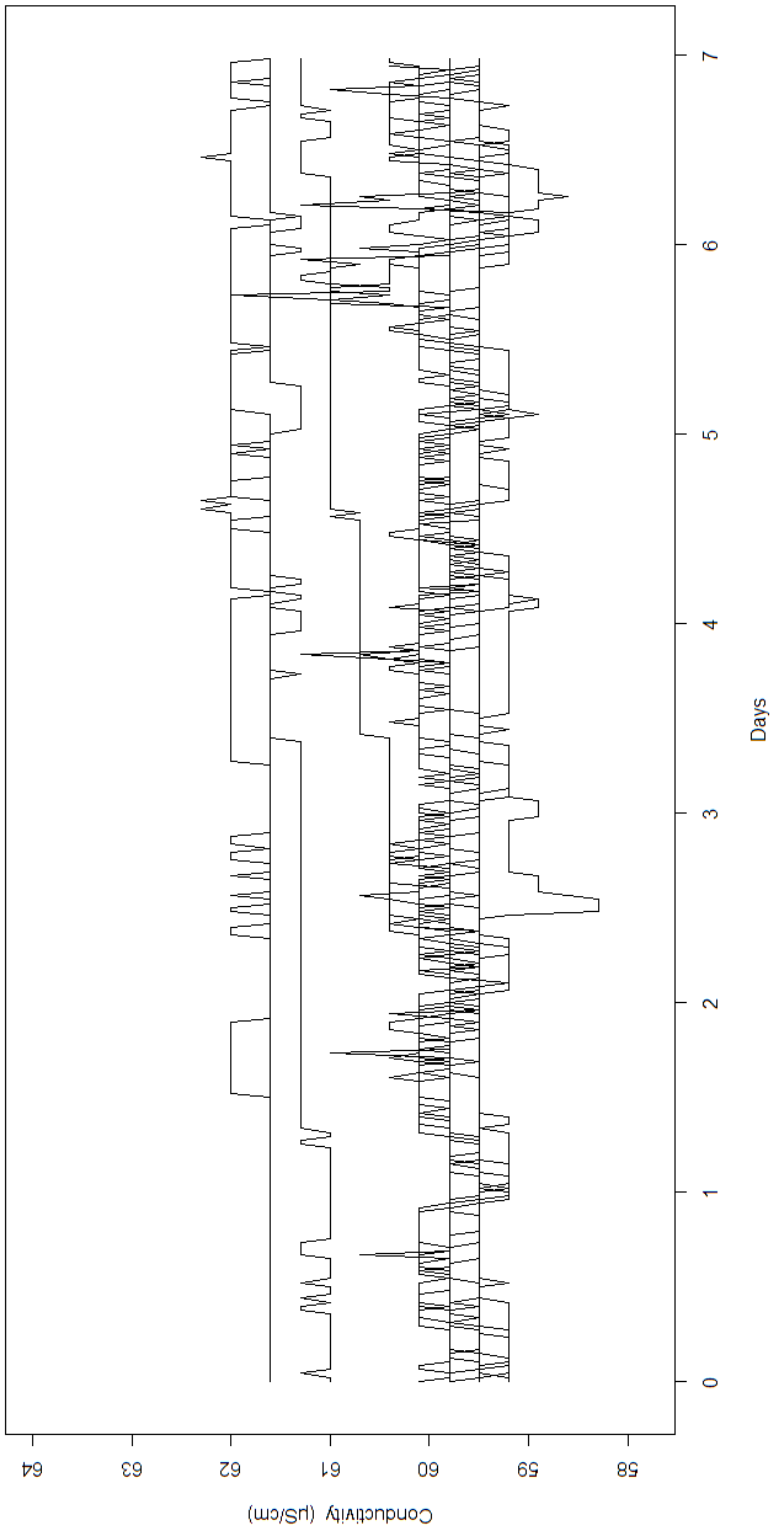

Figure 41. Conductivity weekly overlapping time series

Time series (temperature)

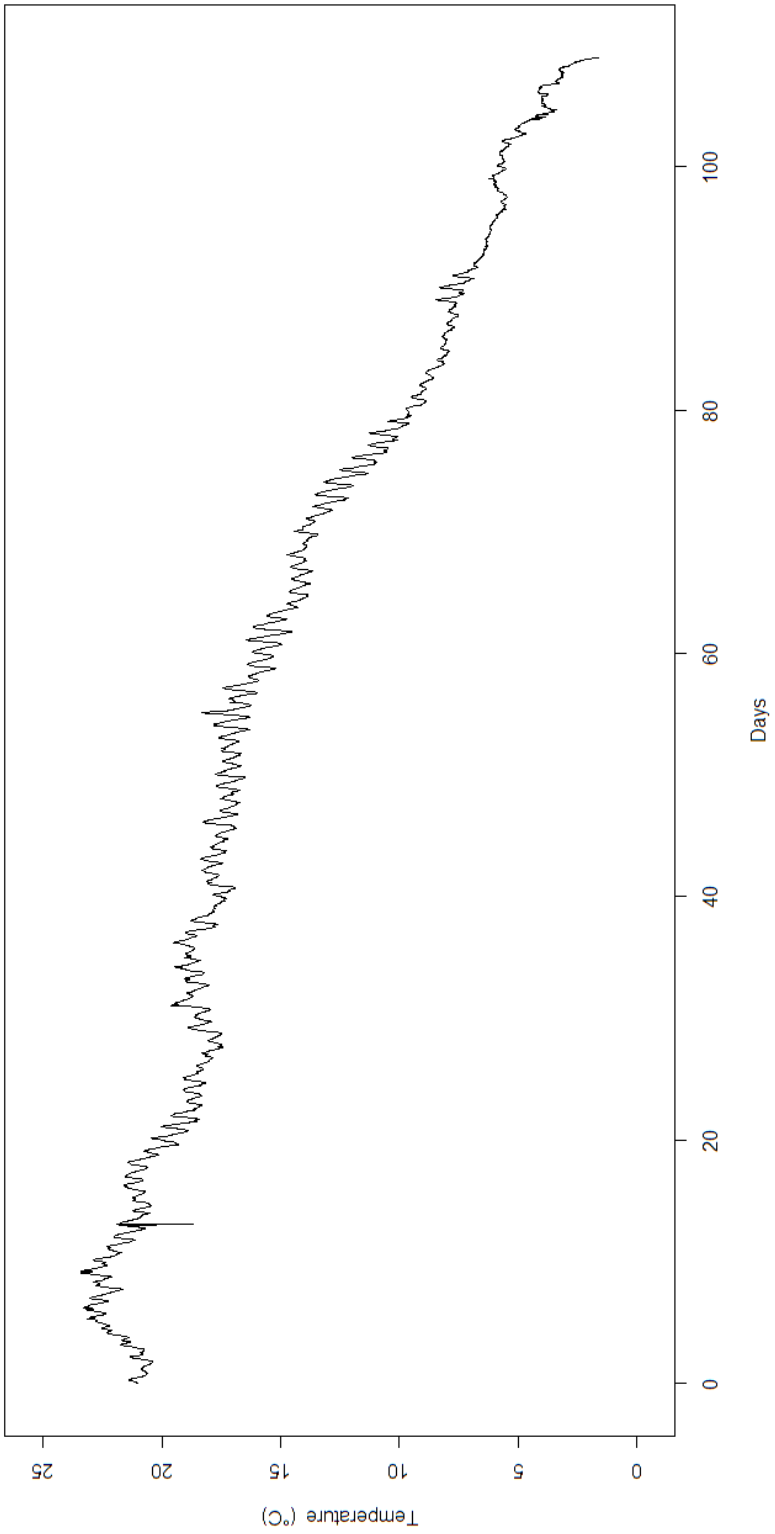

Figure 42. Temperature time series

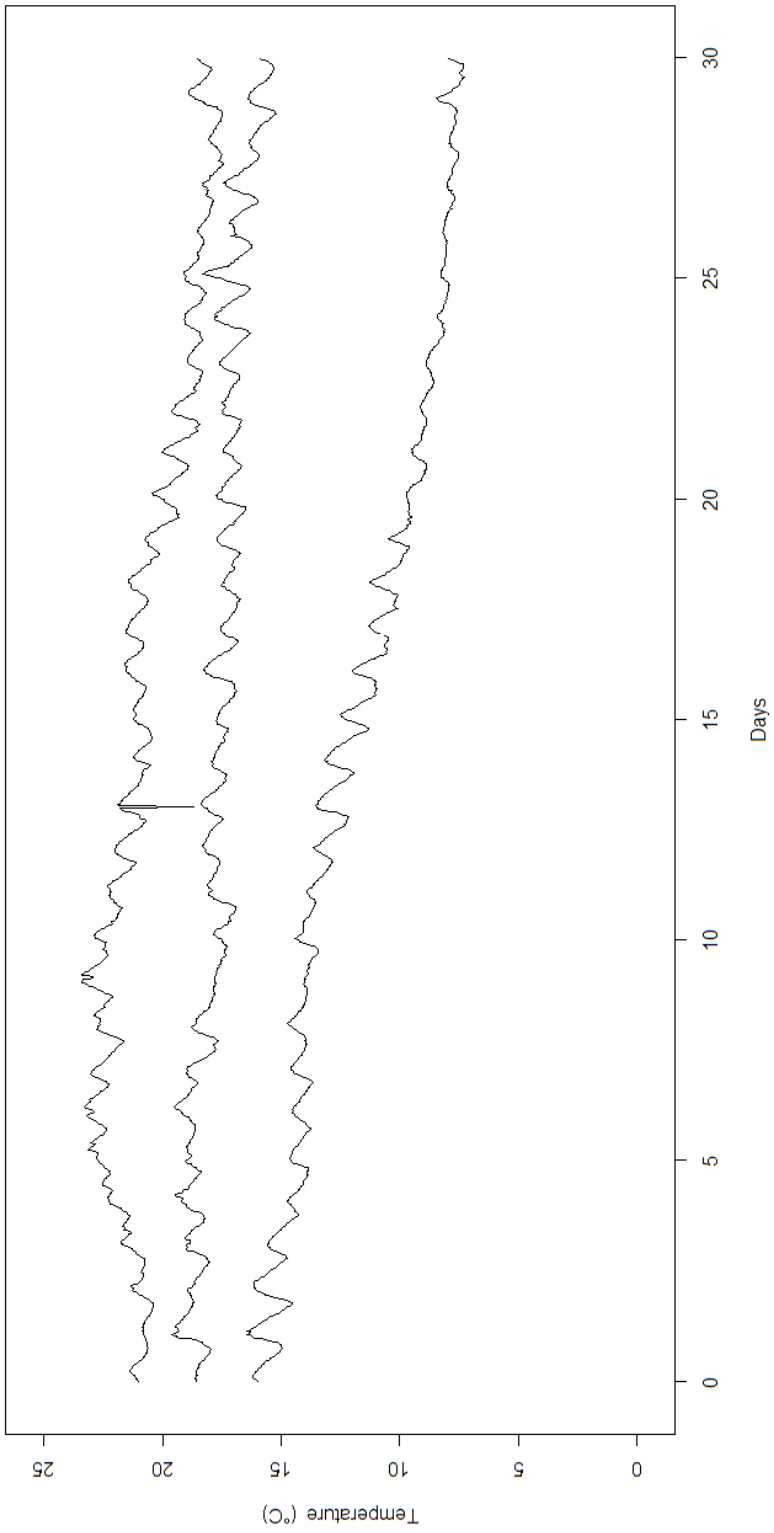

Figure 43. Temperature monthly (30 days) overlapping time series

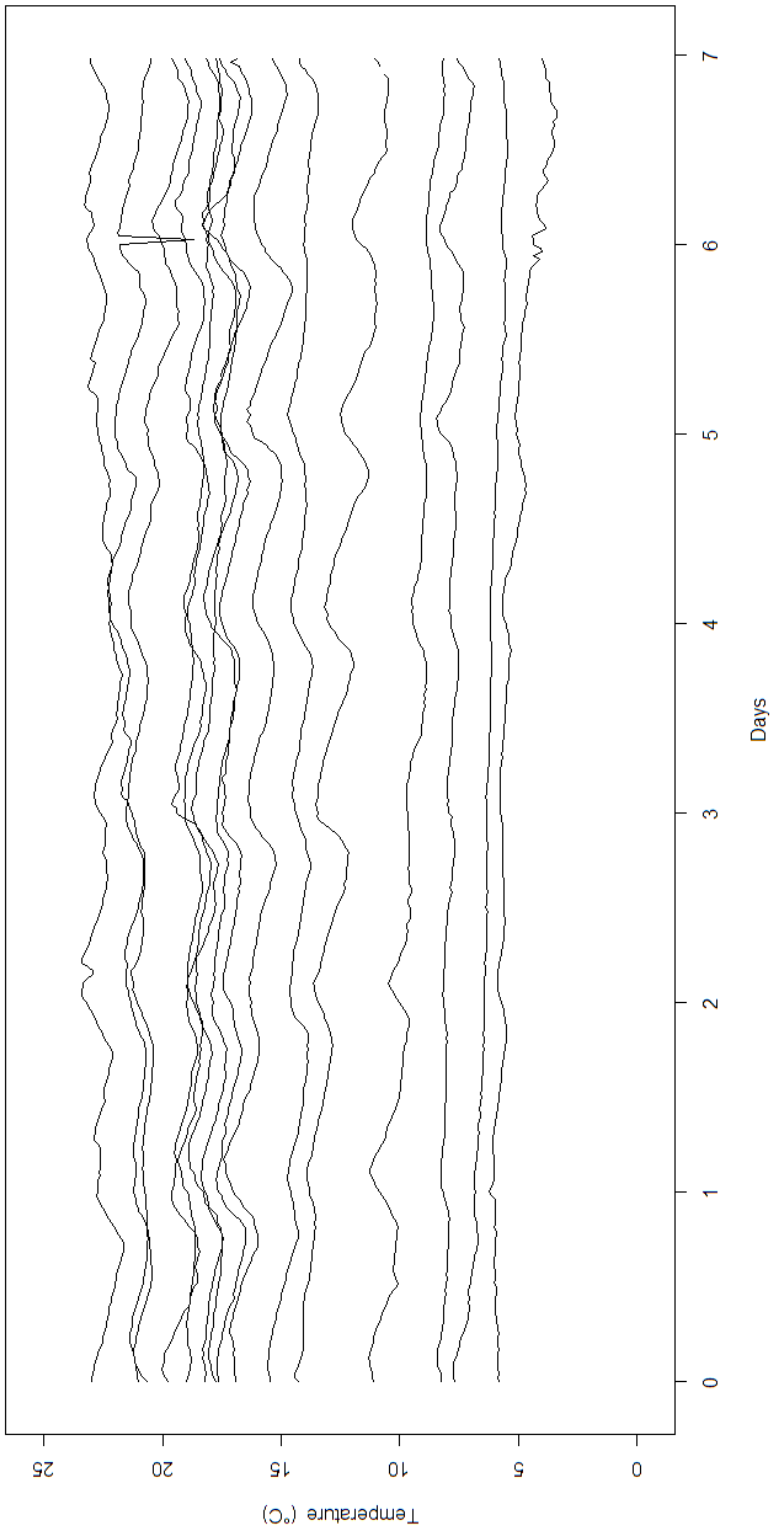

Figure 44. Temperature weekly overlapping time series

Time series (turbidity)

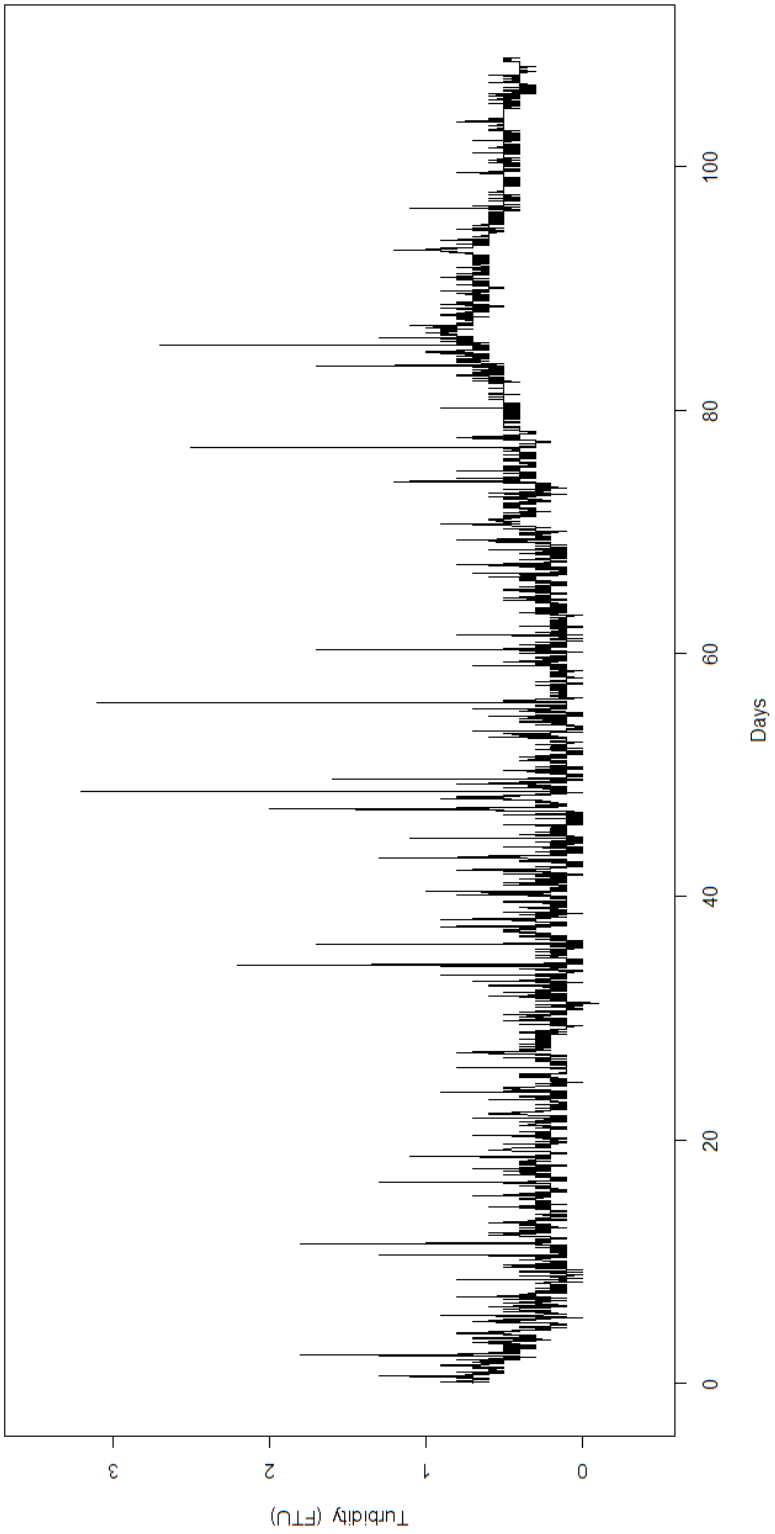

Figure 45. Turbidity time series

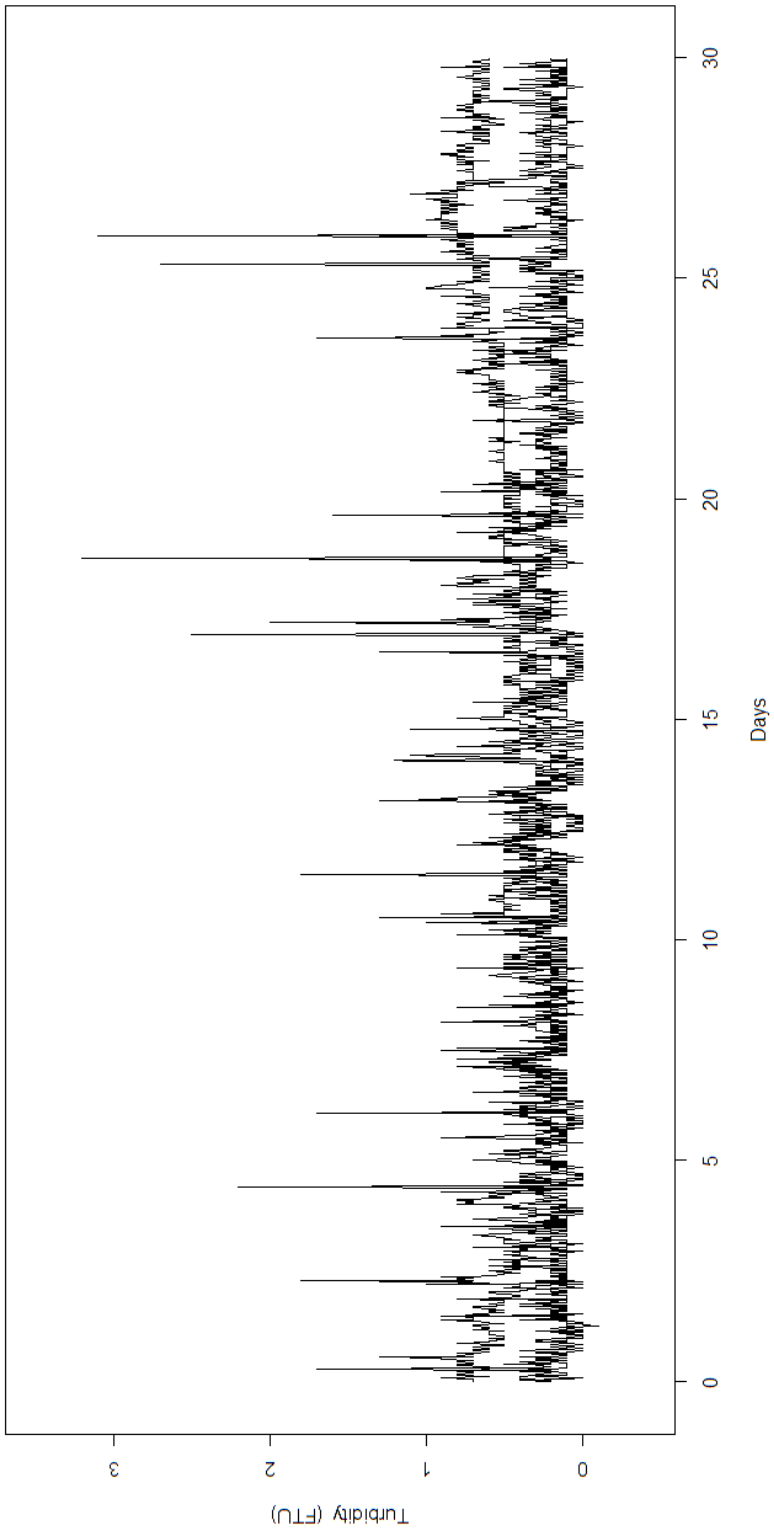

Figure 46. Turbidity monthly (30 days) overlapping time series

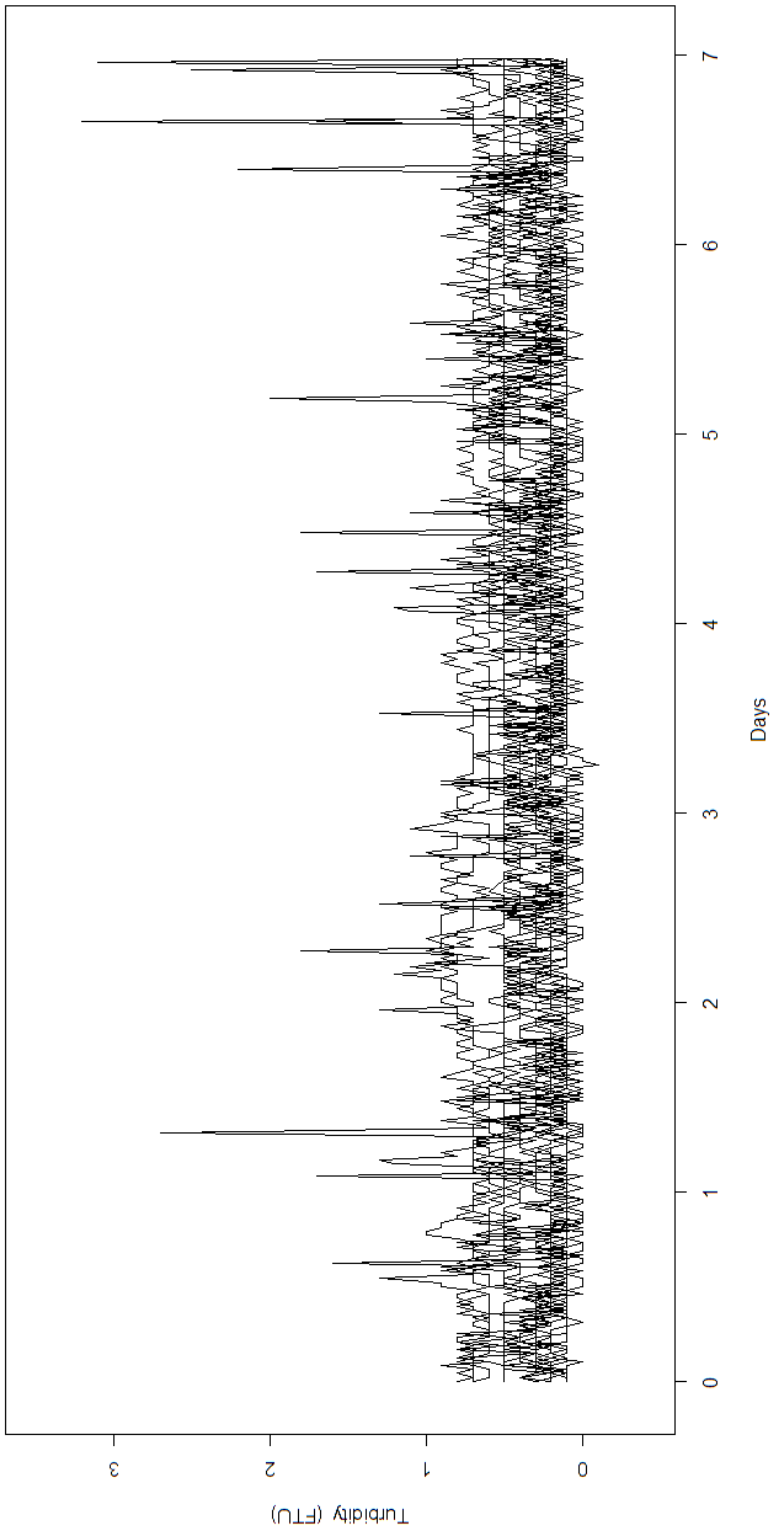

Figure 47. Turbidity weekly overlapping time series

Time series (dissolved oxygen)

