DATA MAGIC:

USING R AND R MARKDOWN FOR WATER QUALITY DATA SCIENCE

Report to the
Water Research Commission

by

S Markham[1], LZ Coetzee[2], D Trollip[3], K Hodgson[3], T White[1] and AD Ceronio [2]

[1]Marquis and Lord
[2]CSVWater Consulting Engineers
[3]Umgeni Water

WRC Report No. 2730/1/19
ISBN 978-0-6392-0079-8

September 2019

Obtainable from
Water Research Commission
Private Bag X03
GEZINA, 0031

orders@wrc.org.za or download from www.wrc.org.za

EXECUTIVE SUMMARY

BACKGROUND

Most water utilities have enormous databases of water quality data which has been used for operational and compliance monitoring. Although this data is often used only once for a pass/fail (compliance) result there is information in this dataset which has not been utilised and can be of enormous value to the water utility, if explored. However, exploring such a large database becomes problematic using our conventional spreadsheets. These large datasets encompass millions of results and hundreds of sample points, making manipulation of such a dataset very difficult in a spreadsheet. It therefore becomes necessary to change our current practices if we as water scientists want to "squeeze the juice" from our data. There are now programs which are able to deal with such large datasets easily but these programs require a particular set of skills which includes coding, statistical knowledge, analytical thinking, creativity, mathematical skills and a brain which is geared to the business at hand; in this case the water business. Data is just data until it is visualised and becomes meaningful to the user.

With the use of R, the author/user has complete control and reproducibility functionality within the R Markdown workspace and it is here where the visualisation of data in the form of graphics and the interpretation of these graphics can all be produced without separation – a true marriage of science, data, visualisation and statistics. This software is able to allow the user extensive freedom and unlimited packages to service every need in terms of visualisation of data. And just like the move from a dial tone telephone to the smartphone or the film to the digital camera, data visualisation needs to change, develop and move forward and this is never a comfortable process, albeit rewarding in the end.

RATIONALE

Water Service Providers (WSPs) generate large amounts of water quality data which is generally used for quality control and quality assurance purposes on the final product. However, this data may contain some information which is not immediately apparent from the pass/fail type of test but is available to the enquiring mind. Therefore, the open source R language (free to download and use by anyone) with its packages and statistical computing and graphics along with a myriad of packages designed by the users for the users in the worlds of science, environment, marketing, finance, health and research, has been chosen as the most appropriate tool for the exploration of the large datasets of water quality data used in this report. Water quality data for a period of twenty years was obtained from one of the largest WSPs in the country and a nearby municipality. This data pool consists of several million individual test results from more than 650 locations.

OBJECTIVES AND AIMS

With this software, the aim was to introduce the water sector in South Africa to the power of R software and to do this by using R, R Markdown and R Studio to achieve the following aims:

1. What is the statistical relevance of rare microbiological or chemical exceedances and the associated water quality data (free and total chlorine/ pH/ turbidity, etc.) in treated water samples?
2. What can appropriate statistical analysis techniques tell us about the relationships between out of range microbiological data (total coliforms, Heterotrophic plate counts, *E. coli*.)
3. Develop a decision support tool in the form of this manual to assist with the investigation and verification of future microbiological data

4. Find and develop appropriate data presentation and visualisation techniques to allow statistical information to be visualised in a way that can easily be understood by technical and non-technical staff alike.

METHODOLOGY

The R software has been recognised by the Royal Statistical Society as one of the key breakthroughs in statistics in the last 100 years. R is the most commonly used statistical analysis package, applied by organisations such as Google and NASA but also the insurance industry, medical profession, consultants and academia. Its extensive use by PhD students has helped disseminate the use of R through industry. New advances in statistical analysis have nearly all been made using open source software or are quickly available for use through R; usually through an R package. The strength of R comes from the world-wide community that maintain, use and develop it; this community has grown exponentially during the last 5 years. R is an open source free software environment for statistical computing and graphics and it has an integrated software suite which allows for data manipulation, calculations and graphical display.