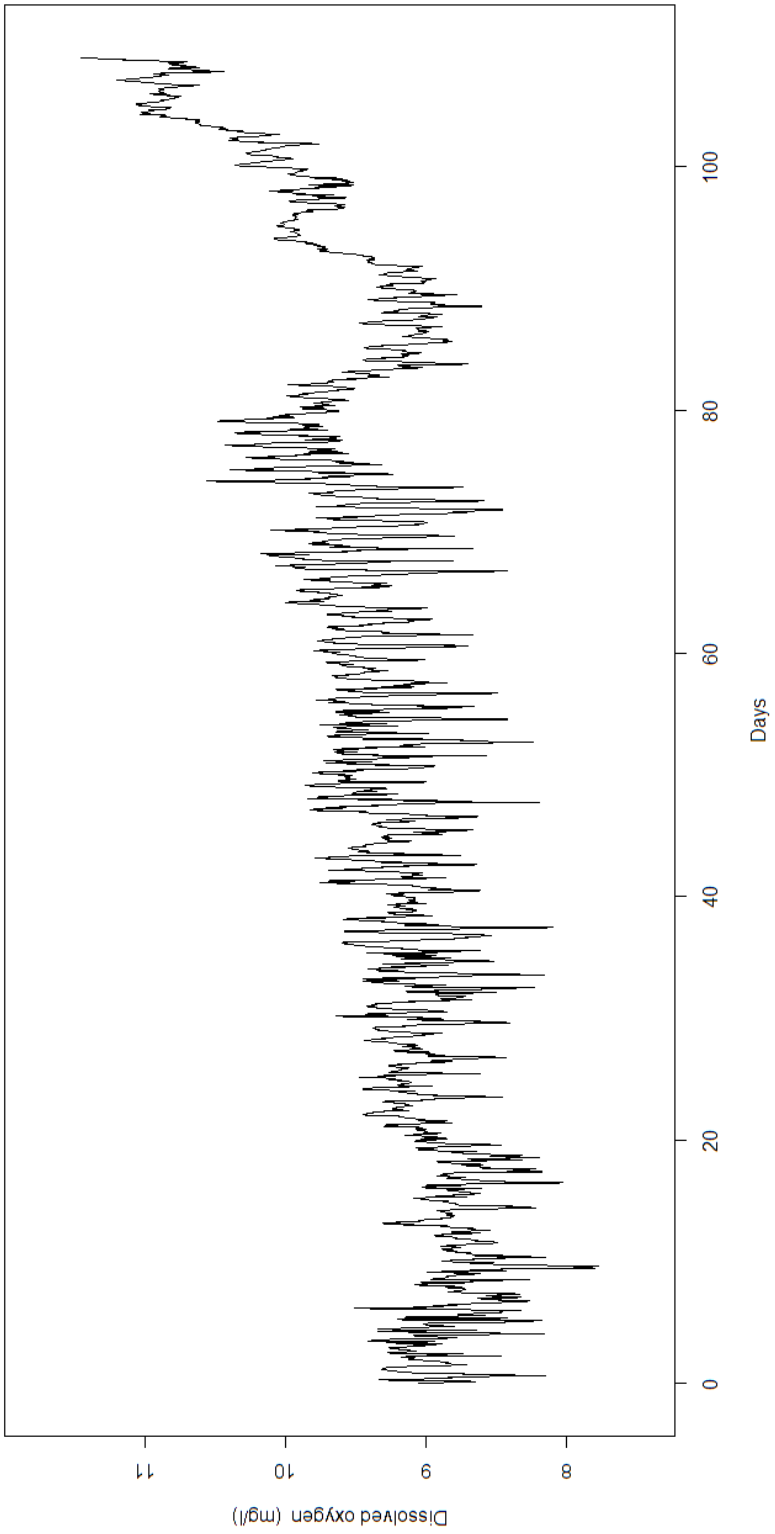

Figure 48. Dissolved oxygen time series

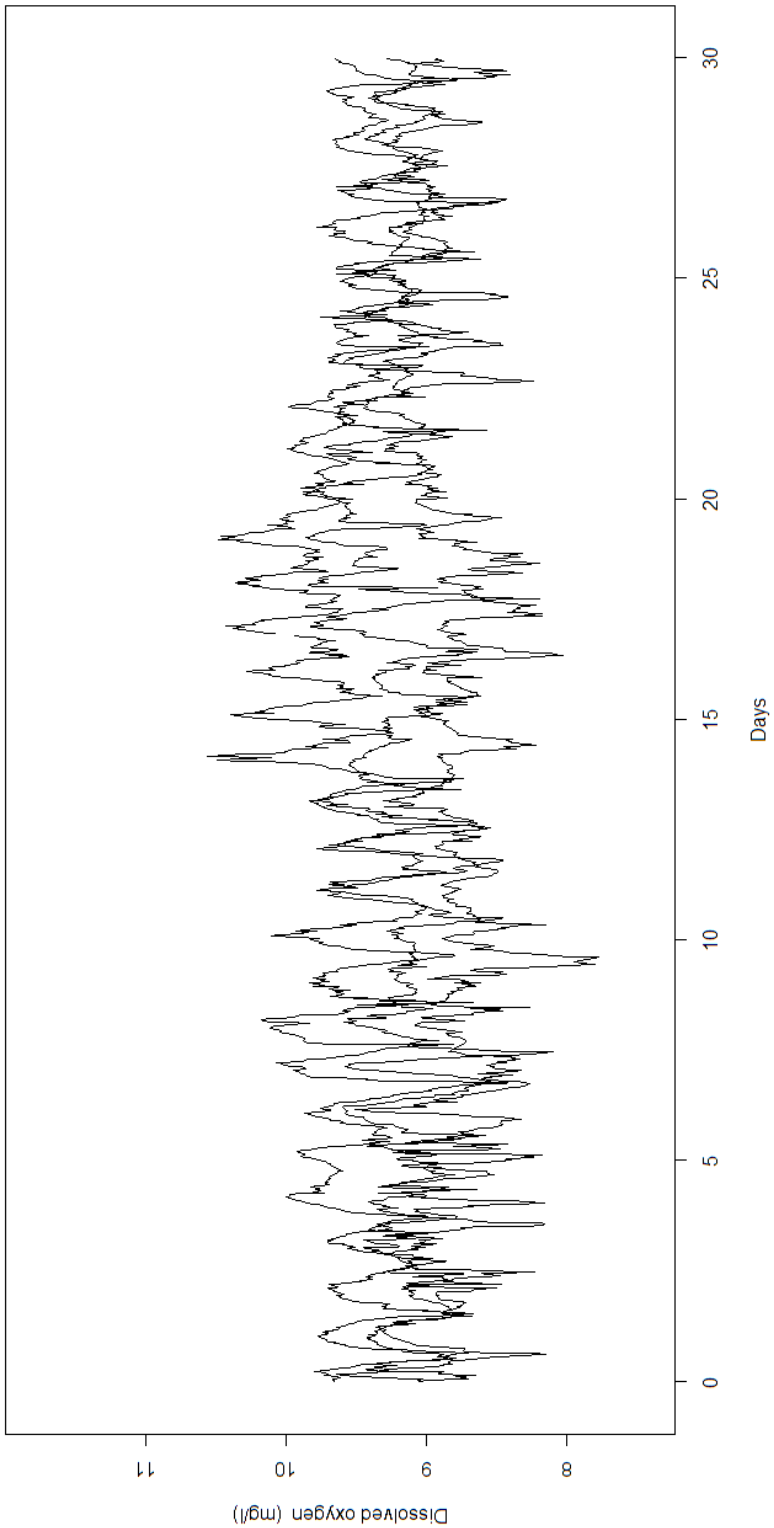

Figure 49. Dissolved oxygen monthly (30 days) overlapping time series

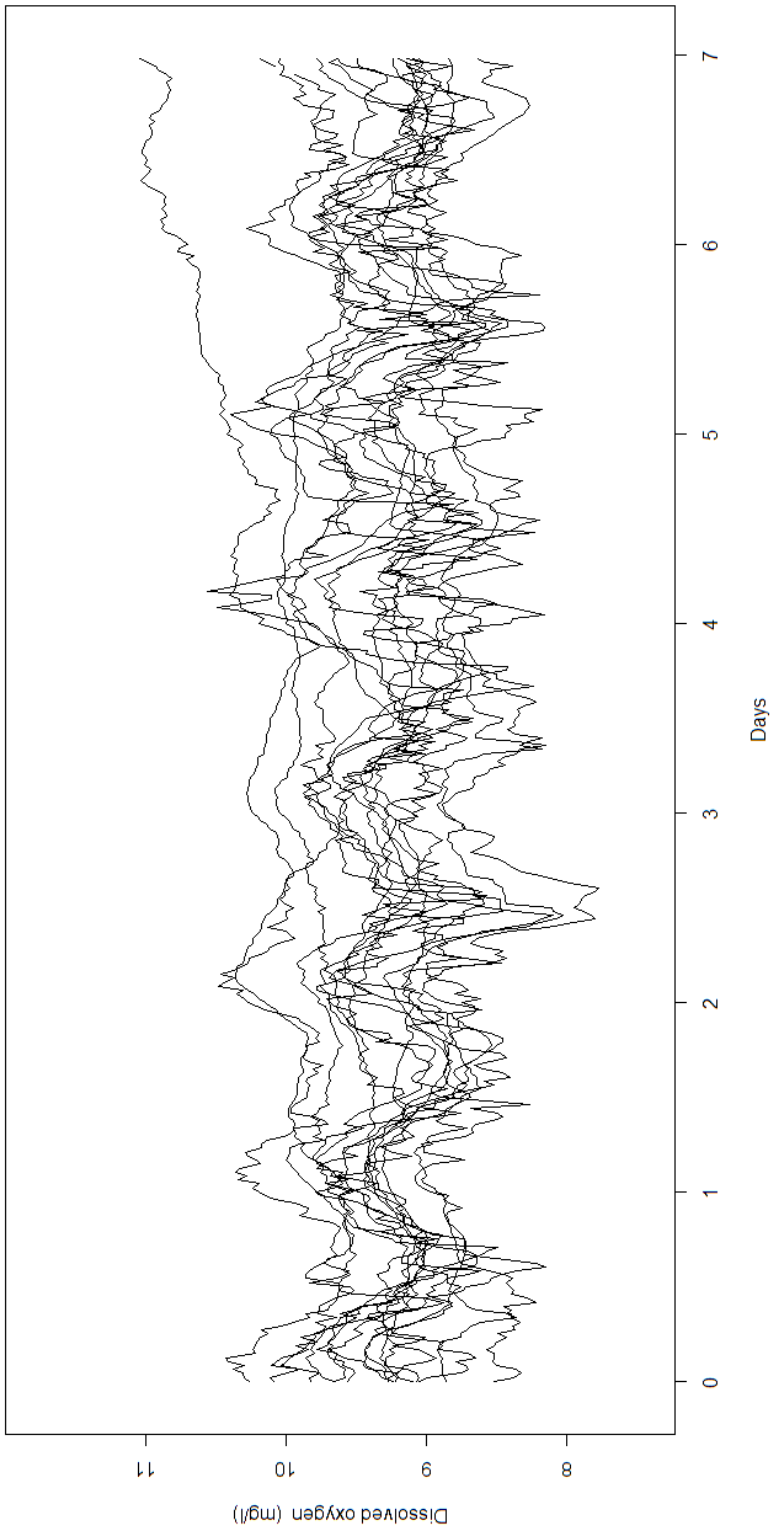

Figure 50. Dissolved oxygen weekly overlapping time series

Time series (chlorophyll-a)

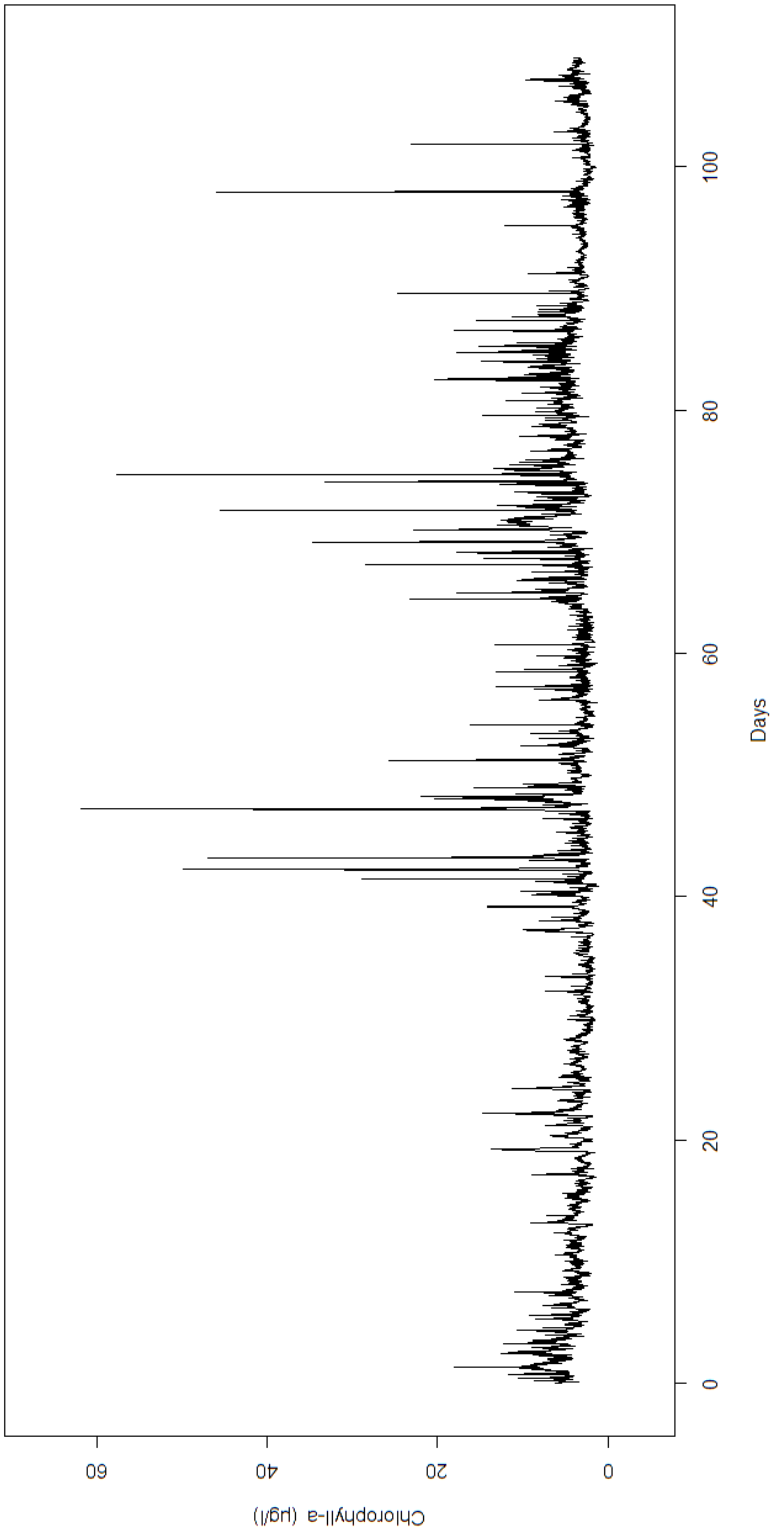

Figure 51. Chlorophyll-a time series

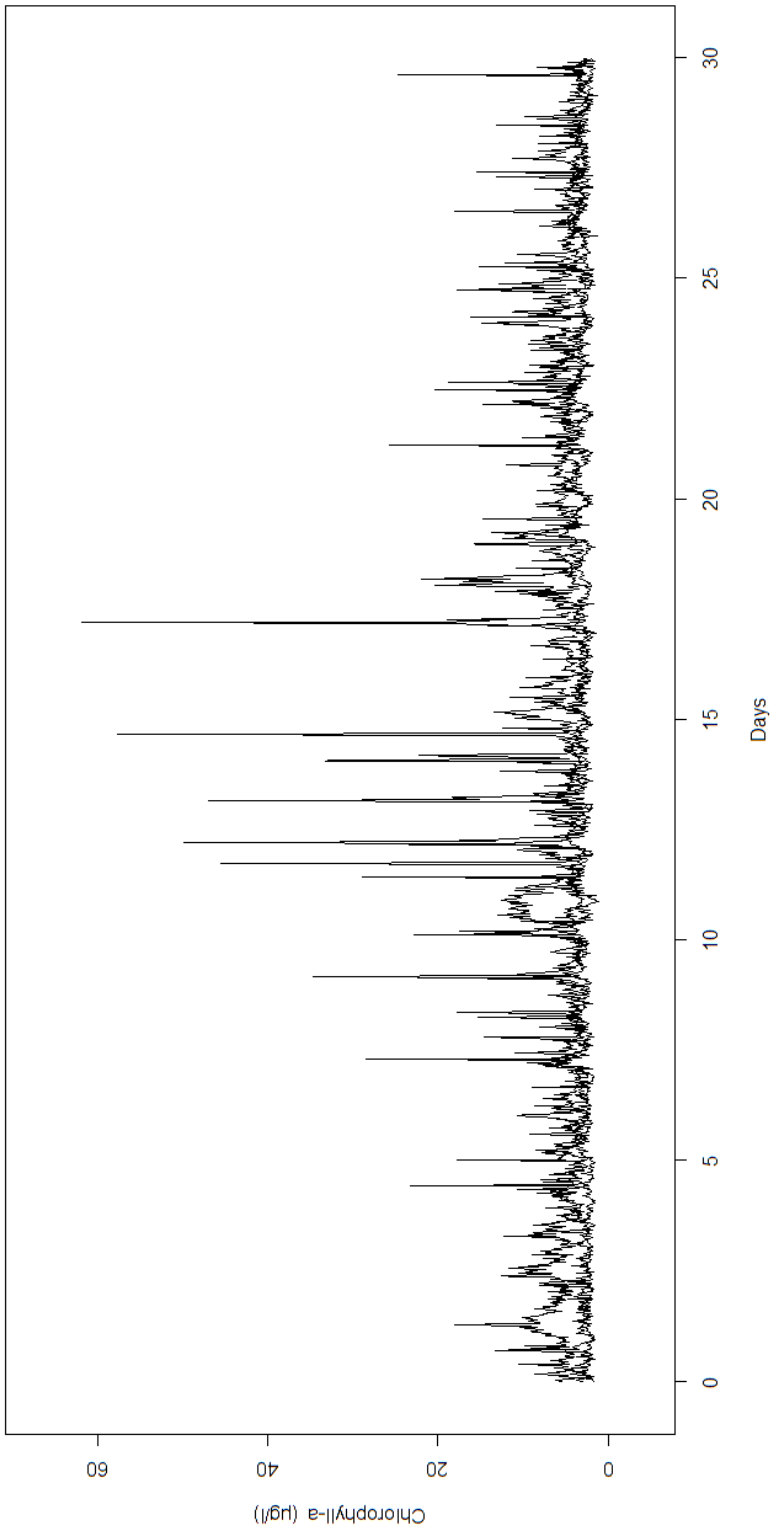

Figure 52. Chlorophyll-a monthly (30 days) overlapping time series

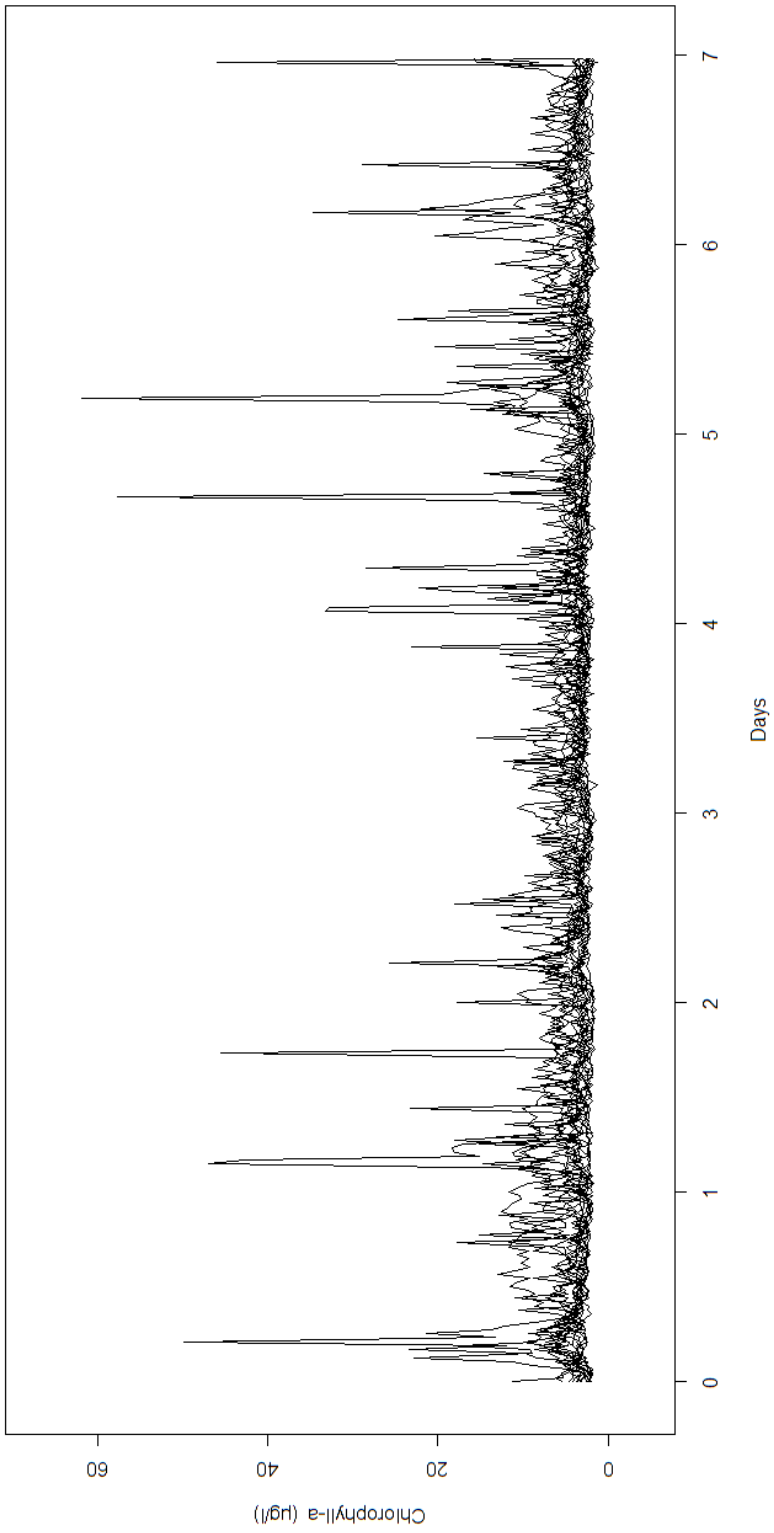

Figure 53. Chlorophyll-a weekly overlapping time series

Comparison of the turbidity and chlorophyll-a (standardized using Equation 2)

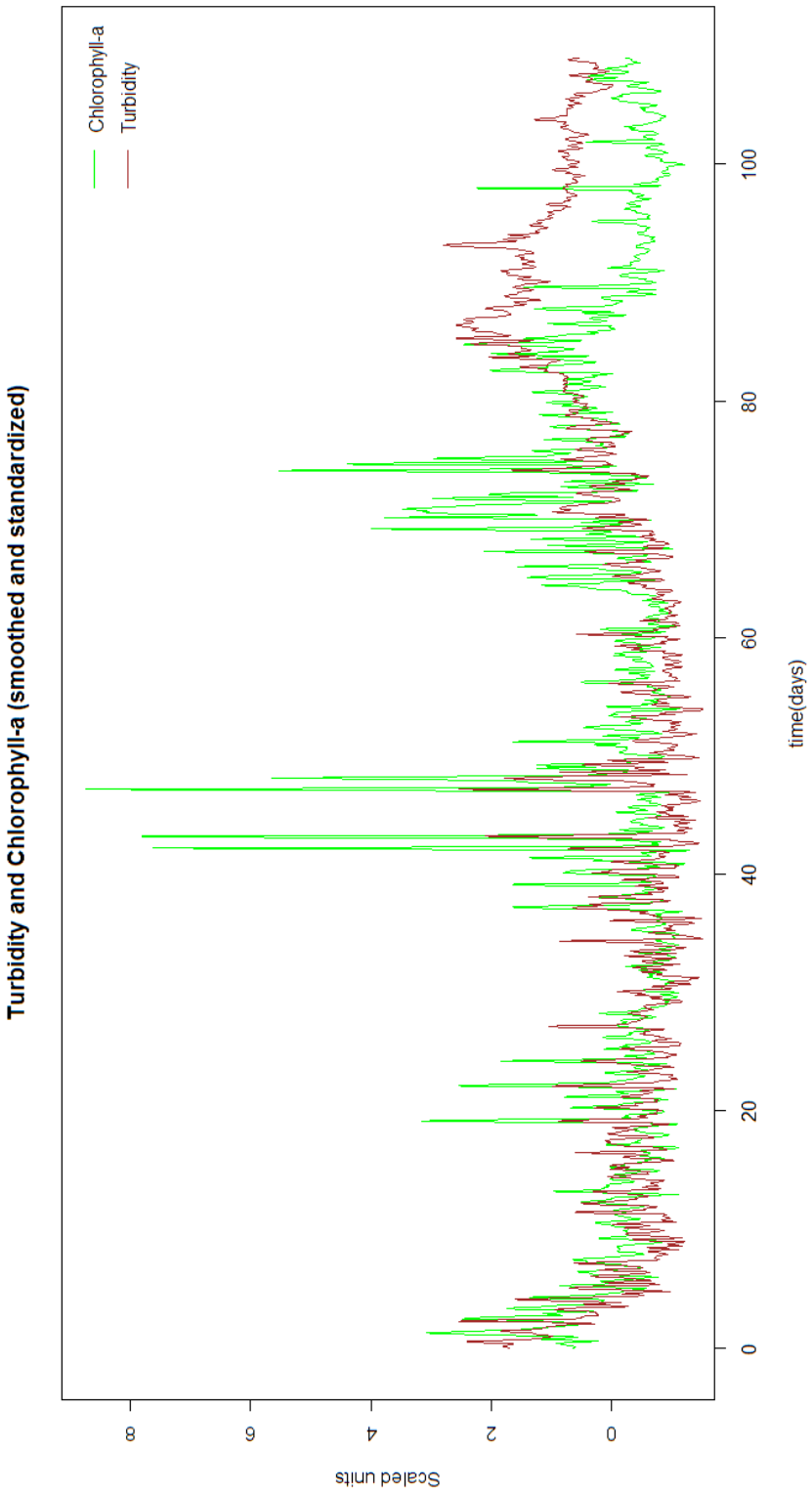

Figure 54. Turbidity and Chlorophyll-a comparison (standardized using Equation 2)

Comparison of the turbidity and chlorophyll-a (standardized using Equation 3)