RESULTS AND DISCUSSION

Twenty years of water quality data was provided from two different sources. Data from two different sources was tidied and some data management issues were discovered which require a laboratory data management policy (in progress) to be formulated to describe the procedures for the following:

- reporting limit policies
- data field identifier issues
- rogue data rejection protocols
- sampling point location identifiers
- comma inclusive systematic naming
- data unit inconsistencies.

Once the data was tidy the water quality database had 1,712,175 records and covers the period from 1 January 1990 to 30 April 2017. Two questions were asked of the data which was to assist the user by introducing them to the R software using R Markdown and R Studio. The methodology used to get to these answers showcases the ability of the R packages to generate answers from querying this huge dataset. The workshops that were held were extremely valuable for all participants. It enabled the participants to see the way in which the code works and that the data product generated by the code can be used repeatedly on different datasets to produce the data visualisation that is needed for making strategic decisions based on the data provided.

CONCLUSIONS

"I was taught the way of progress is neither swift nor easy"
-Marie Curie (1867-1934), French physicist and two-time winner of the Nobel Prize

The ability to look beyond our current data manipulation and visualisation techniques using the faithful Excel spreadsheet, does not come easily and indeed may be quite a paradigm shift for many participants in the water sector. When learning R, it should be borne in mind that this requires discipline and tenacity, perseverance and the ability to search for answers when one hits a brick wall. However, this language provides real progress and a different way of processing any sized dataset until the visualisation of the dataset meets with the data and water scientist's expectations. R is not easy to learn but the rewards are great.

RECOMMENDATIONS FOR FUTURE RESEARCH

In the area of data management within the laboratory, during the course of the data tidying process the following issues were identified during the consultative process:

- There is a need for standardisation of field identifiers between study datasets.
- Result reporting policy influence on the data requires additional investigation prior to applying any further interpretation to the results relative to making deductions about water quality.
- The presence of “rogue” data points or outliers causes compression of graphical apices and limits the visual assessment of time series data. A protocol was agreed upon for addressing this issue for this project which limits the risk of losing relevant information about isolated genuine events. It is also relevant to anyone who attempts similar projects.
- Should some sample locations be absence from the first assessment, the lack of data should be investigated to ensure that the information is available in the database and is presented in future iterations.
- Consistency in record naming of different analysis to allow for easier data manipulation.
- It is also strongly recommended that a team is constituted to investigate the value of and requirement for a Laboratory Data Management Policy to guide decision-making regarding laboratory data. This may include, amongst others, issues related to reporting limit changes, documentation of analytical method changes, issues related to decimal point reporting, data validation, etc. It is recommended that this team is led by the Quality Team from Laboratory Services, but staff from other water company departments will also need to contribute.
- In future, geo spatial mapping of sample points and water quality can be investigated.
- Interactive consultation with water service providers in South Africa to generate a water quality data science community.

ACKNOWLEDGEMENTS

The authors would like to thank Umgeni Water and eThekwini Metropolitan Municipality who shared their monitoring data; and the Reference Group for their anticipated guidance and technical input.

Reference Group	Affiliation
Dr Nonhlanhla Kalebaila	Water Research Commission (Chairperson)
Ms Charmaine Khanyile	Water Research Commission (Project Administrator)
Mr Kenneth Dukuza	University of Pretoria
Dr Michael Silberbauer	Department of Water and Sanitation (DWS)
Mrs Siobhan Jackson	eThekwini Metropolitan Municipality
Mr Sibongile Maqubela	eThekwini Metropolitan Municipality
Mr Henry Mwambi	University of KwaZulu-Natal
Mr Oliver Bodhlyerao	University of KwaZulu-Natal
Mr Marc Arenstein	Consultant (Umgeni water)
Ms Mpharu Hloyi	City of Cape Town Metropolitan Municipality

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF CODING EXAMPLES

This page was intentionally left blank

CHAPTER 1: INTRODUCTION TO DATA SCIENCE

1.1 INTRODUCTION

Data science is a multidisciplinary blend of data inference, algorithm development, and technology in order to solve analytically complex problems.[1] At the core of this is data. Lots of information which is stored somewhere and much value to be found by mining it and using this data in creative ways to generate business value. Data science is about diving in at a granular level to mine and understand trends and relationships within the dataset that are not immediately apparent and which may enable the user to make better decisions based on the hidden insights. For example:

- Netflix data mines movie viewing patterns to understand what drives user interest, and uses that to make decisions on which Netflix original series to produce.
- Target identifies what are major customer segments within its base and the unique shopping behaviours within those segments, which helps to guide messaging to different market audiences.
- Proctor & Gamble utilizes time series models to more clearly understand future demand, which help plan for production levels more optimally.