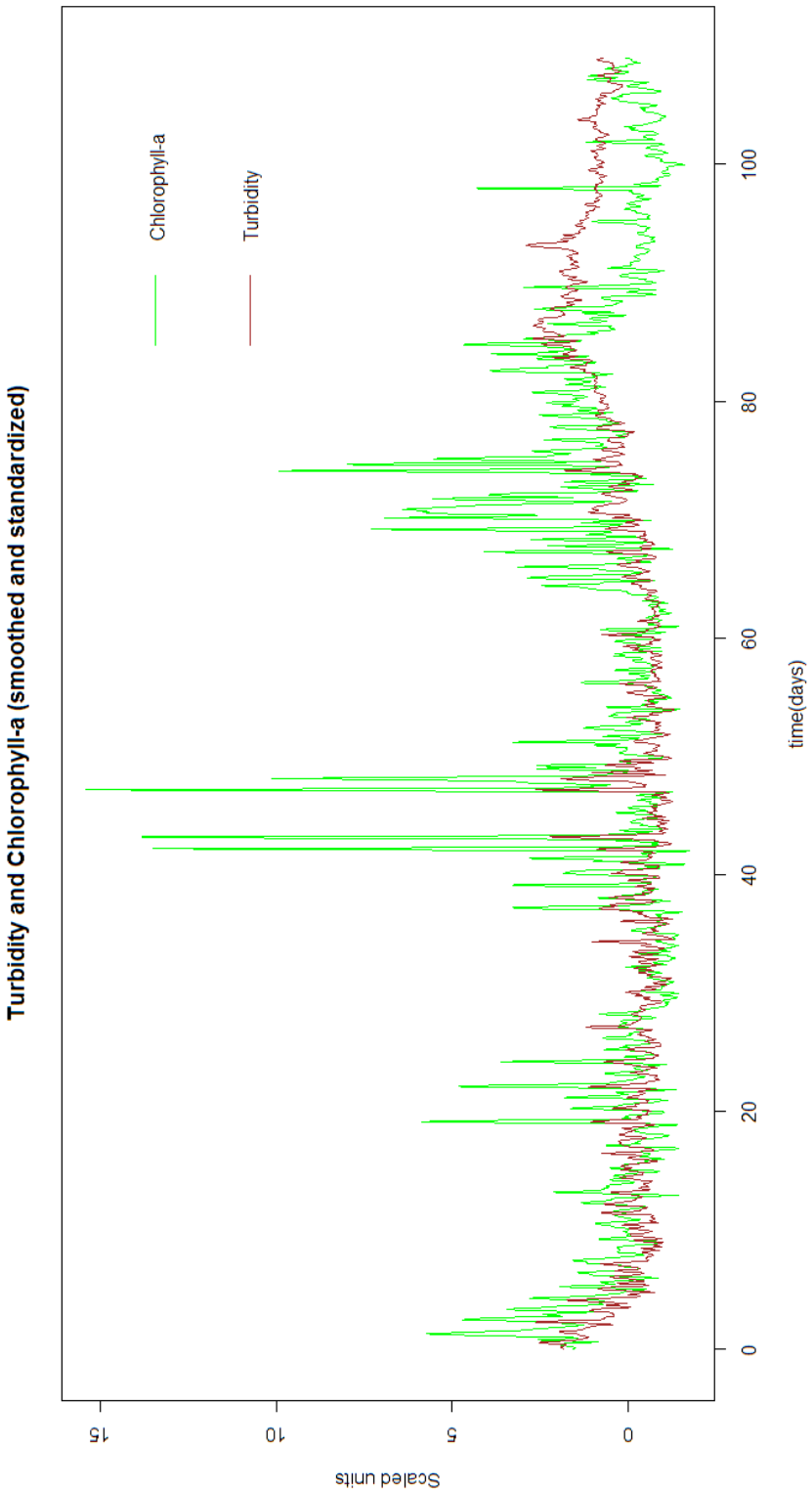

Figure 55. Turbidity and chlorophyll comparison standardized using Equation 3

Smoothing

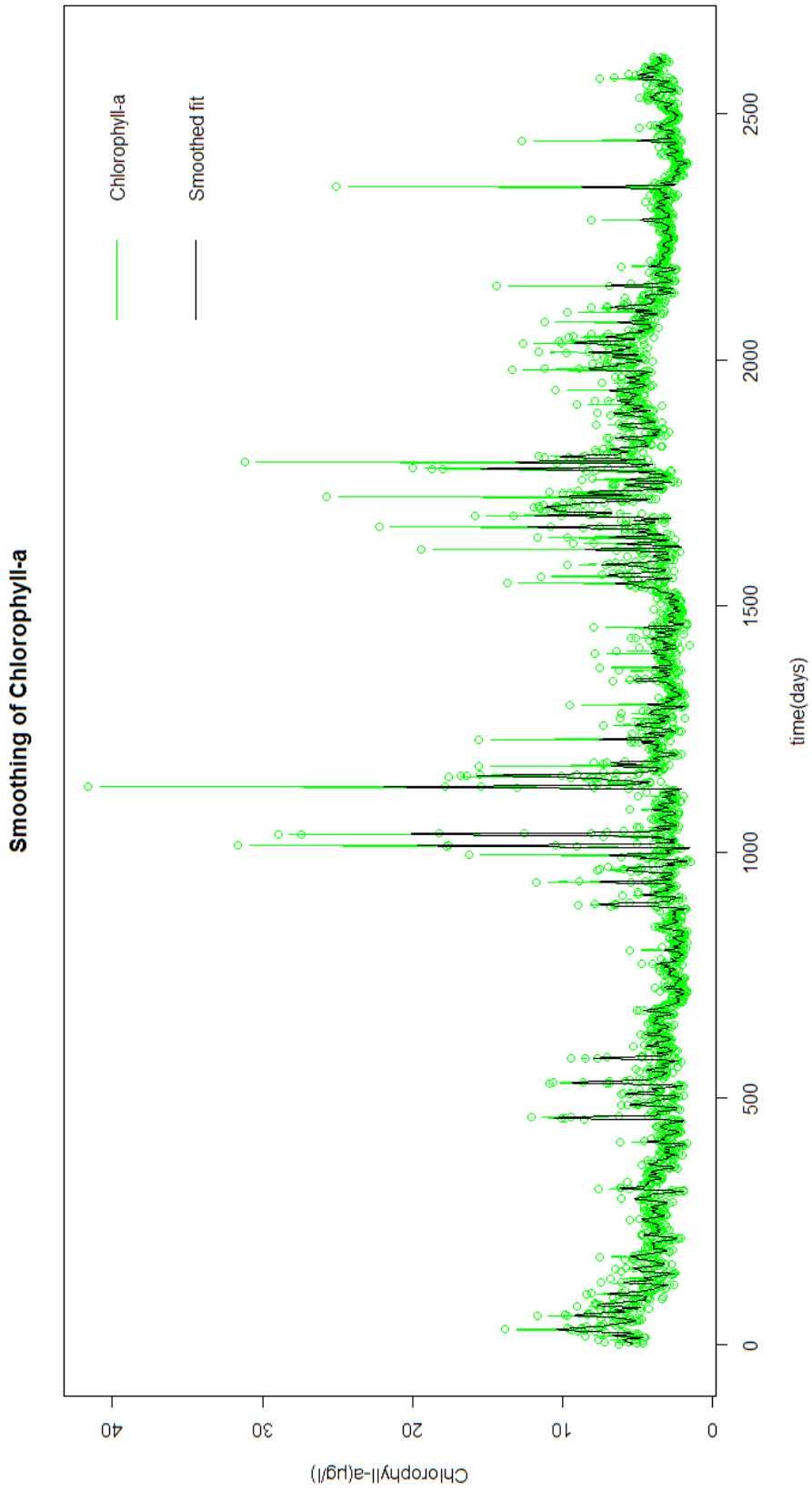

Figure 56. Chlorophyll-a smoothed curve

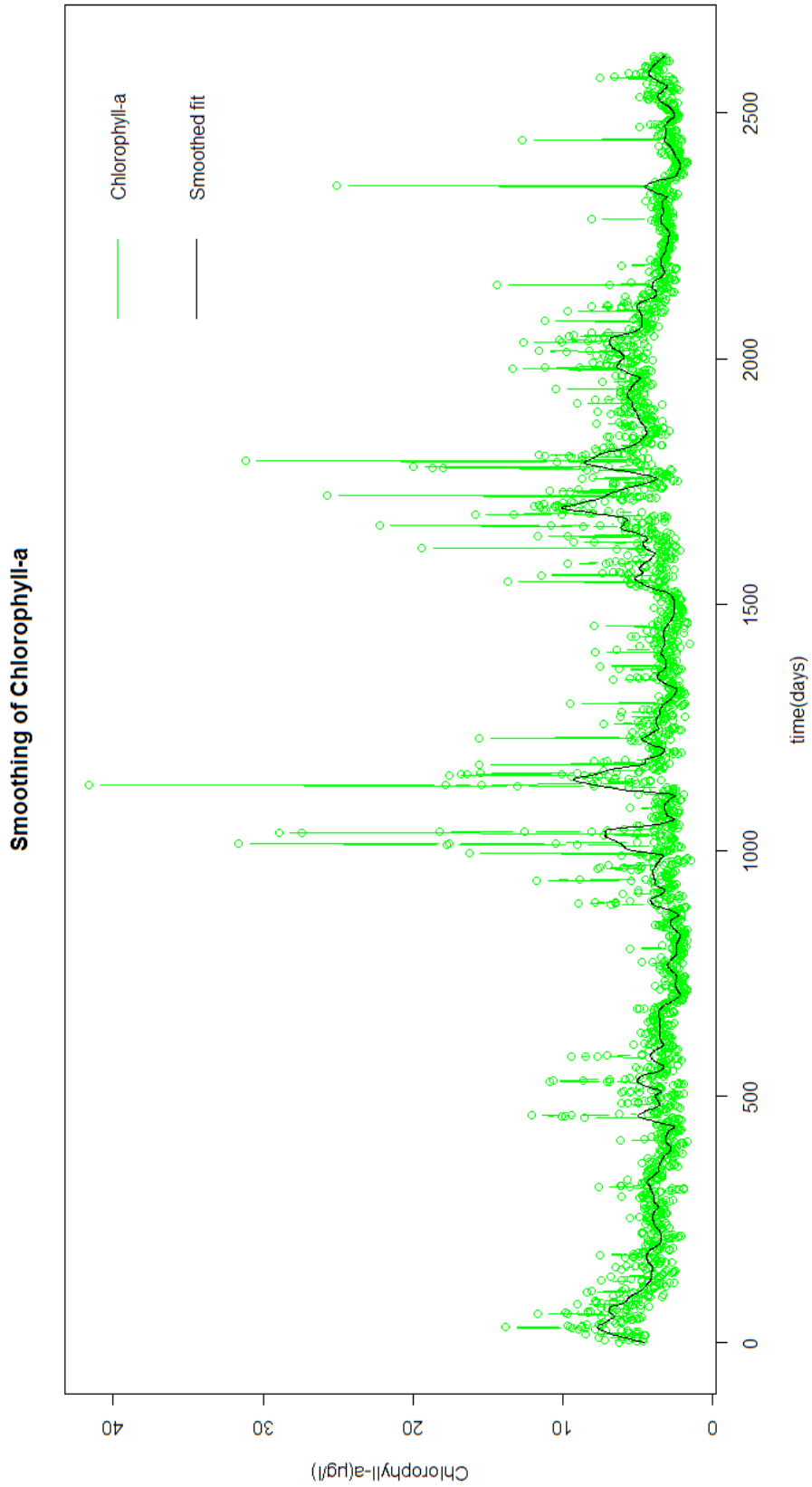

Figure 57, Chlorophyll-a strongly smoothed curve

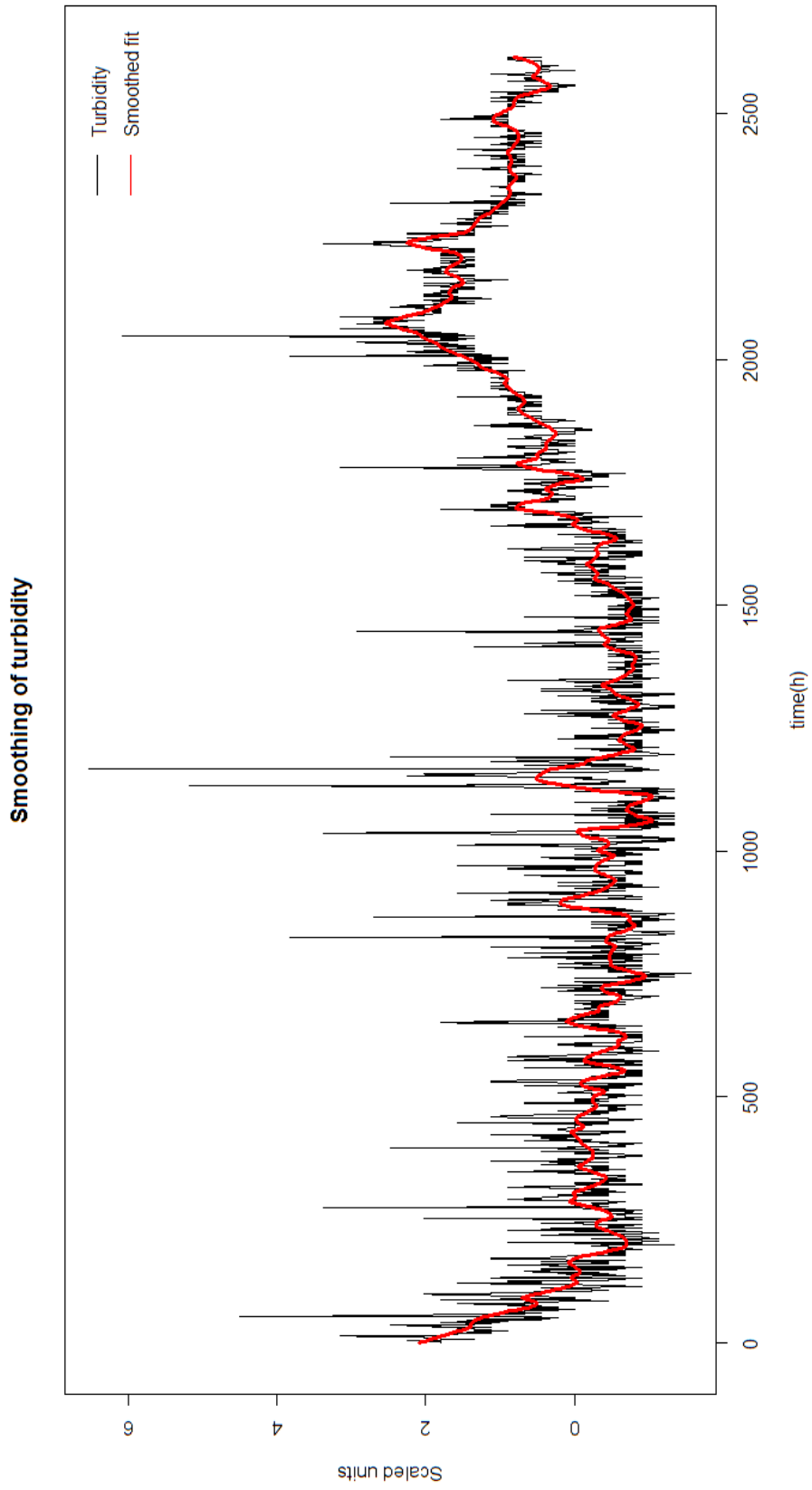

Figure 58. Turbidity smoothed curve

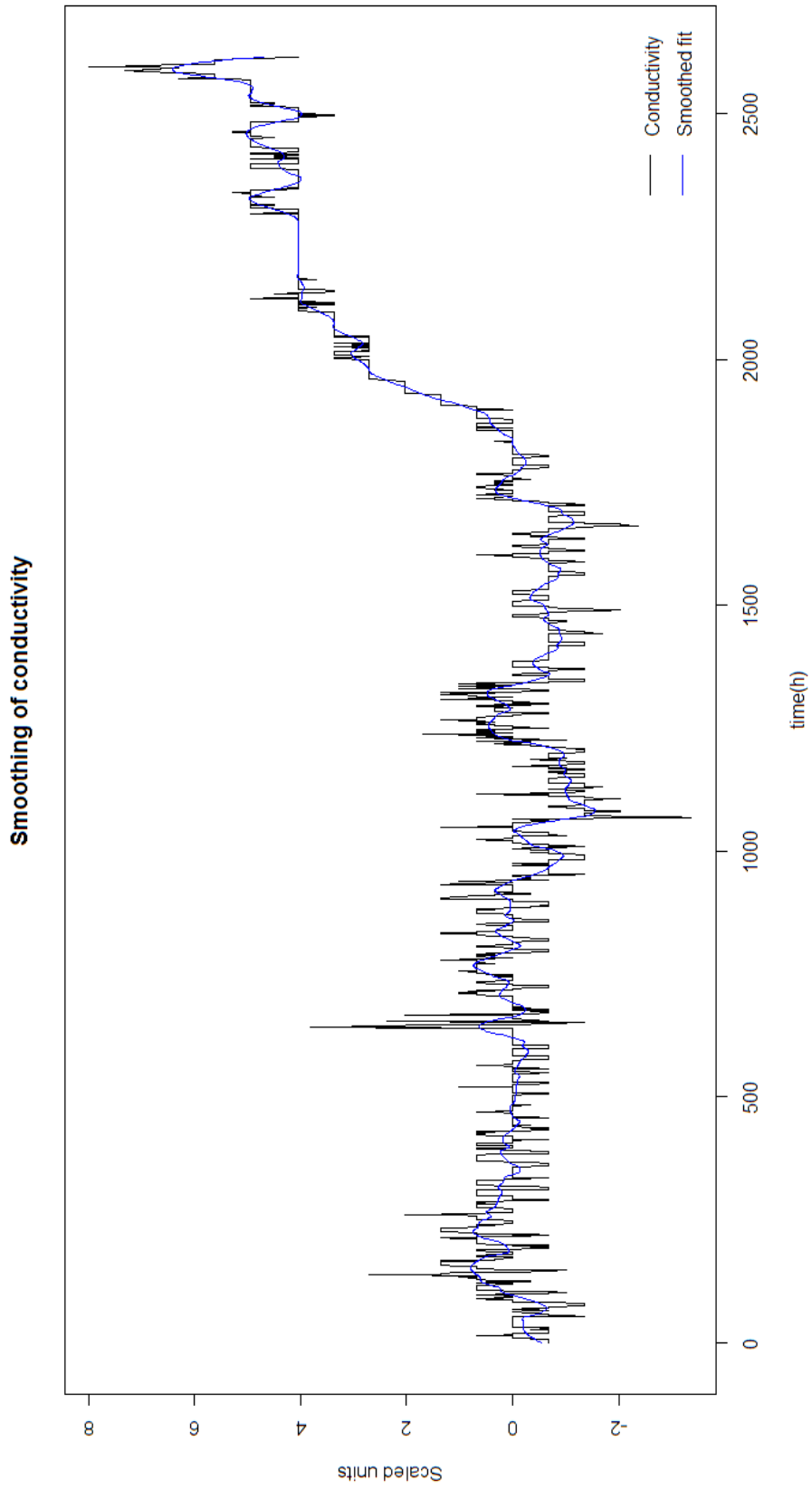

Figure 59. Conductivity smoothed curve

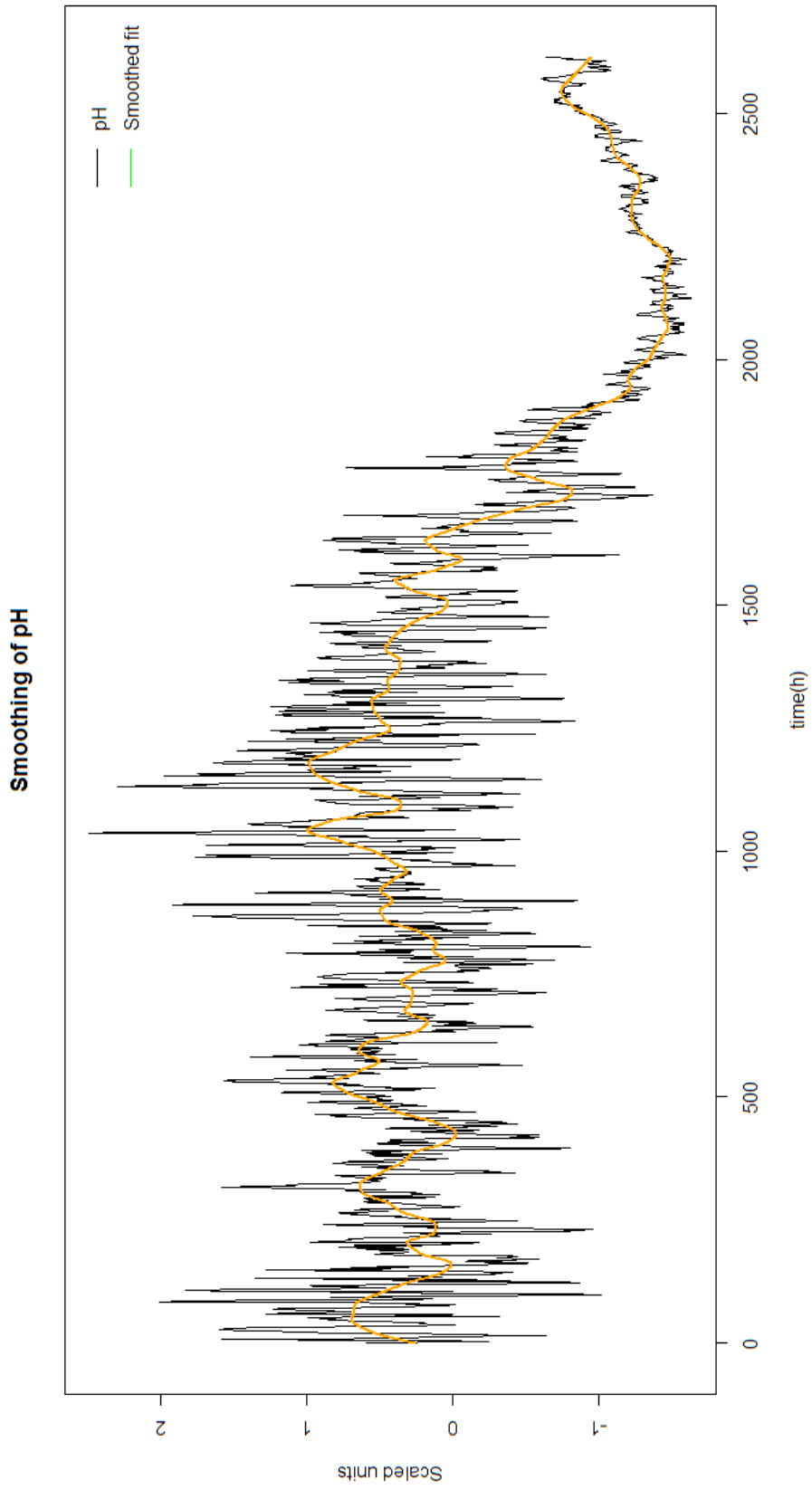

Figure 60. pH smoothed curve

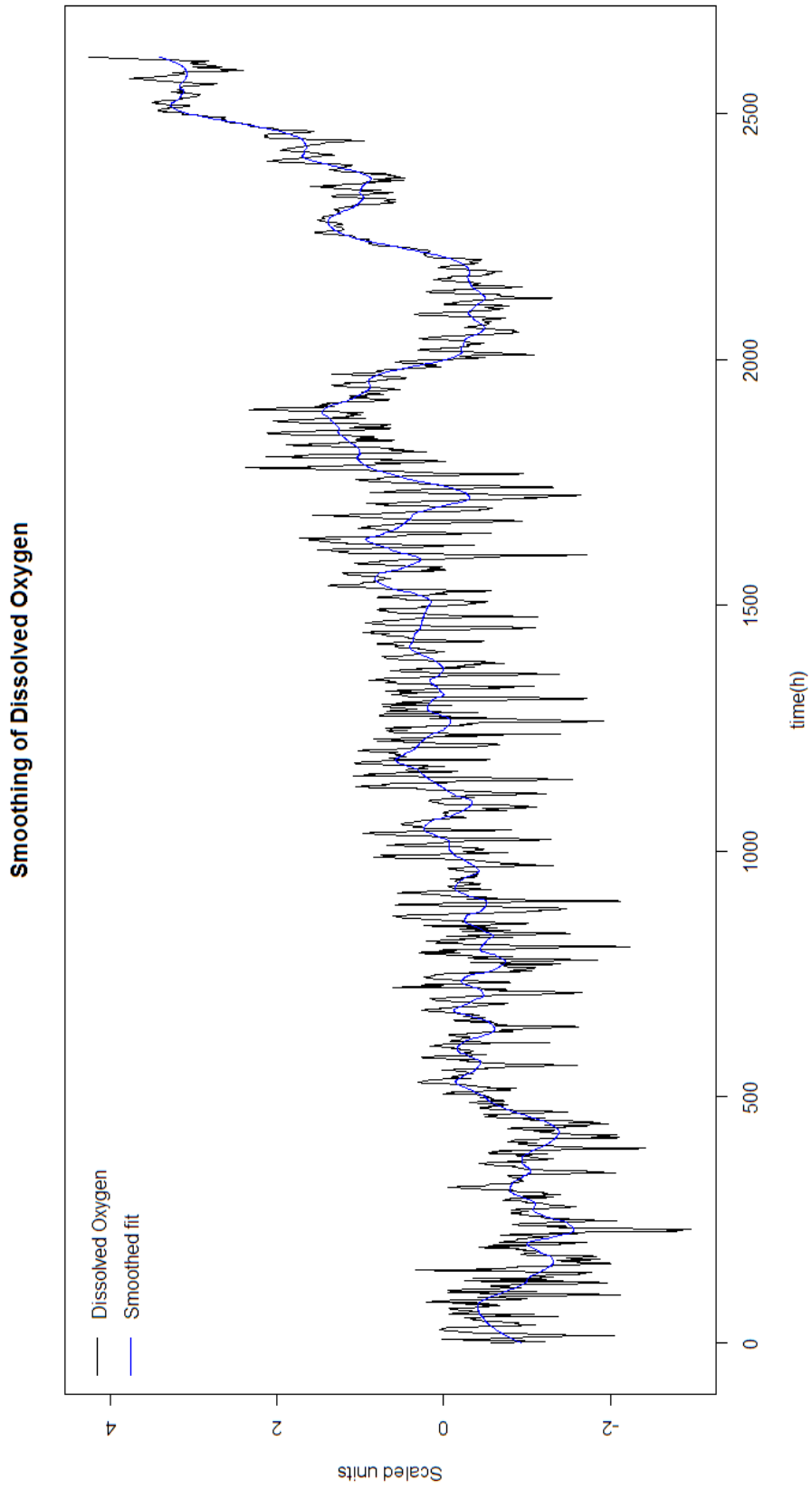

Figure 61. Dissolved Oxygen smoothed curve

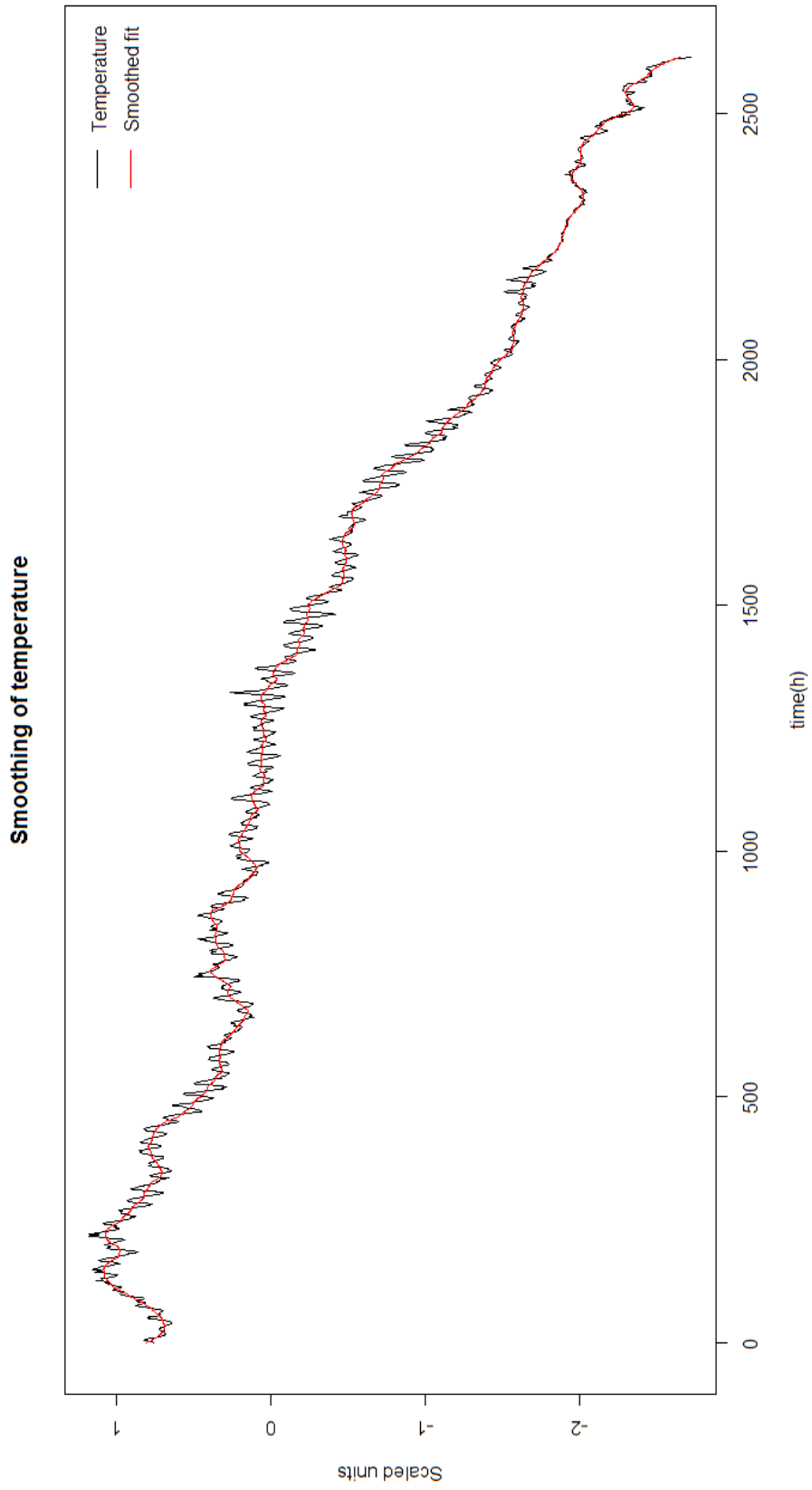

Figure 62. Temperature smoothed curve

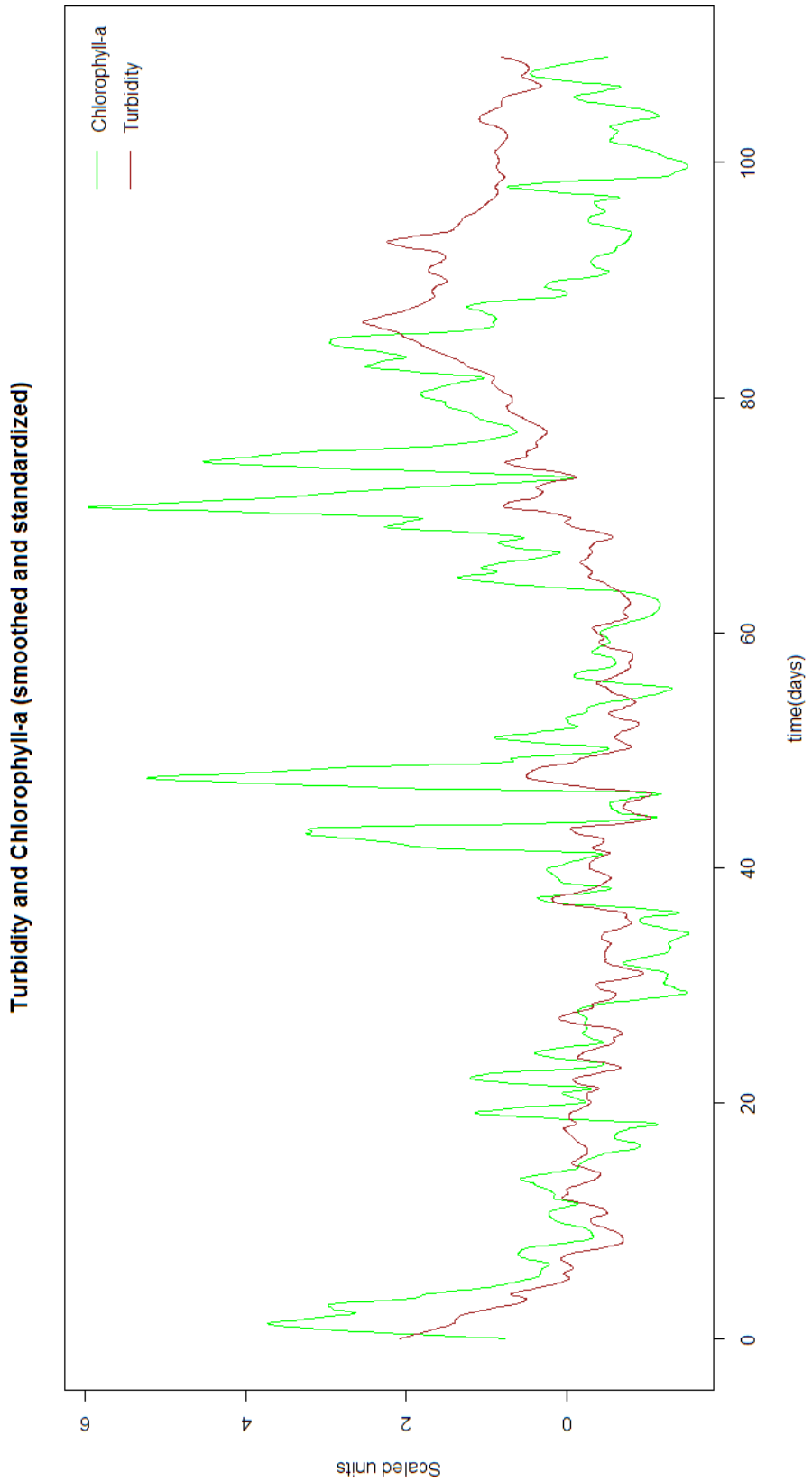

Figure 63. Comparison of turbidity and chlorophyll-a from smooth fits

Results for MLR1 (using standardization defined in Equation 2) – Run #1

Formula:

```
y ~ turbidity + I(conductivity * turbidity) + I(pH^2) + conductivity +
    temperature + I(pH * DO) + I(conductivity^2) + I(temperature *
    conductivity) + I(temperature * turbidity) + I(pH * turbidity) +
    I(conductivity * DO) + I(temperature^2) + I(DO^2) + pH +
    DO + I(temperature * DO) + I(temperature * pH)
```

Model summary:

```
Residuals:
     Min       1Q   Median       3Q      Max
-19.3356  -0.7823  -0.1999   0.5633  24.8196

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                   4.13701    0.14449  28.632  < 2e-16 ***
turbidity                     1.28710    0.06827  18.853  < 2e-16 ***
I(conductivity * turbidity)  -1.11034    0.12491  -8.889  < 2e-16 ***
I(pH^2)                       1.25007    0.22228   5.624 2.12e-08 ***
conductivity                 -0.99252    0.15712  -6.317 3.25e-10 ***
temperature                  -1.96088    0.35897  -5.462 5.26e-08 ***
I(pH * DO)                   -1.13590    0.45851  -2.477  0.01331 *
I(conductivity^2)             0.83974    0.11490   7.308 3.84e-13 ***
I(temperature * conductivity) 2.45924    0.27057   9.089  < 2e-16 ***
I(temperature * turbidity)   -1.16853    0.10513 -11.116  < 2e-16 ***
I(pH * turbidity)             0.71648    0.07425   9.649  < 2e-16 ***
I(conductivity * DO)          1.03079    0.14695   7.015 3.11e-12 ***
I(temperature^2)              3.53728    0.52685   6.714 2.44e-11 ***
I(DO^2)                       0.64239    0.20049   3.204  0.00138 **
pH                            1.12805    0.22335   5.051 4.79e-07 ***
DO                           -0.55543    0.19967  -2.782  0.00546 **
I(temperature * DO)           3.20040    0.63498   5.040 5.05e-07 ***
I(temperature * pH)          -2.94243    0.67665  -4.349 1.44e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

s: 1.852 on 2075 degrees of freedom
Multiple R-squared: 0.4832,
Adjusted R-squared: 0.4789
F-statistic: 114.1 on 17 and 2075 DF,  p-value: < 2.2e-16
```

Root mean square of the residuals: 1.96315

Relative error of the means (model response and test set) : -1.604589 %

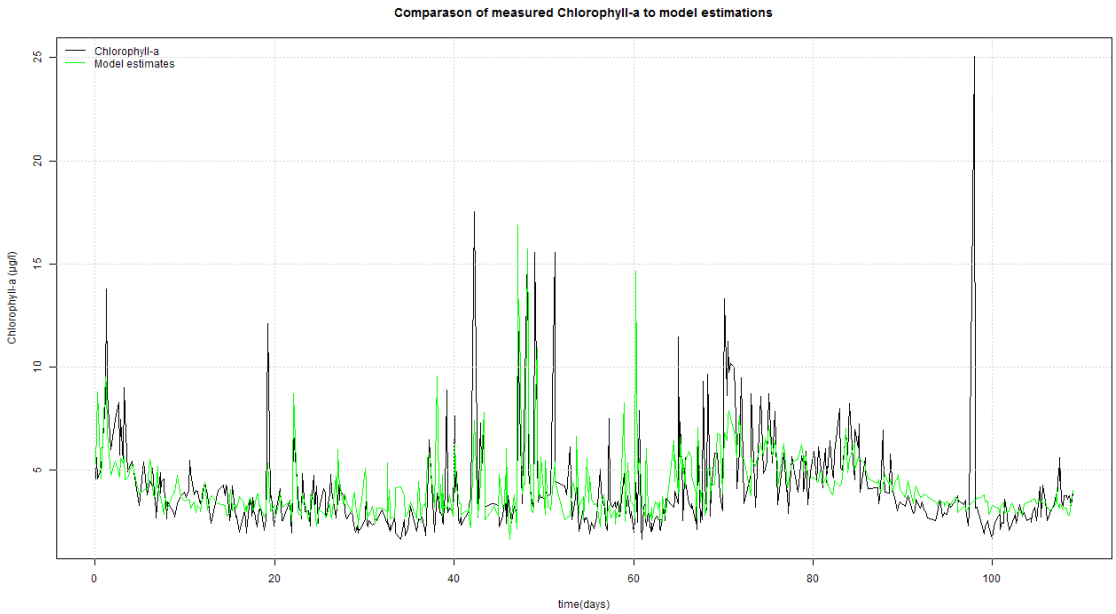

Figure 64. Comparison of measured Chlorophyll-a to model estimations for MLR1, Run #1

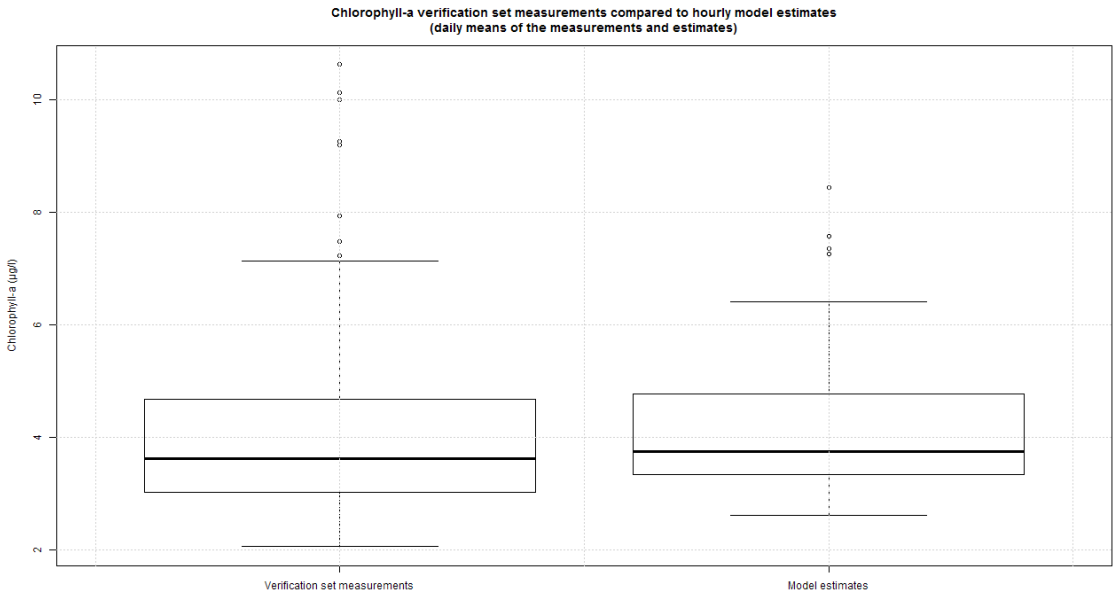

Figure 65. Boxplot of the measured Chlorophyll-a and model estimations for MLR1, Run #1

Results for MLR1 (using standardization defined in Equation 2) – Run #2

Formula:

```
y ~ turbidity + I(conductivity * turbidity) + I(pH^2) + conductivity +
    temperature + I(temperature * turbidity) + I(pH * turbidity) +
    I(conductivity^2) + I(temperature * conductivity) + pH +
    I(temperature^2) + I(DO^2) + I(conductivity * DO) +
    I(temperature * DO) + I(temperature * pH)
```

Model summary:

```
Residuals:
     Min       1Q   Median       3Q      Max
-18.7044  -0.8604  -0.2253   0.5386  24.8751

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                    4.12201    0.15288  26.963  < 2e-16 ***
turbidity                      1.28305    0.07556  16.982  < 2e-16 ***
I(conductivity * turbidity)   -1.06573    0.13261  -8.036 1.54e-15 ***
I(pH^2)                        0.73299    0.09565   7.663 2.76e-14 ***
conductivity                  -1.01048    0.15545  -6.500 1.00e-10 ***
temperature                   -1.13948    0.16545  -6.887 7.51e-12 ***
I(temperature * turbidity)    -1.17804    0.11715 -10.056  < 2e-16 ***
I(pH * turbidity)              0.68805    0.07883   8.728  < 2e-16 ***
I(conductivity^2)              0.82194    0.13049   6.299 3.65e-10 ***
I(temperature * conductivity)  2.36759    0.28715   8.245 2.89e-16 ***
pH                             0.62354    0.14091   4.425 1.01e-05 ***
I(temperature^2)               2.44970    0.34814   7.037 2.67e-12 ***
I(DO^2)                        0.21346    0.08456   2.524   0.0117 *
I(conductivity * DO)           1.03997    0.15123   6.877 8.07e-12 ***
I(temperature * DO)            1.88690    0.26895   7.016 3.08e-12 ***
I(temperature * pH)           -1.34256    0.26918  -4.988 6.62e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

s: 1.978 on 2077 degrees of freedom
Multiple R-squared: 0.4502,
Adjusted R-squared: 0.4462
F-statistic: 113.4 on 15 and 2077 DF,  p-value: < 2.2e-16
```

Root mean square of the residuals: 1.392982

Relative error of the means (model response and test set): -2.935319 %

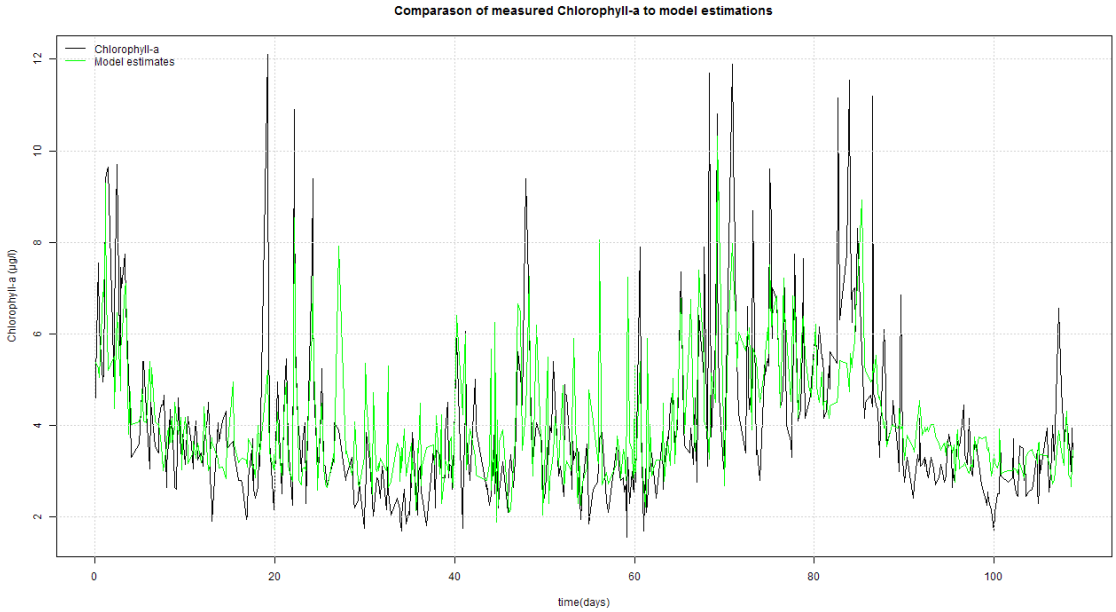

Figure 66. Comparison of measured Chlorophyll-a to model estimations for MLR1, Run #2

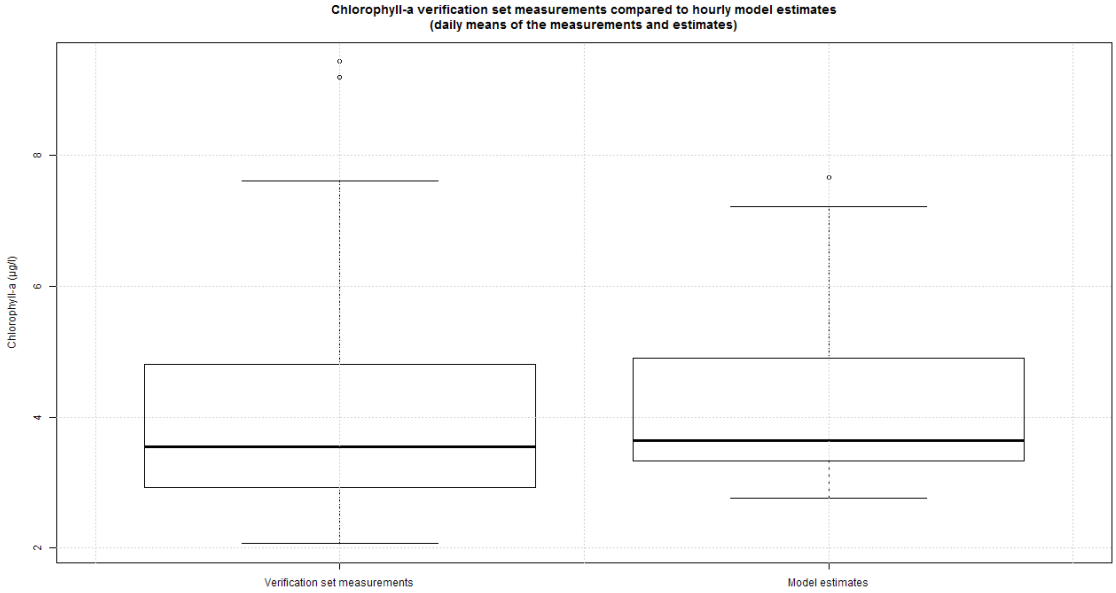

Figure 67. Boxplot of the measured Chlorophyll-a and model estimations for MLR1, Run #2

Results for MLR1 (using standardization defined in Equation 2) – Run #3

Formula:

```
y ~ turbidity + I(conductivity * turbidity) + I(pH^2) + conductivity +
    I(turbidity * DO) + temperature + I(conductivity^2) +
    I(temperature * conductivity) + pH + I(temperature * pH) +
    I(pH * turbidity) + I(temperature * turbidity) +
    I(temperature^2) + I(DO^2) + I(conductivity * DO) +
    I(temperature * DO)
```

Model summary:

```
Residuals:
     Min       1Q   Median       3Q      Max
-19.1414  -0.8467  -0.2289   0.5398  24.9140

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                    4.26166    0.14977  28.455  < 2e-16 ***
turbidity                      1.26866    0.07533  16.841  < 2e-16 ***
I(conductivity * turbidity)   -0.89494    0.13282  -6.738 2.07e-11 ***
I(pH^2)                        0.67805    0.09656   7.022 2.95e-12 ***
conductivity                  -0.86466    0.15309  -5.648 1.85e-08 ***
I(turbidity * DO)             -0.31413    0.15514  -2.025   0.0430 *
temperature                   -1.07119    0.16303  -6.571 6.31e-11 ***
I(conductivity^2)              0.66565    0.12770   5.213 2.05e-07 ***
I(temperature * conductivity)  2.27657    0.28853   7.890 4.84e-15 ***
pH                             0.61430    0.14101   4.356 1.39e-05 ***
I(temperature * pH)           -1.06739    0.26790  -3.984 7.00e-05 ***
I(pH * turbidity)              0.99857    0.12849   7.772 1.21e-14 ***
I(temperature * turbidity)    -1.36365    0.20970  -6.503 9.85e-11 ***
I(temperature^2)               2.15230    0.35789   6.014 2.13e-09 ***
I(DO^2)                        0.21270    0.09247   2.300   0.0215 *
I(conductivity * DO)           1.05032    0.15312   6.859 9.10e-12 ***
I(temperature * DO)            1.70713    0.30169   5.659 1.74e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

s: 1.966 on 2076 degrees of freedom
Multiple R-squared: 0.4411,
Adjusted R-squared: 0.4368
F-statistic: 102.4 on 16 and 2076 DF,  p-value: < 2.2e-16
```

Root mean square of the residuals: 1.460119

Relative error of the means (model response and test set): -2.641434 %

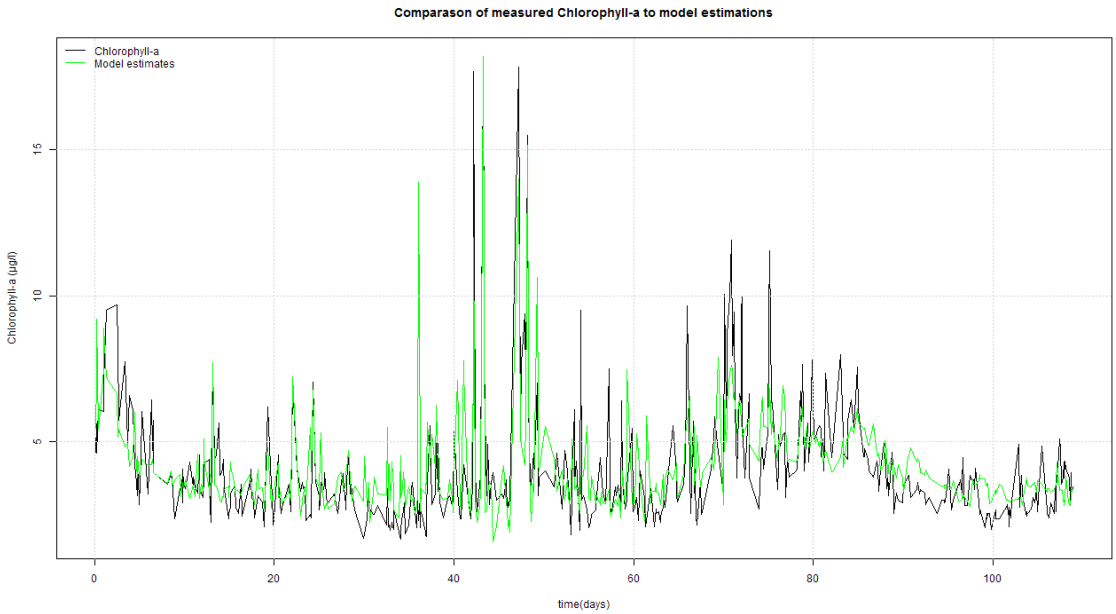

Figure 68. Comparison of measured Chlorophyll-a to model estimations for MLR1, Run #3

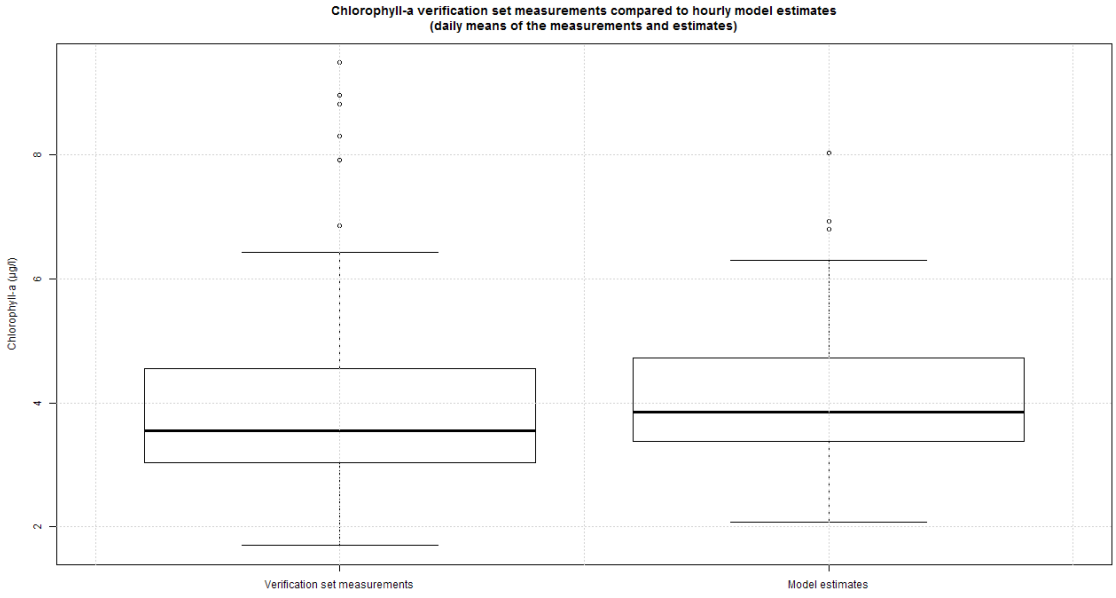

Figure 69. Boxplot of the measured Chlorophyll-a and model estimations for MLR1, Run #3

Results for MLR2 (using standardization defined in Equation 3) – Run #1

Model formula:

```
y ~ turbidity + I(conductivity * turbidity) + I(pH^2) + conductivity +
    temperature + I(pH * DO) + I(conductivity^2) + I(conductivity *
    DO) + I(temperature^2) + I(pH * turbidity) + I(temperature *
    turbidity) + I(temperature * conductivity) + I(DO^2) + pH +
    I(temperature * pH) + I(temperature * DO)
```

Model summary:

```
Residuals:
     Min       1Q   Median       3Q      Max
-18.6179  -0.8121  -0.1966   0.5387  24.9032

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                    3.92620    0.08669  45.288  < 2e-16 ***
turbidity                      1.46482    0.06574  22.281  < 2e-16 ***
I(conductivity * turbidity)   -0.50402    0.06133  -8.218 3.37e-16 ***
I(pH^2)                        1.59587    0.28762   5.549 3.21e-08 ***
conductivity                  -0.36065    0.07880  -4.577 4.96e-06 ***
temperature                   -0.83003    0.10826  -7.667 2.56e-14 ***
I(pH * DO)                    -0.91831    0.45284  -2.028  0.04268 *
I(conductivity^2)              0.19126    0.02727   7.013 3.05e-12 ***
I(conductivity * DO)           0.44715    0.06164   7.254 5.46e-13 ***
I(temperature^2)               2.97984    0.46384   6.424 1.60e-10 ***
I(pH * turbidity)              0.87215    0.08616  10.122  < 2e-16 ***
I(temperature * turbidity)    -1.05474    0.10269 -10.271  < 2e-16 ***
I(temperature * conductivity)  1.15060    0.12488   9.214  < 2e-16 ***
I(DO^2)                        0.41331    0.14669   2.818  0.00488 **
pH                             0.31079    0.13334   2.331  0.01985 *
I(temperature * pH)           -2.82072    0.71970  -3.919 9.14e-05 ***
I(temperature * DO)            2.42218    0.49464   4.897 1.04e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

s: 1.884 on 2337 degrees of freedom
Multiple R-squared: 0.4569,
Adjusted R-squared: 0.4531
F-statistic: 122.9 on 16 and 2337 DF,  p-value: < 2.2e-16
```

Root mean square of the residuals: 1.774387

Relative error of the means (model response and test set): -0.8808585 %

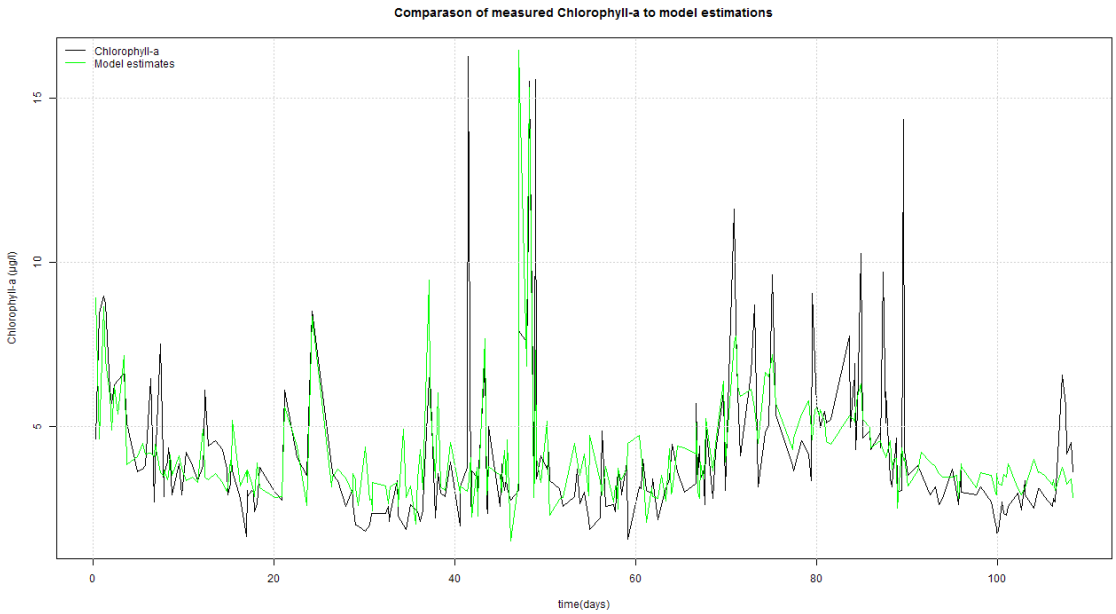

Figure 70. Comparison of measured Chlorophyll-a to model estimations for MLR2, Run #1

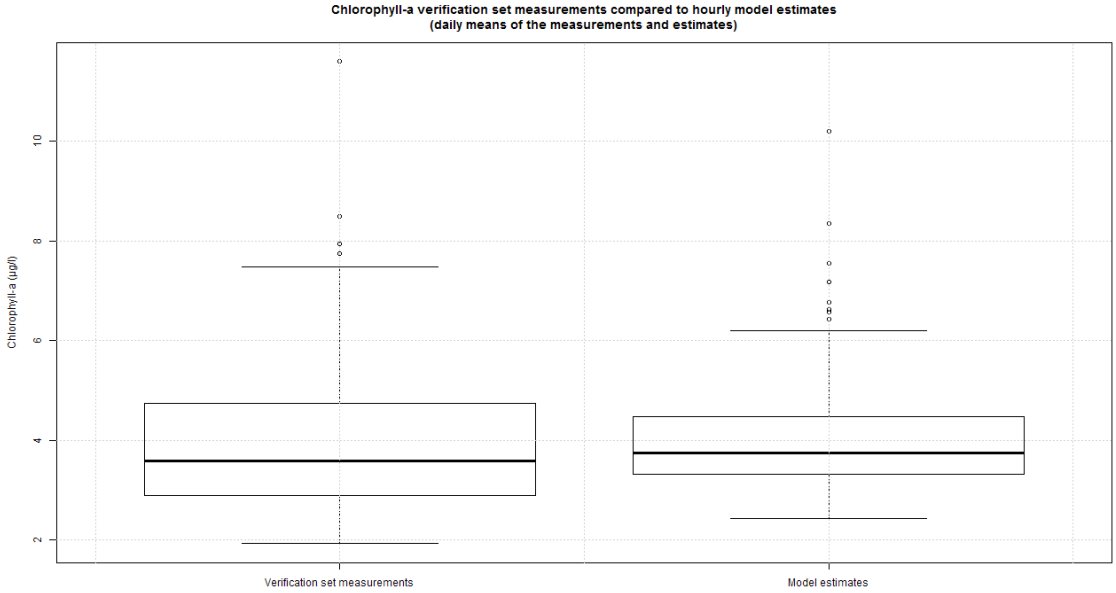

Figure 71. Boxplot of the measured Chlorophyll-a and model estimations for MLR2, Run #1

Results for MLR2 (using standardization defined in Equation 3) – Run #2

Model formula:

```
y ~ turbidity + I(conductivity * turbidity) + I(pH^2) + conductivity +
    I(turbidity * DO) + temperature + I(pH * DO) + I(conductivity^2) +
    I(conductivity * pH) + I(conductivity * DO) + I(temperature^2) +
    I(DO^2) + I(temperature * conductivity) + I(pH * turbidity) +
    I(temperature * turbidity)
```

Model summary:

```
Residuals:
     Min       1Q   Median       3Q      Max
-17.8714  -0.8440  -0.2038   0.5510  24.8179

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                    4.01711    0.08881  45.233  < 2e-16 ***
turbidity                      1.26884    0.07440  17.055  < 2e-16 ***
I(conductivity * turbidity)   -0.46928    0.06469  -7.254 5.47e-13 ***
I(pH^2)                        0.90312    0.10748   8.402  < 2e-16 ***
conductivity                  -0.51706    0.07393  -6.994 3.48e-12 ***
I(turbidity * DO)             -0.44693    0.13235  -3.377 0.000745 ***
temperature                   -0.77215    0.08359  -9.237  < 2e-16 ***
I(pH * DO)                     0.65991    0.15708   4.201 2.76e-05 ***
I(conductivity^2)              0.19445    0.02751   7.068 2.07e-12 ***
I(conductivity * pH)           0.38797    0.15141   2.562 0.010460 *
I(conductivity * DO)           0.23425    0.07934   2.952 0.003185 **
I(temperature^2)               0.78959    0.17128   4.610 4.24e-06 ***
I(DO^2)                       -0.19981    0.05584  -3.578 0.000353 ***
I(temperature * conductivity)  0.71563    0.15407   4.645 3.59e-06 ***
I(pH * turbidity)              1.25242    0.14072   8.900  < 2e-16 ***
I(temperature * turbidity)    -1.57205    0.18727  -8.395  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

s: 1.906 on 2338 degrees of freedom
Multiple R-squared: 0.4313,
Adjusted R-squared: 0.4276
F-statistic: 118.2 on 15 and 2338 DF,  p-value: < 2.2e-16
```

Root mean square of the residuals: 1.611111

Relative error of the means (model response and test set): -1.018662 %

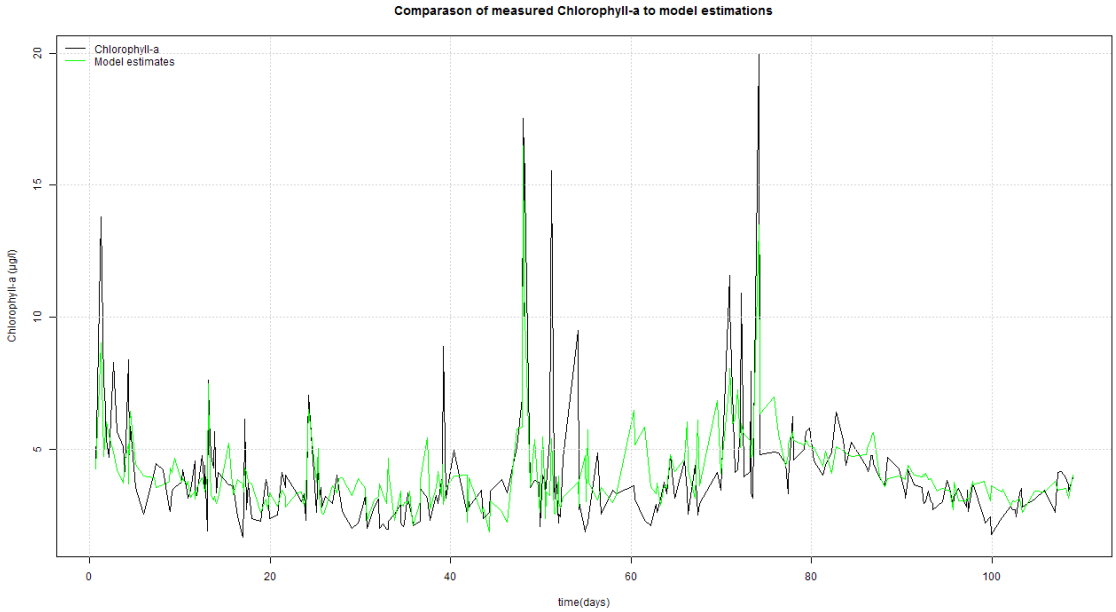

Figure 72. Comparison of measured Chlorophyll-a to model estimations for MLR2, Run #2

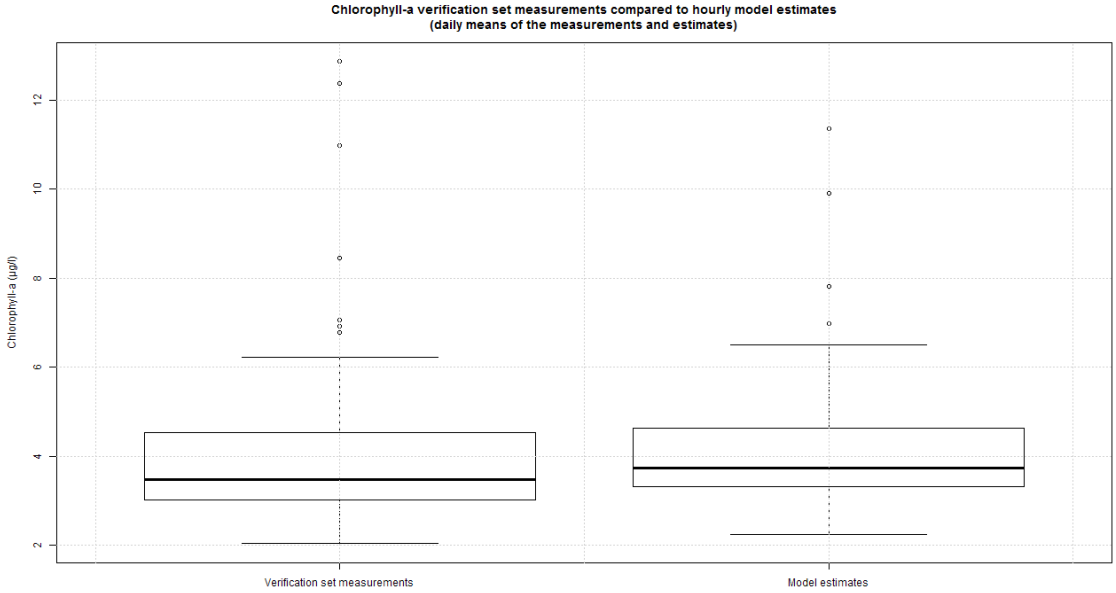

Figure 73. Boxplot of the measured Chlorophyll-a and model estimations for MLR2, Run #2

Results for MLR2 (using standardization defined in Equation 3) – Run #3

Model formula:

```
y ~ turbidity + I(conductivity * turbidity) + I(pH^2) + conductivity +
    I(turbidity * DO) + temperature + I(pH * DO) + I(conductivity^2) +
    I(conductivity * pH) + I(conductivity * DO) + I(temperature *
    turbidity) + I(pH * turbidity) + I(temperature * conductivity) +
    I(temperature^2) + I(DO^2)
```

Model summary:

```
Residuals:
     Min       1Q   Median       3Q      Max
-17.9141  -0.8414  -0.2015   0.5653  24.9150

Coefficients:
                             Estimate Std. Error t value Pr(>|t|)
(Intercept)                   4.03505    0.08603  46.904  < 2e-16 ***
turbidity                     1.37088    0.07204  19.030  < 2e-16 ***
I(conductivity * turbidity)  -0.51498    0.06508  -7.912 3.86e-15 ***
I(pH^2)                       0.85539    0.10708   7.988 2.13e-15 ***
conductivity                 -0.56863    0.07233  -7.861 5.76e-15 ***
I(turbidity * DO)            -0.24163    0.12608  -1.916 0.055427 .
temperature                  -0.72716    0.08121  -8.954  < 2e-16 ***
I(pH * DO)                    0.64303    0.14958   4.299 1.79e-05 ***
I(conductivity^2)             0.19578    0.02804   6.981 3.80e-12 ***
I(conductivity * pH)          0.47730    0.14908   3.202 0.001385 **
I(conductivity * DO)          0.15898    0.07679   2.070 0.038538 *
I(temperature * turbidity)   -1.38492    0.18248  -7.589 4.61e-14 ***
I(pH * turbidity)             1.01632    0.13611   7.467 1.15e-13 ***
I(temperature * conductivity) 0.54919    0.15041   3.651 0.000267 ***
I(temperature^2)              0.70800    0.16637   4.255 2.17e-05 ***
I(DO^2)                      -0.18850    0.05408  -3.485 0.000501 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

s: 1.842 on 2338 degrees of freedom
Multiple R-squared: 0.4425,
Adjusted R-squared: 0.4389
F-statistic: 123.7 on 15 and 2338 DF,  p-value: < 2.2e-16
```

Root mean square of the residuals: 2.178123

Relative error of the means (model response and test set): 2.117905 %

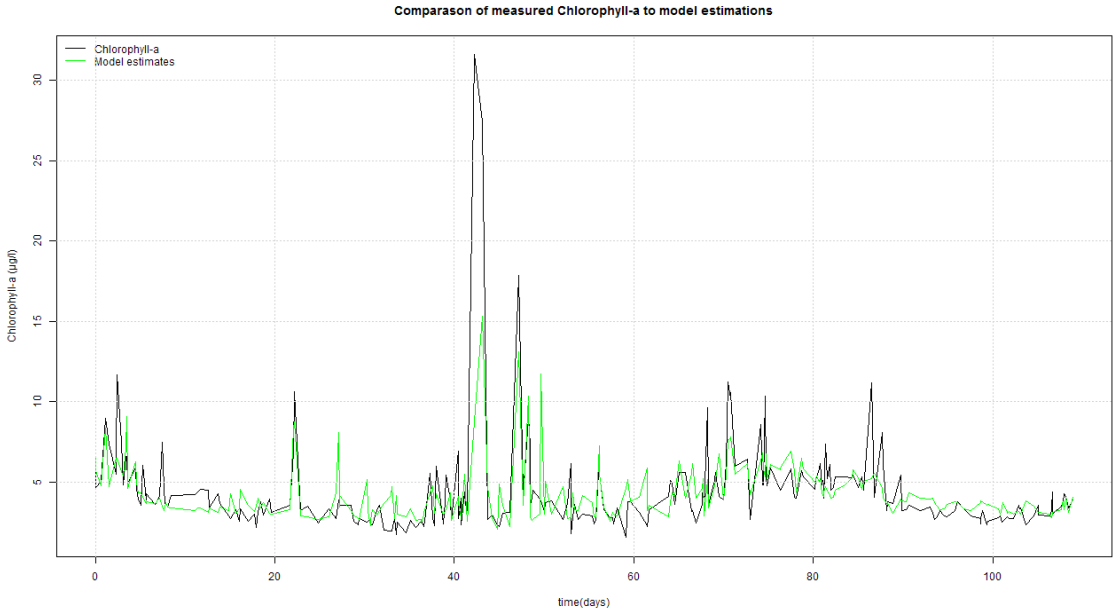

Figure 74. Comparison of measured Chlorophyll-a to model estimations for MLR2, Run #3

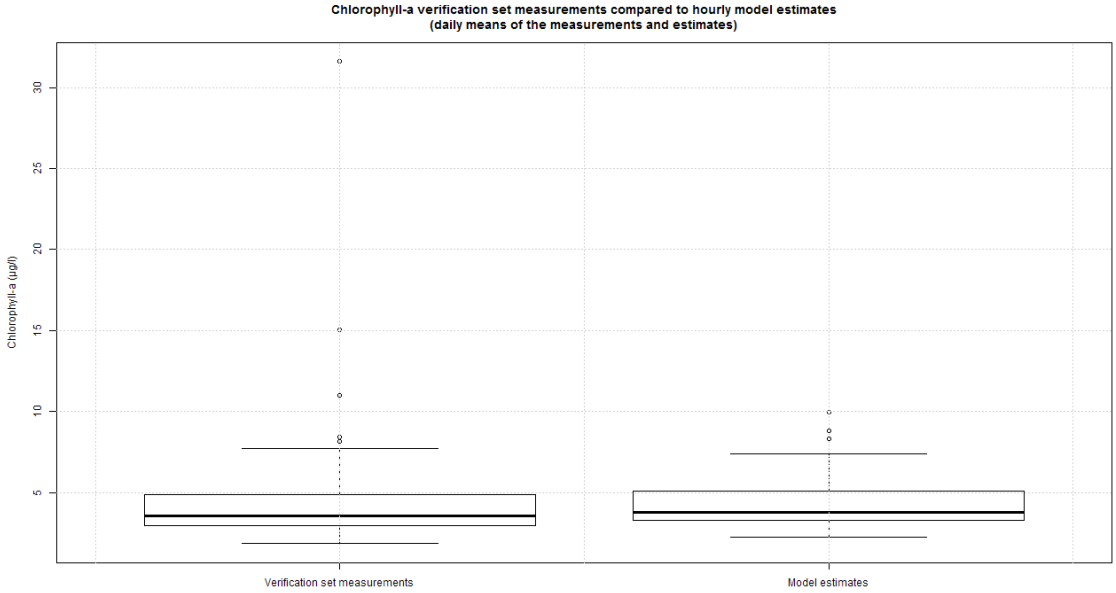

Figure 75. Boxplot of the measured Chlorophyll-a and model estimations for MLR2, Run #3

Principal Component Analysis

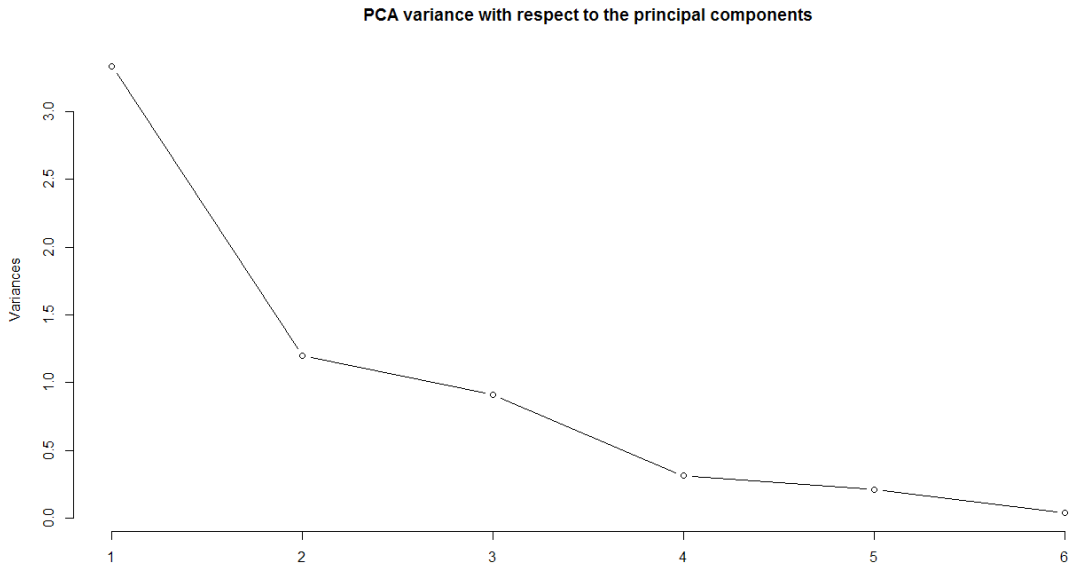

Figure 76. PCA variance with respect to the principal components

Summary of the PCA:

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6
Standard deviation     1.8249 1.0948 0.9536 0.55957 0.45918 0.19402
Proportion of Variance 0.5551 0.1998 0.1516 0.05219 0.03514 0.00627
Cumulative Proportion  0.5551 0.7548 0.9064 0.95858 0.99373 1.00000
```

Rotations of the PCA:

```
                 PC1    PC2    PC3    PC4    PC5    PC6
temperature   -0.524  0.073  0.187 -0.208  0.237  0.765
conductivity   0.491 -0.167  0.118  0.012  0.844  0.065
pH            -0.463  0.101 -0.427 -0.467  0.392 -0.471
turbidity      0.394  0.435  0.303 -0.730 -0.176  0.009
DO             0.339 -0.053 -0.810 -0.149 -0.126  0.434
a.chlorophyll  0.016  0.874 -0.144  0.428  0.175  0.024
```

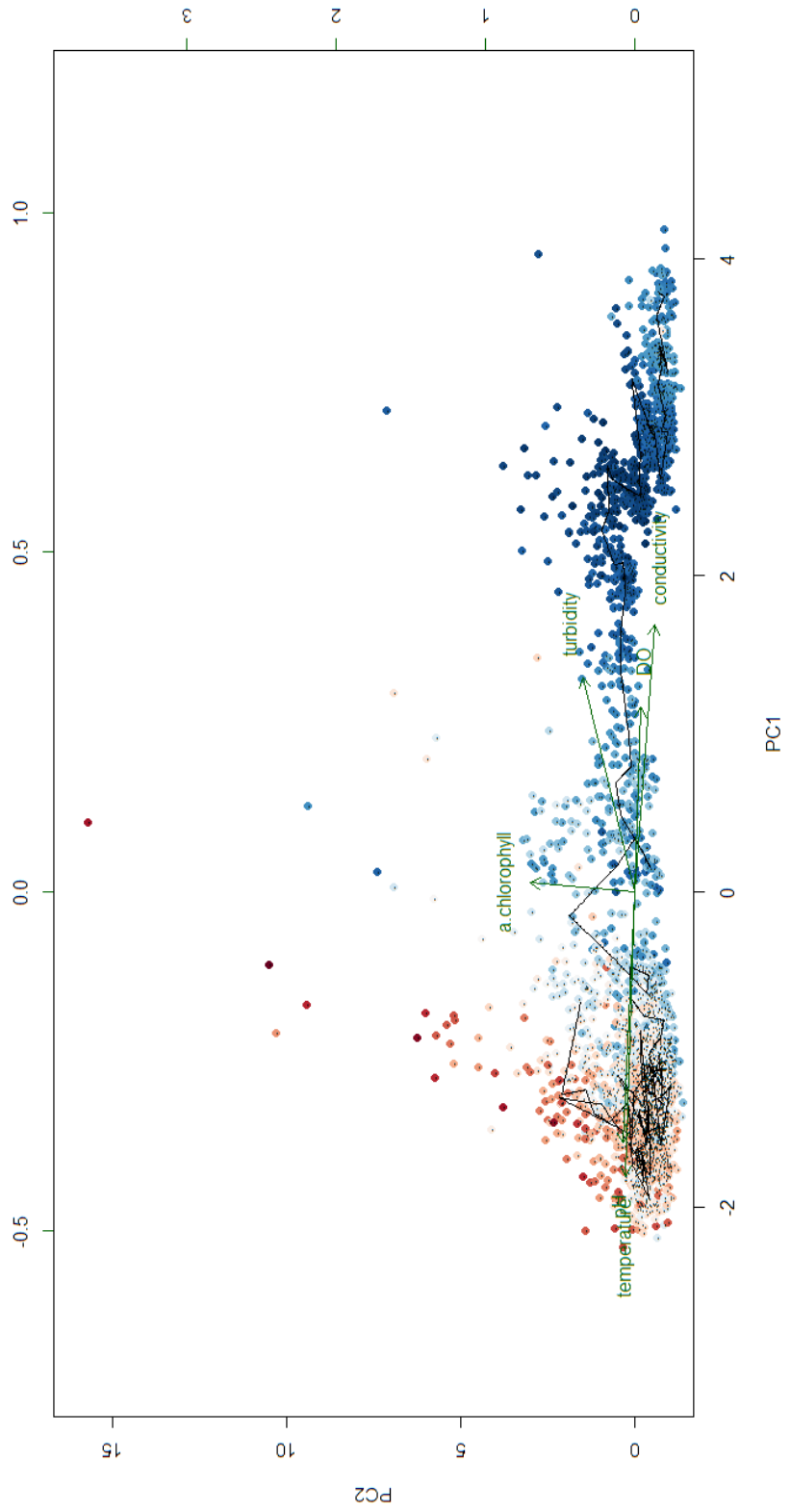

Figure 77. PCA biplot of PC1 and PC2, hot temperatures are red and cold ones are blue, black line represents time

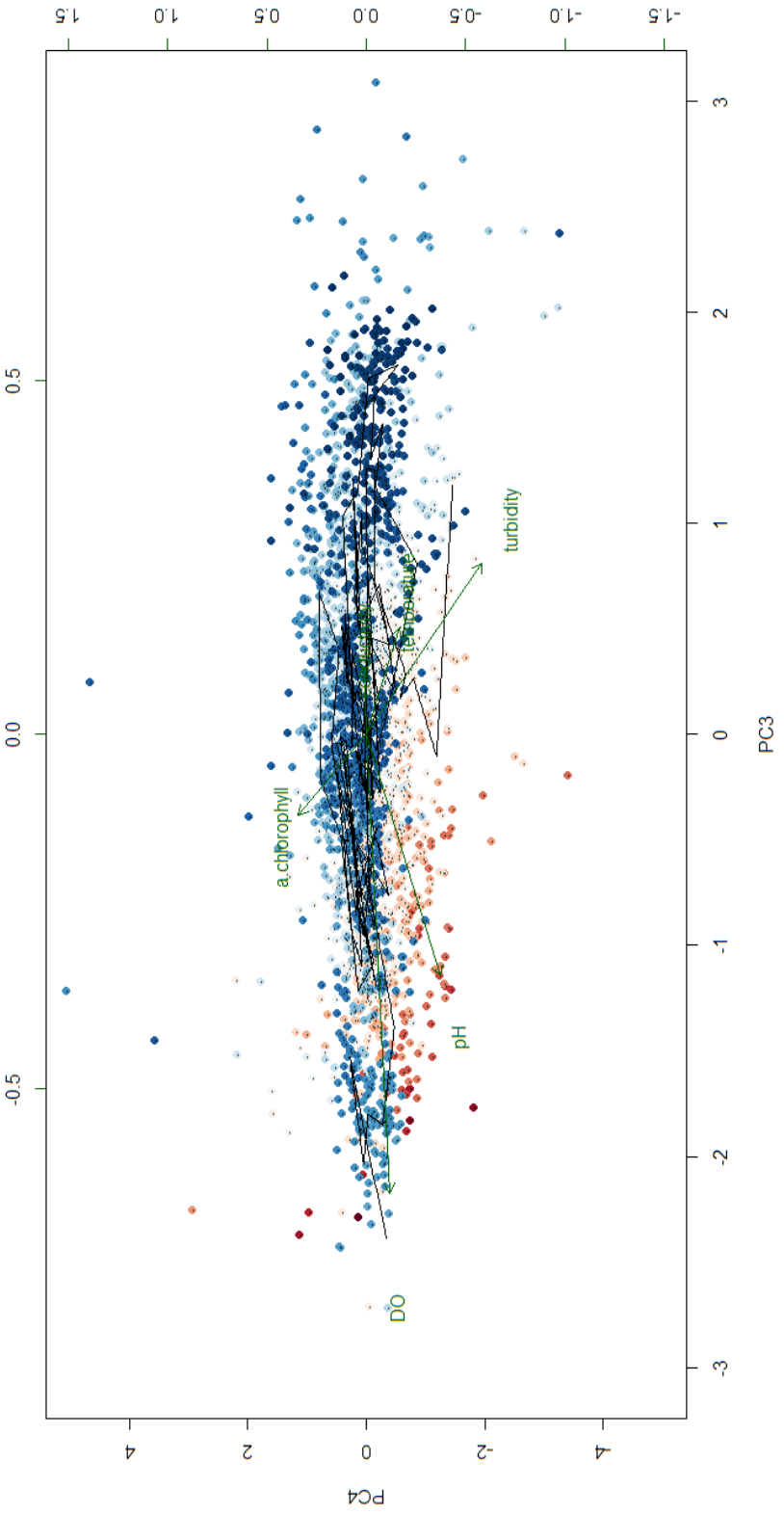

Figure 78. PCA biplot of PC3 and PC4, hot temperatures are red and cold ones are blue, black line represents time

Source code (Lake quality assessment tool)

```
library(ggplot2)

#load data from CSV file
myData <- read.csv2("data_en.CSV")

#remove useless column
myData <- myData[,c(1:9)]

#add ellapsed time to the data
times <- strptime(myData$Date.and.time, format = "%d.%m.%Y %H:%M", tz=
"EET") #create POSIX time
initialTime <- as.numeric(times[4]) #first valid record
times <- (as.numeric(times) - initialTime) #set the time = 0
#times <- times/(24*60*60) #convert into days
times <- times/(24)/150 #convert into hours
myData$ellapsedTime <- times #add to the dataframe
myData <- myData[myData$ellapsedTime>=0,] #keep only valid samples
myData <- myData[complete.cases(myData),] #remove any row with at leas
t one NA

rm(initialTime)
rm(times)


names(myData)<-c("Date.and.time" , "temperature", "conductivity", "pH"
, "turbidity", "DO", "a.chlorophyll", "cyanobacteria", "battery", "ell
apsedTime")

colors <- c("darkblue", "lightskyblue", "seagreen2", "orange", "red")
thresholds <- c(4,10,20,50, Inf)
condition <- c("Excellent", "Good", "Satisfactory", "Passable", "Poor"
)

getAverage <- function(x,t){
  y <- data.frame(temperature=NA, conductivity=NA, pH=NA, turbidity=NA
, DO=NA, a.chlorophyll=NA, cyanobacteria=NA, ellapsedTime=NA, dt =NA)[
numeric(0), ]
  for(i in 0:max(floor(x$ellapsedTime/t))){
    values <- which(floor(x$ellapsedTime/t) >= i & floor(x$ellapsedTim
e/t) < i+1)
    for( j in 1:7){
      y[i+1,j] <- mean(x[values,j+1])
      y[i+1,j+1] <- i
      y[i+1,j+2] <- max(x$ellapsedTime[values])-min(x$ellapsedTime[val
ues])
    }
  }
  return(y)
}

getCondition <- function (x){
  temp <- character()
  for(i in 1:length(x$a.chlorophyll)){
    temp[i] <- colors[min(which(x$a.chlorophyll[i] < thresholds))]
  }
  x$color <- temp
```

```
  for(i in 1:length(x$a.chlorophyll)){
    temp[i] <- condition[min(which(x$a.chlorophyll[i] < thresholds))]
  }
  x$condition <- temp
  return(x)
}

getStats <- function(x, w = TRUE){

  temp <- numeric()
  weight <- numeric()
  for(i in 1:length(colors)){
    temp[i] <- length(which(x$color==colors[i]))
    weight[i]<-sum(x$dt[which(x$color==colors[i])])
  }
  ifelse(w,weight,1)
 if(w)  temp <- data.frame(time.percentage = round(100*(temp*weight)/s
um((temp*weight)),3), color = colors, condition = condition)
          else  temp <- data.frame(time.percentage = round(100*temp/su
m(temp),3), color = colors, condition = condition)
  temp$label <- paste(temp$time.percentage, "%", sep="")
  temp$label.color <- ifelse(temp$time.percentage>0,"black","black")
  temp$label.vjust <- ifelse(temp$time.percentage>0,-0.2,0)
  temp$index <- 1:5
  temp$weight <- weight
  return(temp)
}

makePlot <- function(x1, x2, dp=''){
  dp <- ifelse(dp=='','',paste(" from ", dp, sep=''))
  my.Subtitle <- paste("Chlorophyll-a mean: ", round(mean(x1$a.chloro
hyll), 2), " ; ",length(x1$a.chlorophyll), " data points", dp, sep='')
  my.Title <- paste("Trophic levels for Lake Gennarbyträsket for the s
ampling season (", myData$Date.and.time[which.min(myData$ellapsedTime)
],
                    " - ", myData$Date.and.time[which.max(myData$ellap
sedTime)], ")", sep = '')


  p <- ggplot(x2, aes(x=index, y=time.percentage, fill=color)) +
    geom_bar(stat="identity", fill=colors)+
    geom_text(aes(label=label), vjust=x2$label.vjust, color=x2$label.c
olor, size=3.5)+
    labs(
      title=my.Title,
      subtitle=my.Subtitle,
      x = "Trophic condition",
      y="Time (%) spent in the trophic condition")+
    scale_x_discrete(limits=x2$condition)

  print(p)
}

monthly.Average <- getCondition(getAverage(myData, 730))

daily.Average <- getCondition(getAverage(myData, 24))

daily.Levels<-getStats(daily.Average, F)
```

```
monthly.Levels<-getStats(monthly.Average)

myData <- getCondition(myData)
myData$dt <- rep(1, each=length(myData[[1]]))
data.levels <- getStats(myData, F)

makePlot(myData, data.levels, "raw data")
#makePlot(monthly.Average, monthly.Levels, "monthly averages")
#makePlot(daily.Average,daily.Levels, "daily averages")

myData$condition <- factor(myData$condition, condition)

myBox<-boxplot(myData$a.chlorophyll ~ myData$condition, main="Do not u
se this graph")
text(1:5, (1:5)*10, rep("not for use",25), col = "red")

cond.n <- character()
for(i in 1:length(condition)){
  cond.n[i] <- paste("(", myBox$n[i], " observations)", sep='')
}


boxplot(myData$a.chlorophyll ~ myData$condition, main="Boxplots of the
chlorophyll-a distribution according to the lake's trophic level")

text(1:5, c(7,13,7,17, 50),cond.n)

plot(myData$ellapsedTime/24, myData$a.chlorophyll,
     col="black",
     xlab = "Time (days)", ylab="Chlorophyll-a (µg/l)",
     main="Chlorophyll-a with respect to time",
     type = 'l')

plot(myData$ellapsedTime/24, myData$a.chlorophyll,
     col=myData$color,
     xlab = "Time (days)", ylab="Chlorophyll-a (µg/l)",
     main="Chlorophyll-a with respect to time color coded with SYKE's
water quality levels",
     sub="Dark blue = excellent, blue = good, green = satisfactory, or
ange = passable, red = poor",
     pch=19,lwd=2)
```

Source code (MLR1 and smoothing)

```
source("http://users.metropolia.fi/~velimt/R/DOE_functions_v5.r")
if (!require("RColorBrewer")) {
  install.packages("RColorBrewer")
  library(RColorBrewer)
}

rms <- function(x) sqrt(mean(x^2))

#load data from CSV file
myData <- read.csv2("data_en.CSV")

#remove useless column
myData <- myData[,c(1:9)]

#add ellapsed time to the data
times <- strptime(myData$Date.and.time, format = "%d.%m.%Y %H:%M", tz=
"EET") #create POSIX time
initialTime <- as.numeric(times[4]) #first valid record
times <- (as.numeric(times) - initialTime) #set the time = 0
#times <- times/(24*60*60) #convert into days
times <- times/(24)/150 #convert into hours
myData$ellapsedTime <- times #add to the dataframe
myData <- myData[myData$ellapsedTime>=0,] #keep only valid samples
myData <- myData[complete.cases(myData),] #remove any row with at leas
t one NA

create_average <- function(DF, column_name_value, column_name_time, a,
b, column_name = 'mean.value'){
  mean.value <- numeric()
  time_mean <- numeric()
  for(i in a:b){
    time_mean <- append(time_mean, i)
    mean.value <- append(
      mean.value,
      mean(
        DF[[column_name_value]][
          DF[[column_name_time]]<=i&
            DF[[column_name_time]]>(i-1)
          ]
      )
    )
  }
  DF <- data.frame(time_mean=time_mean ,mean.value=mean.value)
  colnames(DF)<-c('time_mean', column_name)
  return(DF)
}

my.Data.Av <- create_average(myData,'Lämpötila...C.' , 'ellapsedTime',
0, 2615, 'temperature')
my.Data.Av <- merge(
  my.Data.Av,
  create_average(myData,'Johtokyky..µS.cm.' , 'ellapsedTime', 0, 2615,
'conductivity'),
  by.x = 'time_mean', by.y='time_mean'
)
```

```
my.Data.Av <- merge(
  my.Data.Av,
  create_average(myData,'pH....' , 'ellapsedTime', 0, 2615, 'pH'),
  by.x = 'time_mean', by.y='time_mean'
)

my.Data.Av <- merge(
  my.Data.Av,
  create_average(myData,'Sameus..FTU.' , 'ellapsedTime', 0, 2615, 'tur
bidity'),
  by.x = 'time_mean', by.y='time_mean'
)

my.Data.Av <- merge(
  my.Data.Av,
  create_average(myData,'Happi..mg.l.' , 'ellapsedTime', 0, 2615, 'DO'
),
  by.x = 'time_mean', by.y='time_mean'
)

my.Data.Av <- merge(
  my.Data.Av,
  create_average(myData,'A.klorofylli..µg.l.' , 'ellapsedTime', 0, 261
5, 'a.chlorophyll'),
  by.x = 'time_mean', by.y='time_mean'
)

my.Data.Av <- merge(
  my.Data.Av,
  create_average(myData,'Sinilevien.osuus..µg.l.' , 'ellapsedTime', 0,
2615, 'cyanobacteria'),
  by.x = 'time_mean', by.y='time_mean'
)



my.Data.Av.Scaled <- my.Data.Av
for(i in 2:6)my.Data.Av.Scaled[[i]] <- scale(my.Data.Av.Scaled[[i]],T,
T)

maxX <- max(my.Data.Av.Scaled[[2]])
minX <- min(my.Data.Av.Scaled[[2]])
for(i in 3:6){
  maxX<-max(maxX, my.Data.Av.Scaled[[i]])
  minX<-min(minX, my.Data.Av.Scaled[[i]])
}


colors <- brewer.pal(7, "Dark2")

plot(my.Data.Av.Scaled$time_mean, my.Data.Av.Scaled$temperature, type=
'l', xlab='time (hours)', ylab = 'Scaled values', ylim=c(floor(minX),
ceiling(maxX)), col=colors[1])


for(i in 3:6){
  lines(my.Data.Av.Scaled$time_mean, my.Data.Av.Scaled[[i]], col=color
s[i])
}
```

```
my.Data.Av.Scaled.Smoothed <- my.Data.Av.Scaled

chlo.smooth.model <- loess(a.chlorophyll~time_mean, data=myDF<-data.fr
ame(time_mean=my.Data.Av.Scaled$time_mean,a.chlorophyll=my.Data.Av.Sca
led[[7]]),
                           span=.005)
myDF$pred <- predict(chlo.smooth.model, data=myDF)

my.Data.Av.Scaled.Smoothed$a.chlorophyll <- predict(chlo.smooth.model,
data=myDF)

plot(myDF$time_mean,myDF$a.chlorophyll, col='green', type='b', xlab="t
ime(days)", ylab="Chlorophyll-a(µg/l)", main = "Smoothing of Chlorophy
ll-a")
lines(myDF$time_mean,myDF$pred)
legend(2000,45, # places a legend at the appropriate place
       c("Chlorophyll-a","Smoothed line"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',

       #lwd=c(2.5,2.5),
       col=c("green","black")) # gives the legend lines the correct co
lor and width

plot(myDF$time_mean/24,myDF$a.chlorophyll, col='green', type='b',xlim=
c(1000,1200)/24, xlab="time(days)", ylab="Chlorophyll-a(µg/l)", main =
"Smoothing of Chlorophyll-a (detailed view)")
lines(myDF$time_mean/24,myDF$pred)
legend(48,45, # places a legend at the appropriate place
       c("Chlorophyll-a","Smoothed line"), # puts text in the legend
       cex = 0.8,
       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',

       #lwd=c(2.5,2.5),
       col=c("green","black")) # gives the legend lines the correct co
lor and width

plot(myDF$time_mean/24,myDF$a.chlorophyll, col='green', type='b',xlim=
c(1500,1850)/24, xlab="time(days)", ylab="Chlorophyll-a(µg/l)", main =
"Smoothing of Chlorophyll-a (detailed view)")
lines(myDF$time_mean/24,myDF$pred)
legend(62,45, # places a legend at the appropriate place
       c("Chlorophyll-a","Smoothed line"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',

       #lwd=c(2.5,2.5),
       col=c("green","black")) # gives the legend lines the correct co
lor and width
```

```
plot(myDF$time_mean,scale(myDF$a.chlorophyll,T,T), col='green', type='
b', xlab="time(h)", ylab="Scaled units")
lines(myDF$time_mean,scale(myDF$pred,T,T))
lines(my.Data.Av.Scaled$time_mean, my.Data.Av.Scaled$turbidity, col='r
ed')

plot(myDF$time_mean,scale(myDF$a.chlorophyll,T,T), col='green', type='
b', xlab="time(h)", ylab="Scaled units",xlim=c(1000,1200))
lines(myDF$time_mean,scale(myDF$pred,T,T))
lines(my.Data.Av.Scaled$time_mean, my.Data.Av.Scaled$turbidity, col='r
ed')


turb.smooth.model <- loess(turbidity~time_mean, data=myDF.turbidity<-d
ata.frame(time_mean=my.Data.Av.Scaled$time_mean,turbidity=my.Data.Av.S
caled$turbidity),
                           span=.005)
myDF.turbidity$pred <- predict(turb.smooth.model, data=myDF.turbidity)
my.Data.Av.Scaled.Smoothed$turbidity <- predict(turb.smooth.model, dat
a=myDF)


plot(myDF.turbidity$time_mean,myDF.turbidity$turbidity,type='l')
lines(myDF.turbidity$time_mean,myDF.turbidity$pred,col='red')

plot(myDF.turbidity$time_mean,myDF.turbidity$turbidity,type='l',xlim=c
(1000,1200))
lines(myDF.turbidity$time_mean,myDF.turbidity$pred,col='red')


conductivity.smooth.model <- loess(conductivity~time_mean,
                                   data=myDF.conductivity<-data.frame(
time_mean=my.Data.Av.Scaled$time_mean,conductivity=my.Data.Av.Scaled$c
onductivity),
                                   span=.03)

my.Data.Av.Scaled.Smoothed$conductivity <- predict(conductivity.smooth
.model, data=my.Data.Av.Scaled)

plot(my.Data.Av.Scaled$time_mean, my.Data.Av.Scaled$conductivity,type
= 'l', xlab="time(h)", ylab="Conductivity (scaled units)")
lines(my.Data.Av.Scaled.Smoothed$time_mean, my.Data.Av.Scaled.Smoothed
$conductivity, col='blue')
legend(2000,1.5, # places a legend at the appropriate place
       c("Conductivity","Smoothed line"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',

       #lwd=c(2.5,2.5),
       col=c("black","blue")) # gives the legend lines the correct col
or and width

plot(my.Data.Av.Scaled$time_mean, my.Data.Av.Scaled$conductivity,type
= 'l', xlab="time(h)", ylab="Conductivity (scaled units)", xlim=c(600,
700))
lines(my.Data.Av.Scaled.Smoothed$time_mean, my.Data.Av.Scaled.Smoothed
$conductivity, col='blue')
```

```
legend(675,4, # places a legend at the appropriate place
       c("Conductivity","Smoothed line"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)
       cex = .75,
       bty = 'n',

       #lwd=c(2.5,2.5),
       col=c("black","blue")) # gives the legend lines the correct col
or and width

pH.smooth.model <- loess(pH~time_mean,
                         data=myDF.pH<-data.frame(time_mean=my.Data.Av
.Scaled$time_mean,pH=my.Data.Av.Scaled$pH),
                         span=.01)

my.Data.Av.Scaled.Smoothed$pH <- predict(pH.smooth.model, data=my.Data
.Av.Scaled)

plot(my.Data.Av.Scaled$time_mean, my.Data.Av.Scaled$pH,type = 'l', xla
b="time(h)", ylab="pH (scaled units)")
lines(my.Data.Av.Scaled.Smoothed$time_mean, my.Data.Av.Scaled.Smoothed
$pH, col='green')
legend(2000,1.5, # places a legend at the appropriate place
       c("pH","Smoothed line"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',

       #lwd=c(2.5,2.5),
       col=c("black","green")) # gives the legend lines the correct co
lor and width


plot(my.Data.Av.Scaled$time_mean/24, my.Data.Av.Scaled$pH,type = 'l',
xlab="time(h)", ylab="pH (scaled units)", xlim=c(500,688)/24)
lines(my.Data.Av.Scaled.Smoothed$time_mean/24, my.Data.Av.Scaled.Smoot
hed$pH, col='green')


DO.smooth.model <- loess(DO~time_mean,
                         data=myDF.DO<-data.frame(time_mean=my.Data.Av
.Scaled$time_mean,DO=my.Data.Av.Scaled$DO),
                         span=.04)

my.Data.Av.Scaled.Smoothed$DO <- predict(DO.smooth.model, data=my.Data
.Av.Scaled)

plot(my.Data.Av.Scaled$time_mean, my.Data.Av.Scaled$DO,type = 'l', xla
b="time(h)", ylab="Dissolved Oxygen (scaled units)")
lines(my.Data.Av.Scaled.Smoothed$time_mean, my.Data.Av.Scaled.Smoothed
$DO, col='blue')
legend(00,4, # places a legend at the appropriate place
       c("Dissolved Oxygen","Smoothed line"), # puts text in the legen
d

       lty=c(1,1), # gives the legend appropriate symbols (lines)
```

```
       bty = 'n',

       #lwd=c(2.5,2.5),
       col=c("black","blue")) # gives the legend lines the correct col
or and width


temperature.smooth.model <- loess(temperature~time_mean,
                                  data=myDF.temperature<-data.frame(ti
me_mean=my.Data.Av.Scaled$time_mean,temperature=my.Data.Av.Scaled$temp
erature),
                                  span=.03)

my.Data.Av.Scaled.Smoothed$temperature <- predict(temperature.smooth.m
odel, data=my.Data.Av.Scaled)

plot(my.Data.Av.Scaled$time_mean, my.Data.Av.Scaled$temperature,type =
'l', xlab="time(h)", ylab="Temperature (scaled units)")
lines(my.Data.Av.Scaled.Smoothed$time_mean, my.Data.Av.Scaled.Smoothed
$temperature, col='red')
legend(2000,1.5, # places a legend at the appropriate place
       c("Temperature","Smoothed line"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',

       #lwd=c(2.5,2.5),
       col=c("black","red")) # gives the legend lines the correct colo
r and width


plot(myDF$time_mean/24,scale(myDF$pred,T,T), col='green', type='l', xl
ab="time(days)", ylab="Scaled units",
     main="Turbidity and Chlorophyll-a (smoothed and standardized)")
lines(myDF.turbidity$time_mean/24, myDF.turbidity$pred, col='brown')
legend("topright", # places a legend at the appropriate place
       c("Chlorophyll-a","Turbidity"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',

       #lwd=c(2.5,2.5),
       col=c("green","brown")) # gives the legend lines the correct co
lor and width


plot(myDF$time_mean/24,myDF$a.chlorophyll, col='green',xlim=c(1500,185
0)/24, xlab="time(h)", ylab="Chlorophyll-a(µg/l)")


temp <- sample(nrow(my.Data.Av.Scaled),round(nrow(my.Data.Av.Scaled)*.
2)) #take 10% of the data to check the model and 90% to train it
check <- my.Data.Av.Scaled[temp,]
check <- check[order(check$time_mean),]

train <- my.Data.Av.Scaled[-temp,]
train <- train[order(train$time_mean),]
```

```
rm(temp)

X=train[,2:6]
Y=train[,7]

models<-quad.CV.fwd(X,Y,20,2)
which.max(models$Q2)
fmla <- models$formulas[[which.max(models$Q2)]]
print(summary(model1<-lm(fmla,data=train)))
plot(check$time_mean, check$a.chlorophyll, type='l')
lines(check$time_mean, predict(model1, newdata = check), col='green')

continue = TRUE
model2 <- model1
fmla2 <- fmla

while(continue){
  least.significant <- names(summary(model2)$coefficients[,4])[which.m
ax(summary(model2)$coefficients[,4])]
  least.significant.p.value <- summary(model2)$coefficients[,4][which.
max(summary(model2)$coefficients[,4])]
  cat(paste("The least significant therm is: ", least.significant, " p
-value: ", least.significant.p.value))
  ans <- readline('Do you want to remove the least significant therm (
answer y or n): ')
  if(tolower(ans)=='y') continue = TRUE else{
    continue = FALSE
    break
  }
  print(continue)
  fmla2 <- update(fmla2, paste(".~.-", least.significant))
  print(summary(model2<-lm(fmla2,data=train)))
}
print(fmla2)
print("RSM: ")
print(myRMS <- rms(predict(model2, newdata = check)-check$a.chlorophyl
l))



y.predict <- predict(model2, newdata = check)
cat("rel error of the means ", (1-(mean(y.predict)/mean(check$a.chloro
phyll)))*100, "%")


run.nb <- readline('What is the run number?: ')
png(file=paste("MLR1_Run_", run.nb, "_Chl_vs_model.png", sep=''), bg="
white", width = 1280, height = 720)

plot(check$time_mean/24,check$a.chlorophyll, type='l',
     xlab="time(days)",
     ylab="Chlorophyll-a (µg/l)",
     main="Comparason of measured Chlorophyll-a to model estimations"
)
lines(check$time_mean/24, y.predict,col='green')
legend("topleft", # places a legend at the appropriate place
       c("Chlorophyll-a","Model estimates"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)
```

```
       bty = 'n',
       #lwd=c(2.5,2.5),
       col=c("black","green"))
grid()
dev.off()


myDF <- check
myDF$predict <- y.predict
myDF$time_mean <- myDF$time_mean/24


daily.predict <- merge(
  create_average(myDF, "a.chlorophyll", "time_mean", 1,floor(max(myDF$
time_mean)), "mean.chl" ),
  create_average(myDF, "predict", "time_mean", 1,floor(max(myDF$time_m
ean)), "mean.chl.pred" ),
  by.x = 'time_mean', by.y='time_mean'
)

daily.predict$diff <- daily.predict$mean.chl - daily.predict$mean.chl.
pred

png(file=paste("MLR1_Run_", run.nb, "_Boxplot.png", sep=''), bg="white
", width = 1280, height = 720)
boxplot(data.frame(mean.chl=daily.predict$mean.chl, mean.chl.pred = da
ily.predict$mean.chl.pred),
        main="Chlorophyll-a verification set measurements compared to
hourly model estimates\n(daily means of the measurements and estimates
)",
        ylab="Chlorophyll-a (µg/l)",
        names=c("Verification set measurements", "Model estimates"))
grid()
dev.off()
run.nb <- readline('What is the run number?: ')
png(file=paste("MLRA_Run_", run.nb, "_Chl_vs_model.png", sep=''), bg="
white", width = 1280, height = 720)

plot(check$time_mean/24,check$a.chlorophyll, type='l',
     xlab="time(days)",
     ylab="Chlorophyll-a (µg/l)",
     main="Comparason of measured Chlorophyll-a to model estimations"
)
lines(check$time_mean/24, y.predict,col='green')
legend("topleft", # places a legend at the appropriate place
       c("Chlorophyll-a","Model estimates"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',
       #lwd=c(2.5,2.5),
       col=c("black","green"))
grid()
dev.off()

png(file=paste("MLRA_Run_", run.nb, "_Boxplot.png", sep=''), bg="white
", width = 1280, height = 720)
```

```
boxplot(data.frame(mean.chl=daily.predict$mean.chl, mean.chl.pred = da
ily.predict$mean.chl.pred),
        main="Chlorophyll-a verification set measurements compared to
hourly model estimates\n(daily means of the measurements and estimates
)",
        ylab="Chlorophyll-a (µg/l)",
        names=c("Verification set measurements", "Model estimates"))
grid()
dev.off()



print(paste("check", sd(check$a.chlorophyll)))
print(summary(check$a.chlorophyll))
print(paste("predict", sd(y.predict)))
print(summary(y.predict))
```

Source code (MLR2 and smoothing)

```
source("http://users.metropolia.fi/~velimt/R/DOE_functions_v5.r")
if (!require("RColorBrewer")) {
  install.packages("RColorBrewer")
  library(RColorBrewer)
}

rms <- function(x) sqrt(mean(x^2))

#load data from CSV file
myData <- read.csv2("data_en.CSV")

#remove useless column
myData <- myData[,c(1:9)]

#add ellapsed time to the data
times <- strptime(myData$Date.and.time, format = "%d.%m.%Y %H:%M", tz=
"EET") #create POSIX time
initialTime <- as.numeric(times[4]) #first valid record
times <- (as.numeric(times) - initialTime) #set the time = 0
#times <- times/(24*60*60) #convert into days
times <- times/(24)/150 #convert into hours
myData$ellapsedTime <- times #add to the dataframe
myData <- myData[myData$ellapsedTime>=0,] #keep only valid samples
myData <- myData[complete.cases(myData),] #remove any row with at leas
t one NA

scale.robust<- function(x){
  return (scale(x,
  center = median(x),
  scale = mad(x)))

}

create_average <- function(DF, column_name_value, column_name_time, a,
b, column_name = 'mean.value'){
  mean.value <- numeric()
  time_mean <- numeric()
  for(i in a:b){
    time_mean <- append(time_mean, i)
    mean.value <- append(
      mean.value,
      mean(
        DF[[column_name_value]][
          DF[[column_name_time]]<=i&
            DF[[column_name_time]]>(i-1)
          ]
      )
    )
  }
  DF <- data.frame(time_mean=time_mean ,mean.value=mean.value)
  colnames(DF)<-c('time_mean', column_name)
  return(DF)
}

my.Data.Av <- create_average(myData,'Lämpötila...C.' , 'ellapsedTime',
0, 2615, 'temperature')
```

```
my.Data.Av <- merge(
  my.Data.Av,
  create_average(myData,'Johtokyky..µS.cm.' , 'ellapsedTime', 0, 2615,
'conductivity'),
  by.x = 'time_mean', by.y='time_mean'
)

my.Data.Av <- merge(
  my.Data.Av,
  create_average(myData,'pH....' , 'ellapsedTime', 0, 2615, 'pH'),
  by.x = 'time_mean', by.y='time_mean'
)

my.Data.Av <- merge(
  my.Data.Av,
  create_average(myData,'Sameus..FTU.' , 'ellapsedTime', 0, 2615, 'tur
bidity'),
  by.x = 'time_mean', by.y='time_mean'
)

my.Data.Av <- merge(
  my.Data.Av,
  create_average(myData,'Happi..mg.l.' , 'ellapsedTime', 0, 2615, 'DO'
),
  by.x = 'time_mean', by.y='time_mean'
)

my.Data.Av <- merge(
  my.Data.Av,
  create_average(myData,'A.klorofylli..µg.l.' , 'ellapsedTime', 0, 261
5, 'a.chlorophyll'),
  by.x = 'time_mean', by.y='time_mean'
)

my.Data.Av <- merge(
  my.Data.Av,
  create_average(myData,'Sinilevien.osuus..µg.l.' , 'ellapsedTime', 0,
2615, 'cyanobacteria'),
  by.x = 'time_mean', by.y='time_mean'
)

my.Data.Av.Scaled <- my.Data.Av
for(i in 2:6)my.Data.Av.Scaled[[i]] <- scale.robust(my.Data.Av.Scaled[
[i]])

maxX <- max(my.Data.Av.Scaled[[2]])
minX <- min(my.Data.Av.Scaled[[2]])
for(i in 3:6){
  maxX<-max(maxX, my.Data.Av.Scaled[[i]])
  minX<-min(minX, my.Data.Av.Scaled[[i]])
}


colors <- brewer.pal(7, "Dark2")

plot(my.Data.Av.Scaled$time_mean, my.Data.Av.Scaled$temperature, type=
'l', xlab='time (hours)', ylab = 'Scaled values', ylim=c(floor(minX),
ceiling(maxX)), col=colors[1])
```

```
for(i in 3:6){
  lines(my.Data.Av.Scaled$time_mean, my.Data.Av.Scaled[[i]], col=color
s[i])
}


my.Data.Av.Scaled.Smoothed <- my.Data.Av.Scaled

chlo.smooth.model <- loess(a.chlorophyll~time_mean, data=myDF<-data.fr
ame(time_mean=my.Data.Av.Scaled$time_mean,a.chlorophyll=my.Data.Av.Sca
led[[7]]),
                           span=.025)
myDF$pred <- predict(chlo.smooth.model, data=myDF)

my.Data.Av.Scaled.Smoothed$a.chlorophyll <- predict(chlo.smooth.model,
data=myDF)

plot(myDF$time_mean,myDF$a.chlorophyll, col='green', type='b', xlab="t
ime(days)", ylab="Chlorophyll-a(µg/l)", main = "Smoothing of Chlorophy
ll-a")
lines(myDF$time_mean,myDF$pred)
legend(2000,45, # places a legend at the appropriate place
       c("Chlorophyll-a","Smoothed fit"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',

       #lwd=c(2.5,2.5),
       col=c("green","black")) # gives the legend lines the correct co
lor and width

plot(myDF$time_mean/24,myDF$a.chlorophyll, col='green', type='b',xlim=
c(1000,1200)/24, xlab="time(days)", ylab="Chlorophyll-a(µg/l)", main =
"Smoothing of Chlorophyll-a (detailed view)")
lines(myDF$time_mean/24,myDF$pred)
legend(48,45, # places a legend at the appropriate place
       c("Chlorophyll-a","Smoothed fit"), # puts text in the legend
       cex = 0.8,
       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',

       #lwd=c(2.5,2.5),
       col=c("green","black")) # gives the legend lines the correct co
lor and width

plot(myDF$time_mean/24,myDF$a.chlorophyll, col='green', type='b',xlim=
c(1500,1850)/24, xlab="time(days)", ylab="Chlorophyll-a(µg/l)", main =
"Smoothing of Chlorophyll-a (detailed view)")
lines(myDF$time_mean/24,myDF$pred)
legend(62,45, # places a legend at the appropriate place
       c("Chlorophyll-a","Smoothed fit"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',
```

```
       #lwd=c(2.5,2.5),
       col=c("green","black")) # gives the legend lines the correct co
lor and width

plot(myDF$time_mean,scale.robust(myDF$a.chlorophyll), col='green', typ
e='b', xlab="time(h)", ylab="Scaled units")
lines(myDF$time_mean,scale.robust(myDF$pred))
lines(my.Data.Av.Scaled$time_mean, my.Data.Av.Scaled$turbidity, col='r
ed')

plot(myDF$time_mean,scale.robust(myDF$a.chlorophyll), col='green', typ
e='b', xlab="time(h)", ylab="Scaled units",xlim=c(1000,1200))
lines(myDF$time_mean,scale.robust(myDF$pred))
lines(my.Data.Av.Scaled$time_mean, my.Data.Av.Scaled$turbidity, col='r
ed')


turb.smooth.model <- loess(turbidity~time_mean, data=myDF.turbidity<-d
ata.frame(time_mean=my.Data.Av.Scaled$time_mean,turbidity=my.Data.Av.S
caled$turbidity),
                           span=.025)
myDF.turbidity$pred <- predict(turb.smooth.model, data=myDF.turbidity)
my.Data.Av.Scaled.Smoothed$turbidity <- predict(turb.smooth.model, dat
a=myDF)


plot(myDF.turbidity$time_mean,myDF.turbidity$turbidity,type='l',
     xlab="time(h)",ylab="Scaled units",
     main = "Smoothing of turbidity"
     )
lines(myDF.turbidity$time_mean,myDF.turbidity$pred,col='red', lwd=3)
legend('topright', # places a legend at the appropriate place
       c("Turbidity","Smoothed fit"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',

       #lwd=c(2.5,2.5),
       col=c("black","red")) # gives the legend lines the correct colo
r and width

plot(myDF.turbidity$time_mean,myDF.turbidity$turbidity,type='l',xlim=c
(1000,1200))
lines(myDF.turbidity$time_mean,myDF.turbidity$pred,col='red')


conductivity.smooth.model <- loess(conductivity~time_mean,
                                   data=myDF.conductivity<-data.frame(
time_mean=my.Data.Av.Scaled$time_mean,conductivity=my.Data.Av.Scaled$c
onductivity),
                                   span=.03)

my.Data.Av.Scaled.Smoothed$conductivity <- predict(conductivity.smooth
.model, data=my.Data.Av.Scaled)

plot(my.Data.Av.Scaled$time_mean, my.Data.Av.Scaled$conductivity,type
= 'l', xlab="time(h)", ylab="Scaled units",
     main = "Smoothing of conductivity")
```

```
lines(my.Data.Av.Scaled.Smoothed$time_mean, my.Data.Av.Scaled.Smoothed
$conductivity, col='blue')
legend('bottomright', # places a legend at the appropriate place
       c("Conductivity","Smoothed fit"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',

       #lwd=c(2.5,2.5),
       col=c("black","blue")) # gives the legend lines the correct col
or and width

plot(my.Data.Av.Scaled$time_mean, my.Data.Av.Scaled$conductivity,type
= 'l', xlab="time(h)", ylab="Scaled units", xlim=c(600,700))
lines(my.Data.Av.Scaled.Smoothed$time_mean, my.Data.Av.Scaled.Smoothed
$conductivity, col='blue')
legend(675,4, # places a legend at the appropriate place
       c("Conductivity","Smoothed fit"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)
       cex = .75,
       bty = 'n',

       #lwd=c(2.5,2.5),
       col=c("black","blue")) # gives the legend lines the correct col
or and width

pH.smooth.model <- loess(pH~time_mean,
                         data=myDF.pH<-data.frame(time_mean=my.Data.Av
.Scaled$time_mean,pH=my.Data.Av.Scaled$pH),
                         span=.05)

my.Data.Av.Scaled.Smoothed$pH <- predict(pH.smooth.model, data=my.Data
.Av.Scaled)

plot(my.Data.Av.Scaled$time_mean, my.Data.Av.Scaled$pH,type = 'l', xla
b="time(h)", ylab="Scaled units",
     main = "Smoothing of pH")
lines(my.Data.Av.Scaled.Smoothed$time_mean, my.Data.Av.Scaled.Smoothed
$pH, col='orange', lwd=2)
legend("topright", # places a legend at the appropriate place
       c("pH","Smoothed fit"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',

       #lwd=c(2.5,2.5),
       col=c("black","green")) # gives the legend lines the correct co
lor and width


plot(my.Data.Av.Scaled$time_mean/24, my.Data.Av.Scaled$pH,type = 'l',
xlab="time(h)", ylab="pH (scaled units)", xlim=c(500,688)/24)
lines(my.Data.Av.Scaled.Smoothed$time_mean/24, my.Data.Av.Scaled.Smoot
hed$pH, col='green')
legend("topleft", # places a legend at the appropriate place
       c("Dissolved Oxygen","Smoothed fit"), # puts text in the legend
```

```
       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',

       #lwd=c(2.5,2.5),
       col=c("black","blue")) # gives the legend lines the correct col
or and width


DO.smooth.model <- loess(DO~time_mean,
                         data=myDF.DO<-data.frame(time_mean=my.Data.Av
.Scaled$time_mean,DO=my.Data.Av.Scaled$DO),
                         span=.04)

my.Data.Av.Scaled.Smoothed$DO <- predict(DO.smooth.model, data=my.Data
.Av.Scaled)

plot(my.Data.Av.Scaled$time_mean, my.Data.Av.Scaled$DO,type = 'l', xla
b="time(h)", ylab="Scaled units",
     main = "Smoothing of Dissolved Oxygen")
lines(my.Data.Av.Scaled.Smoothed$time_mean, my.Data.Av.Scaled.Smoothed
$DO, col='blue')
legend("topleft", # places a legend at the appropriate place
       c("Dissolved Oxygen","Smoothed fit"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',

       #lwd=c(2.5,2.5),
       col=c("black","blue")) # gives the legend lines the correct col
or and width


temperature.smooth.model <- loess(temperature~time_mean,
                                  data=myDF.temperature<-data.frame(ti
me_mean=my.Data.Av.Scaled$time_mean,temperature=my.Data.Av.Scaled$temp
erature),
                                  span=.03)

my.Data.Av.Scaled.Smoothed$temperature <- predict(temperature.smooth.m
odel, data=my.Data.Av.Scaled)

plot(my.Data.Av.Scaled$time_mean, my.Data.Av.Scaled$temperature,type =
'l', xlab="time(h)", ylab="Scaled units",
     main = "Smoothing of temperature")
lines(my.Data.Av.Scaled.Smoothed$time_mean, my.Data.Av.Scaled.Smoothed
$temperature, col='red')
legend("topright", # places a legend at the appropriate place
       c("Temperature","Smoothed fit"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',

       #lwd=c(2.5,2.5),
       col=c("black","red")) # gives the legend lines the correct colo
r and width
```

```
plot(myDF$time_mean/24,scale.robust(myDF$pred), col='green', type='l',
xlab="time(days)", ylab="Scaled units",
     main="Turbidity and Chlorophyll-a (smoothed and standardized)")
lines(myDF.turbidity$time_mean/24, myDF.turbidity$pred, col='brown')
legend("topright", # places a legend at the appropriate place
       c("Chlorophyll-a","Turbidity"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',

       #lwd=c(2.5,2.5),
       col=c("green","brown")) # gives the legend lines the correct co
lor and width


plot(myDF$time_mean/24,myDF$a.chlorophyll, col='green',xlim=c(1500,185
0)/24, xlab="time(h)", ylab="Chlorophyll-a(µg/l)")


temp <- sample(nrow(my.Data.Av.Scaled),round(nrow(my.Data.Av.Scaled)*.
1)) #take 10% of the data to check the model and 90% to train it
check <- my.Data.Av.Scaled[temp,]
check <- check[order(check$time_mean),]

train <- my.Data.Av.Scaled[-temp,]
train <- train[order(train$time_mean),]
rm(temp)

X=train[,2:6]
Y=train[,7]

models<-quad.CV.fwd(X,Y,20,2)
which.max(models$Q2)
fmla <- models$formulas[[which.max(models$Q2)]]
print(summary(model1<-lm(fmla,data=train)))


continue = TRUE
model2 <- model1
fmla2 <- fmla

while(continue){
  least.significant <- names(summary(model2)$coefficients[,4])[which.m
ax(summary(model2)$coefficients[,4])]
  least.significant.p.value <- summary(model2)$coefficients[,4][which.
max(summary(model2)$coefficients[,4])]
  cat(paste("The least significant therm is: ", least.significant, " p
-value: ", least.significant.p.value))
  ans <- readline('Do you want to remove the least significant therm (
answer y or n): ')
  if(tolower(ans)=='y') continue = TRUE else{
    continue = FALSE
    break
  }
  print(continue)
  fmla2 <- update(fmla2, paste(".~.-", least.significant))
```

```
  print(summary(model2<-lm(fmla2,data=train)))
}

y.predict <- predict(model2, newdata = check)
plot(check$time_mean ,y.predict, type='l')
lines(check$time_mean, check$a.chlorophyll,col='green')
plot(check$time_mean,check$a.chlorophyll, type='l',
     xlab="time(days)",
     ylab="Chlorophyll-a (µg/l)",
     main="Comparason of measured Chlorophyll-a to model estimations"
     )
lines(check$time_mean, y.predict,col='green')
legend("topleft", # places a legend at the appropriate place
       c("Chlorophyll-a","Model estimates"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',
       cex=.75,
       #lwd=c(2.5,2.5),
       col=c("black","green"))

plot(check$time_mean ,y.predict-check$a.chlorophyll, type = 'l')
print(paste("predict", sd(y.predict)))
print(summary(y.predict))
print(paste("cehck", sd(check$a.chlorophyll)))
print(summary(check$a.chlorophyll))

boxplot(y.predict-check$a.chlorophyll, outline=FALSE)
boxplot(check$a.chlorophyll, outline=FALSE)

plot(check$time_mean ,check$a.chlorophyll, type='l')
lines(check$time_mean, y.predict,col='green')

myDF <- check
myDF$predict <- y.predict
myDF$time_mean <- myDF$time_mean/732

#(DF, column_name_value, column_name_time, a, b, column_name = 'mean.v
alue')
monthly.predict <- merge(
  create_average(myDF, "a.chlorophyll", "time_mean", 1,3, "mean.chl" )
,
  create_average(myDF, "predict", "time_mean", 1,3, "mean.chl.pred" ),
  by.x = 'time_mean', by.y='time_mean'
)

monthly.predict$diff <- monthly.predict$mean.chl - monthly.predict$mea
n.chl.pred
boxplot(data.frame(mean.chl=monthly.predict$mean.chl, mean.chl.pred =
monthly.predict$mean.chl.pred))


myDF <- check
myDF$predict <- y.predict
myDF$time_mean <- myDF$time_mean/24

daily.predict <- merge(
```

```
  create_average(myDF, "a.chlorophyll", "time_mean", 1,floor(max(myDF$
time_mean)), "mean.chl" ),
  create_average(myDF, "predict", "time_mean", 1,floor(max(myDF$time_m
ean)), "mean.chl.pred" ),
  by.x = 'time_mean', by.y='time_mean'
)

daily.predict$diff <- daily.predict$mean.chl - daily.predict$mean.chl.
pred
boxplot(daily.predict[complete.cases(daily.predict),]$diff)
boxplot(data.frame(mean.chl=daily.predict$mean.chl, mean.chl.pred = da
ily.predict$mean.chl.pred),
        main="Chlorophyll-a verification set measurements compared to
hourly model estimates\n(daily means of the measurements and estimates
)",
        ylab="Chlorophyll-a (µg/l)",
        names=c("Verification set measurements", "Model estimates"))
grid()
print(summary(daily.predict[complete.cases(daily.predict),]$diff))


run.nb <- readline('What is the run number?: ')
png(file=paste("MLR2_Run_", run.nb, "_Chl_vs_model.png", sep=''), bg="
white", width = 1280, height = 720)

plot(check$time_mean/24,check$a.chlorophyll, type='l',
     xlab="time(days)",
     ylab="Chlorophyll-a (µg/l)",
     main="Comparason of measured Chlorophyll-a to model estimations"
)
lines(check$time_mean/24, y.predict,col='green')
legend("topleft", # places a legend at the appropriate place
       c("Chlorophyll-a","Model estimates"), # puts text in the legend

       lty=c(1,1), # gives the legend appropriate symbols (lines)

       bty = 'n',
       #lwd=c(2.5,2.5),
       col=c("black","green"))
grid()
dev.off()

png(file=paste("MLR2_Run_", run.nb, "_Boxplot.png", sep=''), bg="white
", width = 1280, height = 720)
boxplot(data.frame(mean.chl=daily.predict$mean.chl, mean.chl.pred = da
ily.predict$mean.chl.pred),
        main="Chlorophyll-a verification set measurements compared to
hourly model estimates\n(daily means of the measurements and estimates
)",
        ylab="Chlorophyll-a (µg/l)",
        names=c("Verification set measurements", "Model estimates"))
grid()
dev.off()

print(paste("cehck", sd(check$a.chlorophyll)))
print(summary(check$a.chlorophyll))
print(paste("predict", sd(y.predict)))
print(summary(y.predict))
```

```
print("RMS")
print(rms(y.predict-check$a.chlorophyll))


cat("rel error of the means ", (1-(mean(y.predict)/mean(check$a.chloro
phyll)))*100)
```

Source code (PCA)

```
setwd("C:/Users/Benoit/Dropbox/Thesis/Data/R")
#source("http://users.metropolia.fi/~velimt/R/DOE_functions_v5.r")

#something <- sys.frame(1)

#script.dir <- dirname(something$fileName)
T<-TRUE
if (!require("RColorBrewer")) {
  install.packages("RColorBrewer")
  library(RColorBrewer)
}

#load data from CSV file
myData <- read.csv2("data_en.CSV")

#remove useless column
myData <- myData[,c(1:9)]

#add ellapsed time to the data
times <- strptime(myData$Date.and.time, format = "%d.%m.%Y %H:%M", tz=
"EET") #create POSIX time
initialTime <- as.numeric(times[4]) #first valid record
times <- (as.numeric(times) - initialTime) #set the time = 0
#times <- times/(24*60*60) #convert into days
times <- times/(24)/150 #convert into hours
myData$ellapsedTime <- times #add to the dataframe
myData <- myData[myData$ellapsedTime>=0,] #keep only valid samples
myData <- myData[complete.cases(myData),] #remove any row with at leas
t one NA

create_average <- function(DF, column_name_value, column_name_time, a,
b, column_name = 'mean.value'){
  mean.value <- numeric()
  time_mean <- numeric()
  for(i in a:b){
    time_mean <- append(time_mean, i)
    mean.value <- append(
      mean.value,
      mean(
        DF[[column_name_value]][
          DF[[column_name_time]]<=i&
            DF[[column_name_time]]>(i-1)
          ]
      )
    )
  }
  DF <- data.frame(time_mean=time_mean ,mean.value=mean.value)
  colnames(DF)<-c('time_mean', column_name)
  return(DF)
}

my.Data.Av <- create_average(myData,'Lämpötila...C.' , 'ellapsedTime',
0, 2615, 'temperature')
my.Data.Av <- merge(
  my.Data.Av,
```

```
  create_average(myData,'Johtokyky..µS.cm.' , 'ellapsedTime', 0, 2615,
'conductivity'),
  by.x = 'time_mean', by.y='time_mean'
)

my.Data.Av <- merge(
  my.Data.Av,
  create_average(myData,'pH....' , 'ellapsedTime', 0, 2615, 'pH'),
  by.x = 'time_mean', by.y='time_mean'
)

my.Data.Av <- merge(
  my.Data.Av,
  create_average(myData,'Sameus..FTU.' , 'ellapsedTime', 0, 2615, 'tur
bidity'),
  by.x = 'time_mean', by.y='time_mean'
)

my.Data.Av <- merge(
  my.Data.Av,
  create_average(myData,'Happi..mg.l.' , 'ellapsedTime', 0, 2615, 'DO'
),
  by.x = 'time_mean', by.y='time_mean'
)

my.Data.Av <- merge(
  my.Data.Av,
  create_average(myData,'A.klorofylli..µg.l.' , 'ellapsedTime', 0, 261
5, 'a.chlorophyll'),
  by.x = 'time_mean', by.y='time_mean'
)

my.Data.Av <- merge(
  my.Data.Av,
  create_average(myData,'Sinilevien.osuus..µg.l.' , 'ellapsedTime', 0,
2615, 'cyanobacteria'),
  by.x = 'time_mean', by.y='time_mean'
)

my.Data.Av.Scaled <- my.Data.Av
for(i in 2:6)my.Data.Av.Scaled[[i]] <- scale(my.Data.Av.Scaled[[i]],T,
T)

maxX <- max(my.Data.Av.Scaled[[2]])
minX <- min(my.Data.Av.Scaled[[2]])
for(i in 3:6){
  maxX<-max(maxX, my.Data.Av.Scaled[[i]])
  minX<-min(minX, my.Data.Av.Scaled[[i]])
}


my.Data.Av.Scaled.Smoothed <- my.Data.Av.Scaled

chlo.smooth.model <- loess(a.chlorophyll~time_mean, data=myDF<-data.fr
ame(time_mean=my.Data.Av.Scaled$time_mean,a.chlorophyll=my.Data.Av.Sca
led[[7]]),
                           span=.005)
myDF$pred <- predict(chlo.smooth.model, data=myDF)
```

```
my.Data.Av.Scaled.Smoothed$a.chlorophyll <- predict(chlo.smooth.model,
data=myDF)

turb.smooth.model <- loess(turbidity~time_mean, data=myDF.turbidity<-d
ata.frame(time_mean=my.Data.Av.Scaled$time_mean,turbidity=my.Data.Av.S
caled$turbidity),
                           span=.005)
myDF.turbidity$pred <- predict(turb.smooth.model, data=myDF.turbidity)
my.Data.Av.Scaled.Smoothed$turbidity <- predict(turb.smooth.model, dat
a=myDF)


conductivity.smooth.model <- loess(conductivity~time_mean,
                                   data=myDF.conductivity<-data.frame(
time_mean=my.Data.Av.Scaled$time_mean,conductivity=my.Data.Av.Scaled$c
onductivity),
                                   span=.03)

my.Data.Av.Scaled.Smoothed$conductivity <- predict(conductivity.smooth
.model, data=my.Data.Av.Scaled)


pH.smooth.model <- loess(pH~time_mean,
                         data=myDF.pH<-data.frame(time_mean=my.Data.Av
.Scaled$time_mean,pH=my.Data.Av.Scaled$pH),
                         span=.01)

my.Data.Av.Scaled.Smoothed$pH <- predict(pH.smooth.model, data=my.Data
.Av.Scaled)

DO.smooth.model <- loess(DO~time_mean,
                         data=myDF.DO<-data.frame(time_mean=my.Data.Av
.Scaled$time_mean,DO=my.Data.Av.Scaled$DO),
                         span=.04)

my.Data.Av.Scaled.Smoothed$DO <- predict(DO.smooth.model, data=my.Data
.Av.Scaled)
temperature.smooth.model <- loess(temperature~time_mean,
                                  data=myDF.temperature<-data.frame(ti
me_mean=my.Data.Av.Scaled$time_mean,temperature=my.Data.Av.Scaled$temp
erature),
                                  span=.03)

my.Data.Av.Scaled.Smoothed$temperature <- predict(temperature.smooth.m
odel, data=my.Data.Av.Scaled)

T <- TRUE

my.pca <- prcomp(my.Data.Av[2:7], center= T, scale = T)

plot(my.pca, type='l')

T <- my.pca$x
P <- my.pca$rotation

plot(T[,1],T[,2], pch='.', cex=5)

biplot(T,P)
```

```
biplot(T,P, xlabs = rep('.',length(T)/6), col=c('blue'), xlim = c(-5,1
0), ylim=c(-5,10))

lower.lim <- 10
upper.lim <- 14
points(T[my.Data.Av$temperature<lower.lim,1], T[my.Data.Av$temperature
<lower.lim,2], pch='.', cex = 2, col='red')
points(T[my.Data.Av$temperature>upper.lim,1], T[my.Data.Av$temperature
>upper.lim,2], pch='.', cex = 2, col='blue')
points(T[my.Data.Av$temperature<upper.lim&my.Data.Av$temperature>lower
.lim,1], T[my.Data.Av$temperature<upper.lim&my.Data.Av$temperature>low
er.lim,2], pch='.', cex = 2, col='darkgreen')

plot(T[,1],T[,3], pch='.')
plot(T[,1],T[,4], pch='.')

points(T[my.Data.Av$temperature<lower.lim,1], T[my.Data.Av$temperature
<lower.lim,4], pch='.', cex = 2, col='red')
points(T[my.Data.Av$temperature>upper.lim,1], T[my.Data.Av$temperature
>upper.lim,4], pch='.', cex = 2, col='blue')
points(T[my.Data.Av$temperature<upper.lim&my.Data.Av$temperature>lower
.lim,1], T[my.Data.Av$temperature<upper.lim&my.Data.Av$temperature>low
er.lim,4], pch='.', cex = 2, col='darkgreen')


plot(T[,1],T[,5], pch='.')
plot(T[,1],T[,6], pch='.')

plot(T[,2],T[,3], pch='.')
plot(T[,2],T[,4], pch='.')
plot(T[,2],T[,5], pch='.')
plot(T[,2],T[,6], pch='.')


plot(T[,3],T[,4], pch='.')
plot(T[,3],T[,5], pch='.')
plot(T[,3],T[,6], pch='.')

plot(T[,4],T[,5], pch='.')
plot(T[,4],T[,6], pch='.')


plot(T[,5],T[,6], pch='.')



####

plot(T[,1],T[,2], pch='.')

lower.lim <- 1
upper.lim <- 16
points(T[my.Data.Av$a.chlorophyll<upper.lim&my.Data.Av$a.chlorophyll>l
ower.lim,1], T[my.Data.Av$a.chlorophyll<upper.lim&my.Data.Av$a.chloro
hyll>lower.lim,2], pch='.', cex = 5, col='darkgreen')
points(T[my.Data.Av$a.chlorophyll<lower.lim,1], T[my.Data.Av$a.chloro
hyll<lower.lim,2], pch='.', cex = 5, col='red')
points(T[my.Data.Av$a.chlorophyll>upper.lim,1], T[my.Data.Av$a.chloro
hyll>upper.lim,2], pch='.', cex = 5, col='blue')
```

```
lower.lim <- 0
upper.lim <- 0.8
points(T[my.Data.Av$turbidity<upper.lim&my.Data.Av$turbidity>lower.lim
,1], T[my.Data.Av$turbidity<upper.lim&my.Data.Av$turbidity>lower.lim,2
], pch='.', cex = 5, col='green')
points(T[my.Data.Av$turbidity<lower.lim,1], T[my.Data.Av$turbidity<low
er.lim,2], pch='.', cex = 5, col='pink')
points(T[my.Data.Av$turbidity>upper.lim,1], T[my.Data.Av$turbidity>upp
er.lim,2], pch='.', cex = 5, col='brown')



###

T<-TRUE

TempCol <- rev(colorRampPalette(brewer.pal(11,"RdBu"))(100))

Temp_0_to_100 <- round(100*((scale(my.Data.Av$temperature,T,T)-min(sca
le(my.Data.Av$temperature,T,T)))/(max(scale(my.Data.Av$temperature,T,T
)-min(scale(my.Data.Av$temperature,T,T))))))

my.pca <- prcomp(my.Data.Av[2:7], center= T, scale = T)
plot(my.pca, type='l', main="PCA variance with respect to the principa
l components")


T <- my.pca$x
P <- my.pca$rotation


#Rd|Bu


smoothed.pc1 = lowess(T[,1],f=.01)
smoothed.pc2 = lowess(T[,2],f=.01)
#lines(smoothed.pc1,col='red')
plot(T[,1],T[,2], pch=16,cex=1,
     col=TempCol[Temp_0_to_100], ylim=c(-1,16),xlim=c(-3,5),
     xaxt='n', ann=FALSE,yaxt='n')
lines(smoothed.pc1$y,smoothed.pc2$y, col='black',lwd=2)
par(new=TRUE)
biplot(T[,1:2],P[,1:2], xlabs = rep('.',length(T[,1])),
       col=c('black','darkgreen'),ylim=c(-1,16),xlim=c(-3,5))

T<-TRUE

TempCol <- rev(colorRampPalette(brewer.pal(11,"RdBu"))(100))

Temp_0_to_100 <- round(100*((scale(my.Data.Av$temperature,T,T)-min(sca
le(my.Data.Av$temperature,T,T)))/(max(scale(my.Data.Av$temperature,T,T
)-min(scale(my.Data.Av$temperature,T,T))))))

my.pca <- prcomp(my.Data.Av[2:7], center= T, scale = T)
plot(my.pca, type='l', main="PCA variance with respect to the principa
l components")


T <- my.pca$x
```

```
P <- my.pca$rotation


#Rd|Bu


smoothed.pc1 = lowess(T[,1],f=.01)
smoothed.pc2 = lowess(T[,3],f=.01)
#lines(smoothed.pc1,col='red')
plot(T[,1],T[,3], pch=16,cex=1,
     col=TempCol[Temp_0_to_100], ylim=c(-5,5),xlim=c(-3,5),
     xaxt='n', ann=FALSE,yaxt='n')
lines(smoothed.pc1$y,smoothed.pc2$y, col='black',lwd=2)
par(new=TRUE)
biplot(T[,c(1,3)],P[,c(1,3)], xlabs = rep('.',length(T[,1])),
       col=c('black','darkgreen'),ylim=c(-5,5),xlim=c(-3,5))


T<-TRUE

TempCol <- rev(colorRampPalette(brewer.pal(11,"RdBu"))(100))

Temp_0_to_100 <- round(100*((scale(my.Data.Av$temperature,T,T)-min(sca
le(my.Data.Av$temperature,T,T)))/(max(scale(my.Data.Av$temperature,T,T
)-min(scale(my.Data.Av$temperature,T,T))))))

my.pca <- prcomp(my.Data.Av[2:7], center= T, scale = T)
plot(my.pca, type='l', main="PCA variance with respect to the principa
l components")


T <- my.pca$x
P <- my.pca$rotation


#Rd|Bu


smoothed.pc1 = lowess(T[,2],f=.01)
smoothed.pc2 = lowess(T[,3],f=.01)
#lines(smoothed.pc1,col='red')
plot(T[,2],T[,3], pch=16,cex=1,
     col=TempCol[Temp_0_to_100], ylim=c(-5,5),xlim=c(-2,13),
     xaxt='n', ann=FALSE,yaxt='n')
lines(smoothed.pc1$y,smoothed.pc2$y, col='black',lwd=2)
par(new=TRUE)
biplot(T[,c(2,3)],P[,2:3], xlabs = rep('.',length(T[,1])),
       col=c('black','darkgreen'),ylim=c(-5,5),xlim=c(-2,13))

#####


T<-TRUE

TempCol <- rev(colorRampPalette(brewer.pal(11,"RdBu"))(100))

Temp_0_to_100 <- round(100*((scale(my.Data.Av$a.chlorophyll,T,T)-min(s
cale(my.Data.Av$a.chlorophyll,T,T)))/(max(scale(my.Data.Av$a.chlorophy
ll,T,T)-min(scale(my.Data.Av$a.chlorophyll,T,T))))))
```

```
my.pca <- prcomp(my.Data.Av[2:7], center= T, scale = T)
plot(my.pca, type='l', main="PCA variance with respect to the principa
l components")


T <- my.pca$x
P <- my.pca$rotation


#Rd|Bu


smoothed.pc1 = lowess(T[,1],f=.01)
smoothed.pc2 = lowess(T[,2],f=.01)
#lines(smoothed.pc1,col='red')
plot(T[,1],T[,2], pch=16,cex=1,
     col=TempCol[Temp_0_to_100], ylim=c(-1,16),xlim=c(-3,5),
     xaxt='n', ann=FALSE,yaxt='n')
lines(smoothed.pc1$y,smoothed.pc2$y, col='black',lwd=1)
par(new=TRUE)
biplot(T[,1:2],P[,1:2], xlabs = rep('.',length(T[,1])),
       col=c('black','darkgreen'),ylim=c(-1,16),xlim=c(-3,5))


smoothed.pc1 = lowess(T[,2],f=.01)
smoothed.pc2 = lowess(T[,3],f=.01)
#lines(smoothed.pc1,col='red')
plot(T[,2],T[,3], pch=16,cex=1,
     col=TempCol[Temp_0_to_100], ylim=c(-4,4),xlim=c(-1,11),
     xaxt='n', ann=FALSE,yaxt='n')
lines(smoothed.pc1$y,smoothed.pc2$y, col='black',lwd=1)
par(new=TRUE)
biplot(T[,2:3],P[,2:3], xlabs = rep('.',length(T[,1])),
       col=c('black','darkgreen'),ylim=c(-4,4),xlim=c(-1,11))



T<-TRUE

TempCol <- rev(colorRampPalette(brewer.pal(11,"RdBu"))(100))

Temp_0_to_100 <- round(100*((scale(my.Data.Av$pH,T,T)-min(scale(my.Dat
a.Av$pH,T,T)))/(max(scale(my.Data.Av$pH,T,T)-min(scale(my.Data.Av$pH,T
,T))))))

my.pca <- prcomp(my.Data.Av[2:7], center= T, scale = T)
plot(my.pca, type='l', main="PCA variance with respect to the principa
l components")


T <- my.pca$x
P <- my.pca$rotation


#Rd|Bu


smoothed.pc1 = lowess(T[,1],f=.01)
```

```
smoothed.pc2 = lowess(T[,2],f=.01)
#lines(smoothed.pc1,col='red')
plot(T[,1],T[,2], pch=16,cex=1,
     col=TempCol[Temp_0_to_100], ylim=c(-1,16),xlim=c(-3,5),
     xaxt='n', ann=FALSE,yaxt='n')
lines(smoothed.pc1$y,smoothed.pc2$y, col='black',lwd=1)
par(new=TRUE)
biplot(T[,1:2],P[,1:2], xlabs = rep('.',length(T[,1])),
       col=c('black','darkgreen'),ylim=c(-1,16),xlim=c(-3,5))



smoothed.pc1 = lowess(T[,3],f=.01)
smoothed.pc2 = lowess(T[,4],f=.01)
#lines(smoothed.pc1,col='red')
plot(T[,3],T[,4], pch=16,cex=1,
     col=TempCol[Temp_0_to_100], ylim=c(-5,5),xlim=c(-3,3),
     xaxt='n', ann=FALSE,yaxt='n')
lines(smoothed.pc1$y,smoothed.pc2$y, col='black',lwd=1)
par(new=TRUE)
biplot(T[,3:4],P[,3:4], xlabs = rep('.',length(T[,1])),
       col=c('black','darkgreen'),ylim=c(-5,5),xlim=c(-3,3))


smoothed.pc1 = lowess(T[,5],f=.01)
smoothed.pc2 = lowess(T[,6],f=.01)
#lines(smoothed.pc1,col='red')
plot(T[,5],T[,6], pch=16,cex=1,
     col=TempCol[Temp_0_to_100], ylim=c(-1,1),xlim=c(-2,2),
     xaxt='n', ann=FALSE,yaxt='n')
lines(smoothed.pc1$y,smoothed.pc2$y, col='black',lwd=1)
par(new=TRUE)
biplot(T[,5:6],P[,5:6], xlabs = rep('.',length(T[,1])),
       col=c('black','darkgreen'),ylim=c(-1,1),xlim=c(-2,2))



smoothed.pc1 = lowess(T[,1],f=.01)
smoothed.pc2 = lowess(T[,6],f=.01)
#lines(smoothed.pc1,col='red')
plot(T[,1],T[,6], pch=16,cex=1,
     col=TempCol[Temp_0_to_100], ylim=c(-1,1),xlim=c(-2,2),
     xaxt='n', ann=FALSE,yaxt='n')
lines(smoothed.pc1$y,smoothed.pc2$y, col='black',lwd=1)
par(new=TRUE)
biplot(T[,c(1,6)],P[,c(1,6)], xlabs = rep('.',length(T[,1])),
       col=c('black','darkgreen'),ylim=c(-1,1),xlim=c(-2,2))



plot(T[,1],T[,2], pch='.')

lower.lim <- 10
upper.lim <- 14
points(T[my.Data.Av$temperature<=lower.lim,1], T[my.Data.Av$temperatur
e<=lower.lim,2], pch='.', cex = 5, col='blue')
points(T[my.Data.Av$temperature>=upper.lim,1], T[my.Data.Av$temperatur
e>=upper.lim,2], pch='.', cex = 5, col='red')
```

```
points(T[my.Data.Av$temperature<upper.lim&my.Data.Av$temperature>lower
.lim,1], T[my.Data.Av$temperature<upper.lim&my.Data.Av$temperature>low
er.lim,5], pch='.', cex = 5, col='darkgreen')

plot(T[,1],T[,2], pch='.')
lower.lim <- summary(my.Data.Av$conductivity)[5]
upper.lim <- summary(my.Data.Av$conductivity)[5]
points(T[my.Data.Av$conductivity<=lower.lim,1], T[my.Data.Av$conductiv
ity<=lower.lim,2], pch='.', cex = 5, col='blue')
points(T[my.Data.Av$conductivity>=upper.lim,1], T[my.Data.Av$conductiv
ity>=upper.lim,2], pch='.', cex = 5, col='red')
points(T[my.Data.Av$conductivity<upper.lim&my.Data.Av$conductivity>low
er.lim,1], T[my.Data.Av$conductivity<upper.lim&my.Data.Av$conductivity
>lower.lim,5], pch='.', cex = 5, col='darkgreen')


plot(T[,1],T[,2], pch='.')
lower.lim <- summary(my.Data.Av$pH)[3]
upper.lim <- summary(my.Data.Av$pH)[3]
points(T[my.Data.Av$pH<=lower.lim,1], T[my.Data.Av$pH<=lower.lim,2], p
ch='.', cex = 5, col='blue')
points(T[my.Data.Av$pH>=upper.lim,1], T[my.Data.Av$pH>=upper.lim,2], p
ch='.', cex = 5, col='red')
points(T[my.Data.Av$pH<upper.lim&my.Data.Av$pH>lower.lim,1], T[my.Data
.Av$pH<upper.lim&my.Data.Av$pH>lower.lim,5], pch='.', cex = 5, col='da
rkgreen')


plot(T[,1],T[,2], pch='.')
lower.lim <- summary(my.Data.Av$turbidity)[2]
upper.lim <- summary(my.Data.Av$turbidity)[5]
points(T[my.Data.Av$turbidity<=lower.lim,1], T[my.Data.Av$turbidity<=l
ower.lim,2], pch='.', cex = 5, col='blue')
points(T[my.Data.Av$turbidity>=upper.lim,1], T[my.Data.Av$turbidity>=u
pper.lim,2], pch='.', cex = 5, col='red')
points(T[my.Data.Av$turbidity<upper.lim&my.Data.Av$turbidity>lower.lim
,1], T[my.Data.Av$turbidity<upper.lim&my.Data.Av$turbidity>lower.lim,5
], pch='.', cex = 5, col='darkgreen')

plot(T[,1],T[,2], pch='.', cex=5)
lower.lim <- summary(my.Data.Av$a.chlorophyll)[2]
upper.lim <- summary(my.Data.Av$a.chlorophyll)[5]
points(T[my.Data.Av$a.chlorophyll<=lower.lim,1], T[my.Data.Av$a.chloro
phyll<=lower.lim,2], pch='.', cex = 5, col='blue')
points(T[my.Data.Av$a.chlorophyll>=upper.lim,1], T[my.Data.Av$a.chloro
phyll>=upper.lim,2], pch='.', cex = 5, col='red')
points(T[my.Data.Av$a.chlorophyll<upper.lim&my.Data.Av$a.chlorophyll>l
ower.lim,1], T[my.Data.Av$a.chlorophyll<upper.lim&my.Data.Av$a.chlorop
hyll>lower.lim,5], pch='.', cex = 5, col='darkgreen')

plot(T[,1],T[,2], pch='.', cex=5)
lower.lim <- summary(my.Data.Av$DO)[2]
upper.lim <- summary(my.Data.Av$DO)[5]
points(T[my.Data.Av$DO<=lower.lim,1], T[my.Data.Av$DO<=lower.lim,2], p
ch='.', cex = 5, col='blue')
points(T[my.Data.Av$DO>=upper.lim,1], T[my.Data.Av$DO>=upper.lim,2], p
ch='.', cex = 5, col='red')
```

```
points(T[my.Data.Av$DO<upper.lim&my.Data.Av$DO>lower.lim,1], T[my.Data
.Av$DO<upper.lim&my.Data.Av$DO>lower.lim,5], pch='.', cex = 5, col='da
rkgreen')



plot(T[,1],T[,3], pch='.')
plot(T[,1],T[,4], pch='.')

points(T[myData$temperature<lower.lim,1], T[myData$temperature<lower.l
im,4], pch='.', cex = 2, col='red')
points(T[myData$temperature>upper.lim,1], T[myData$temperature>upper.l
im,4], pch='.', cex = 2, col='blue')
points(T[myData$temperature<upper.lim&myData$temperature>lower.lim,1],
T[myData$temperature<upper.lim&myData$temperature>lower.lim,4], pch='.
', cex = 2, col='darkgreen')


plot(T[,1],T[,5], pch='.')
plot(T[,1],T[,6], pch='.')

plot(T[,2],T[,3], pch='.')
plot(T[,2],T[,4], pch='.')
plot(T[,2],T[,5], pch='.')
plot(T[,2],T[,6], pch='.')


plot(T[,3],T[,4], pch='.')
plot(T[,3],T[,5], pch='.')
plot(T[,3],T[,6], pch='.')

plot(T[,4],T[,5], pch='.')
plot(T[,4],T[,6], pch='.')


plot(T[,5],T[,6], pch='.')



####

plot(T[,1],T[,2], pch='.')

lower.lim <- 1
upper.lim <- 16
points(T[myData$A.klorofylli..µg.l.<upper.lim&myData$A.klorofylli..µg.
l.>lower.lim,1], T[myData$A.klorofylli..µg.l.<upper.lim&myData$A.kloro
fylli..µg.l.>lower.lim,2], pch='.', cex = 7, col='darkgreen')
points(T[myData$A.klorofylli..µg.l.<lower.lim,1], T[myData$A.klorofyll
i..µg.l.<lower.lim,2], pch='.', cex = 7, col='red')
points(T[myData$A.klorofylli..µg.l.>upper.lim,1], T[myData$A.klorofyll
i..µg.l.>upper.lim,2], pch='.', cex = 7, col='blue')
lower.lim <- 0
upper.lim <- 0.8
points(T[myData$Sameus..FTU.<upper.lim&myData$Sameus..FTU.>lower.lim,1
], T[myData$Sameus..FTU.<upper.lim&myData$Sameus..FTU.>lower.lim,2], p
ch='x', cex = 1, col='green')
points(T[myData$Sameus..FTU.<lower.lim,1], T[myData$Sameus..FTU.<lower
.lim,2], pch='x', cex = 1, col='pink')
```

```
points(T[myData$Sameus..FTU.>upper.lim,1], T[myData$Sameus..FTU.>upper
.lim,2], pch='x', cex = 1, col='brown')


plot(T[,1],T[,2], pch='.')

lower.lim <- 4.75
upper.lim <- 16
points(T[myData$Lämpötila...C.<upper.lim&myData$Lämpötila...C.>lower.l
im,1], T[myData$Lämpötila...C.<upper.lim&myData$Lämpötila...C.>lower.l
im,2], pch='.', cex = 7, col='darkgreen')
points(T[myData$Lämpötila...C.<lower.lim,1], T[myData$Lämpötila...C.<l
ower.lim,2], pch='.', cex = 7, col='red')
points(T[myData$Lämpötila...C.>upper.lim,1], T[myData$Lämpötila...C.>u
pper.lim,2], pch='.', cex = 7, col='blue')
points(T[,1],T[,2], pch='.')

plot(T[,1],T[,2], pch='.')

lower.lim <- 4.7
upper.lim <- 16
points(T[myData$Lämpötila...C.<upper.lim&myData$Lämpötila...C.>lower.l
im,1], T[myData$Lämpötila...C.<upper.lim&myData$Lämpötila...C.>lower.l
im,2], pch='.', cex = 7, col=rgb(0,1,0,.5))
points(T[myData$Lämpötila...C.<lower.lim,1], T[myData$Lämpötila...C.<l
ower.lim,2], pch='.', cex = 7, col=rgb(0,0,1,.5))
points(T[myData$Lämpötila...C.>upper.lim,1], T[myData$Lämpötila...C.>u
pper.lim,2], pch='.', cex = 7, col=rgb(1,0,0,.5))
points(T[,1],T[,2], pch='.')

plot(T[,1],T[,2], pch='.')

lower.lim <- summary(myData$Lämpötila...C.)[2]
upper.lim <- summary(myData$Lämpötila...C.)[5]
points(T[myData$Lämpötila...C.<upper.lim&myData$Lämpötila...C.>lower.l
im,1], T[myData$Lämpötila...C.<upper.lim&myData$Lämpötila...C.>lower.l
im,2], pch='.', cex = 7, col=rgb(0,1,0,.5))
points(T[myData$Lämpötila...C.<lower.lim,1], T[myData$Lämpötila...C.<l
ower.lim,2], pch='.', cex = 7, col=rgb(0,0,1,.5))
points(T[myData$Lämpötila...C.>upper.lim,1], T[myData$Lämpötila...C.>u
pper.lim,2], pch='.', cex = 7, col=rgb(1,0,0,.5))
points(T[,1],T[,2], pch='.')

plot(T[,1],T[,2], pch='.')

lower.lim <- summary(myData$pH....)[2]
upper.lim <- summary(myData$pH....)[5]
points(T[myData$pH....<upper.lim&myData$pH....>lower.lim,1], T[myData$
pH....<upper.lim&myData$pH....>lower.lim,2], pch='.', cex = 7, col=rgb
(0,1,0,.5))
points(T[myData$pH....<lower.lim,1], T[myData$pH....<lower.lim,2], pch
='.', cex = 7, col=rgb(0,0,1,.5))
points(T[myData$pH....>upper.lim,1], T[myData$pH....>upper.lim,2], pch
='.', cex = 7, col=rgb(1,0,0,.5))
points(T[,1],T[,2], pch='.')
summary(myData$pH....)[5]
```

```
plot(T[,1],T[,2], pch='.')

lower.lim <- summary(myData$Johtokyky..µS.cm.)[2]
upper.lim <- summary(myData$Johtokyky..µS.cm.)[5]
points(T[myData$Johtokyky..µS.cm.<upper.lim&myData$Johtokyky..µS.cm.>l
ower.lim,1], T[myData$Johtokyky..µS.cm.<upper.lim&myData$Johtokyky..µS
.cm.>lower.lim,2], pch='.', cex = 7, col=rgb(0,1,0,.5))
points(T[myData$Johtokyky..µS.cm.<lower.lim,1], T[myData$Johtokyky..µS
.cm.<lower.lim,2], pch='.', cex = 7, col=rgb(0,0,1,.5))
points(T[myData$Johtokyky..µS.cm.>upper.lim,1], T[myData$Johtokyky..µS
.cm.>upper.lim,2], pch='.', cex = 7, col=rgb(1,0,0,.5))
points(T[,1],T[,2], pch='.')

library(chemometrics)
library(mvoutlier)
library(robustbase)

C_MCD <- covMcd(my.Data.Av[2:7], cor=TRUE)
X.rpc <- princomp(my.Data.Av[2:7],covmat = C_MCD,cor=TRUE)
#res <- pcaDiagplot(my.Data.Av[2:7], X.rpc, a=2)

T.rob <- X.rpc$scores
P.rob <- X.rpc$loadings
T.rob.df <- as.data.frame(T.rob)
#frame()

plot(T.rob.df$Comp.1 ,T.rob.df$Comp.2, pch='.')


smoothed.pc1 = lowess(T.rob[,1],f=.01)
smoothed.pc2 = lowess(T.rob[,2],f=.01)
#lines(smoothed.pc1,col='red')
plot(T.rob[,1],T.rob[,2], pch=16,cex=1,
     col=TempCol[Temp_0_to_100], ylim=c(-26,16),xlim=c(-5,3),
     xaxt='n', ann=FALSE,yaxt='n')
lines(smoothed.pc1$y,smoothed.pc2$y, col='yellow',lwd=2)
par(new=TRUE)
biplot(T.rob[,1:2],P.rob[,1:2], xlabs = rep('.',length(T[,1])),
       col=c('black','yellow4'),ylim=c(-26,16),xlim=c(-5,3))

x = c(0, 10, 10, 0)
y = c(0, 10, 10, 0)
legend.gradient(pnts=cbind(x = x - 150, y = y - 30),
                cols = TempCol, title = "Temperature", limits = "")

plot(T[,1],T[,2], pch='.')


lower.lim <- summary(my.Data.Av$pH....)[2]
upper.lim <- summary(my.Data.Av$pH....)[5]
points(T.rob.df[my.Data.Av$pH....<upper.lim&my.Data.Av$pH....>lower.li
m,1], T.rob.df[my.Data.Av$pH....<upper.lim&my.Data.Av$pH....>lower.lim
,2], pch='.', cex = 7, col=rgb(0,1,0,.5))
points(T.rob.df[my.Data.Av$pH....<lower.lim,1], T.rob.df[my.Data.Av$pH
....<lower.lim,2], pch='.', cex = 7, col=rgb(0,0,1,.5))
points(T.rob.df[my.Data.Av$pH....>upper.lim,1], T.rob.df[my.Data.Av$pH
....>upper.lim,2], pch='.', cex = 7, col=rgb(1,0,0,.5))
points(T.rob.df[,1],T.rob.df[,2], pch='.')
```

```
res <- pcaCV(my.Data.Av[2:7], segments = 4, repl = 100)
res2 <- pcaVarexpl(my.Data.Av[2:7], a=2)
X_fa <- factanal(my.Data.Av[2:7],factors = 2,rotation = "varimax", sco
res="regression")

P_fa <- X_fa$loadings
T_fa <- X_fa$scores

X_dist <- dist(my.Data.Av[2:7])
X_cluster <- hclust(X_dist)
plot(X_cluster)

library(som)
Xs <- scale(my.Data.Av[2:7])
Xn <- Xs/sqrt(apply(Xs^2,1,sum))
X_SOM <- som(Xn,xdim = 4,ydim = 4)
#plotsom(X_SOM,grp,type = "num")
#plotsom(X_SOM,grp,type = "bar")

library(MASS)
X_dist <- dist(scale(my.Data.Av[2:7]))
sam1 <- isoMDS(X_dist, p=1)
plot(sam1$points)
```

Source code (Time Series)

```
#set working directory
setwd("~/Thesis")
library("TTR")
#load data from CSV file
myData <- read.csv2("data_en.CSV")

## Function to clean the column names
cleanColumnName <- function(columnName){
  columnName <- gsub("\\s", " ", trimws(gsub("."," ",columnName,fixed
= TRUE)))
  return(columnName)
}

## Fuunction to extract data over a time frame (in days)
extractDataOverDate <- function(dataFrame, start, end, include.start=T
RUE, include.end=TRUE){
  if(include.start){
    dataFrame <- dataFrame[dataFrame$ellapsedTime>=start,]
  } else {
    dataFrame <- dataFrame[dataFrame$ellapsedTime>start,]
  }
  if(include.end){
    dataFrame <- dataFrame[dataFrame$ellapsedTime<=end,]
  } else{
    dataFrame <- dataFrame[dataFrame$ellapsedTime<end,]
  }
  return(dataFrame)
}

timeWindow <- function(dataFrame, columnName, startingTime = 0, period
= 5,
                       color = 'black', linewdt = 1, gtype='l',
                       new.name = NA, loess.coef = NA){


  #convert entry times to R time objects
  #format is %d.%m.%Y %H:%M
  times <- strptime(dataFrame$Date.and.time, format = "%d.%m.%Y %H:%M"
, tz="EET")

  #convert into unix time (aka. POSIXct)
  #and chose the starting point (in this case 3 because the first read
ing making sense is #3)
  initialTime <- as.numeric(times[3])
  print("initial time")
  print(initialTime)

  #convert all times into relative times from the starting time
  #this is equivalent to seconds ellapsed since beginning of the sampl
ing
  times <- (as.numeric(times) - initialTime)

  #times converted in days
  times <- times/(24*60*60)

  #add an ellapsedTime column to the dataframe
```

```
  dataFrame$ellapsedTime <- times
  #remove any entries before the starting point
  tempData <<- dataFrame[dataFrame$ellapsedTime>=startingTime,]

  #remove 'na' values
  tempData <<- tempData[!is.na(tempData[[columnName]]),]

  print(summary(tempData))

  upperLimit <- max(tempData[[columnName]])+0.1*(max(tempData[[columnN
ame]])-min(tempData[[columnName]]))
  lowerLimit <- min(tempData[[columnName]])-0.1*(max(tempData[[columnN
ame]])-min(tempData[[columnName]]))

  print(columnName)
  print(upperLimit)
  print(lowerLimit)

  tempData <<- extractDataOverDate(tempData, startingTime,period, TRUE
, FALSE)

  yaxisname = cleanColumnName(columnName)

  if(is.na(new.name)) new.name<-yaxisname

  plot(tempData$ellapsedTime, tempData[[columnName]],
       lwd=linewdt, type = gtype, col=color,
       xlab="Days", ylab=new.name,
       #xlim=c(50,55),
       ylim=c(lowerLimit, upperLimit))

  if(!is.na(loess.coef)){
    #emperature.smooth.model <- loess(temperature~time_mean,
    #                                 data=myDF.temperature<-data.fram
e(time_mean=my.Data.Av.Scaled$time_mean,
    #
temperature=my.Data.Av.Scaled$temperature),
    #                                 span=.03)

    my.Data.Av.Scaled.Smoothed$temperature <- predict(temperature.smoo
th.model, data=my.Data.Av.Scaled)

  }

  i <- period
  j<- i+period
  k <- max(dataFrame$ellapsedTime)
  while(j<k){
    tempData <<- extractDataOverDate(dataFrame, i,j, TRUE, FALSE)
    if(gtype == 'l') lines(tempData$ellapsedTime-i, tempData[[columnNa
me]], col=color,lwd=linewdt)
    else if(gtype == 'p')points(tempData$ellapsedTime-i, tempData[[col
umnName]], col=color,lwd=linewdt)
    i<-j
    j<-i+period
  }
}
```

```
timeWindow(myData, "pH....", 0, 7,rgb(0,0,0,0.25), 1.5, new.name = 'pH
')
timeWindow(myData, "Lämpötila...C.", 0, 7, rgb(0,0,0,0.25), 1.5, 'l',
new.name = 'Temperature (°C)')
#myData$Happi..mg.l.
timeWindow(myData, "Sameus..FTU.", 0, 1e20,rgb(0,0,0,0.5), 1.5, 'l', n
ew.name = 'Turbidity (FTU)')
timeWindow(myData, "Happi..mg.l.", 0, 7,rgb(0,0,0,0.25), 1.5, 'l', new
.name = 'Dissolved Oxygen (mg/l)')
timeWindow(myData, "Happi..mg.l.", 0, 2,rgb(0,0,0,0.25), 1.5, 'l', new
.name = 'Dissolved Oxygen (mg/l)')

timeWindow(myData, "pH....", 0, 30.5,rgb(0,0,0,0.25), 1.5, new.name =
'pH')

#timeWindow(myData, , 0, 30.5,rgb(0,0,0,0.25), 1.5, new.name = 'pH')



tempData <<- myData[ , !(names(myData) %in% c("X", "Date.and.time", "A
kku..V."))]
for(index in c(1:length(colnames(tempData))))colnames(tempData)[index]
<- cleanColumnName(colnames(tempData)[index])

tempData<-tempData[complete.cases(tempData),] #remove NA's
tempData<-tempData[tempData[[7]]!=0,] #remove zeros (generates -inf) -
- bad idea?
tempData<-tempData[tempData[[4]]!=0,]
```