This process starts with data exploration and a good data scientist becomes a data detective to investigate patterns or trends within the dataset. This requires a significant amount of analytical creativity from the data scientist!

From this process a data product can be produced which is an asset to the business. Designed specifically on the data that the business generates, it utilises data as the input and processes that data to return algorithmically generated results. This can include time series forecasting, segmentation analysis and synthetic control experiments to allow the user to view what the data is really saying. A classic example of a data product is a recommendation engine, which ingests user data, and makes personalized recommendations based on that data. Some examples of data products include:

- Amazon's recommendation engines suggest items for you to buy, determined by their algorithms. Netflix recommends movies to you. Spotify recommends music to you.
- Gmail's spam filter is data product – an algorithm behind the scenes processes incoming mail and determines if a message is junk or not.
- Tesla's self-driving cars require computer vision which is also a data product – machine learning algorithms are able to recognize traffic lights, other cars on the road, pedestrians, etc.

1.2 WATER UTILITIES AND DATA

Water Utilities are known for being data rich organisations. Water Utilities collect raw water and process (treatment) data, chemical dosage rate data and flow records, operational monitoring data, treatment plant data, compliance data for final water reservoirs and distribution systems, water quality at the final contact tank, distribution reservoirs, networks and consumer complaint information. Many of the larger water utilities have online instrumentation such as Supervisory Control and Data Acquisition (SCADA) which collects information on flow and quality parameters which is used by the process control staff for operational purposes and quality assurance and control.

[1] https://datajobs.com/what-is-data-science

In the South African drinking water sector, there were over 1.9 million drinking water analyses submitted to the Department of Water and Sanitation in 2011. This formed part of the required submissions by Water Service Authorities (WSAs) for the Department of Water and Sanitation as part of the Blue Drop Certification requirements (2012 Blue Drop Report). Much of this data is captured from a laboratory report to an Excel or csv format spreadsheet while the bigger water services authorities make use of Laboratory Information Management Systems (LIMS) to submit data. It is also important to ensure that this water quality data is generated accurately by the laboratory. WSAs' spend significant portions of their budgets on proving that the drinking water complies with the SANS 241:2015 requirements including laboratory services and staff, transport costs, quality control and assurance in the laboratory, transport for sampling runs, sample transit integrity and troubleshooting.

Therefore WSAs' are naturally suitable for new developments in the analysis of data because there is so much data available and much of the competency required is already available in the analytical minds of the scientists and engineers who work in such organisations.

1.3 NEED FOR A NEW WATER DATA ANALYSIS APPROACH

Our world has changed with the dawn of the Information Age. The way that we exchange information today has changed significantly within the last few decades. Instead of letters we use e-mail, instead of meetings we use Skype, photographs are made of pixels not paper and we send large chunks of information almost instantaneously via the internet to people on the other side of the world. The water sector is becoming increasingly information intensive. However, the way we handle our data in the water utility sector has not really kept up with the way the world now shares information. The vast amounts of data that we collect are kept on a LIMS system but we fail to extract sufficient value from this data. Although the large volumes of water data that we generate is utilised by the water utilities to prove that the water is safe to drink, there is more value within the data pool that has not been extracted.

Much of this data is only used once or twice – to determine compliance to the internal or external limits required before being archived. Water data is also used for reporting purposes by the WSA to the water board, managers and to the consumers. In fact, a survey conducted in 2015 in the USA indicated that only 10% of the available data is analysed to create value. The remaining 90% of data is not utilised (Prevos, 2017). The situation is in all probability very similar in South Africa.

This data can be used opportunistically to extract value from information which has previously been collected for another purpose – in other words allowing us to "squeeze the juice" from our datasets. This is particularly important when failures occur which are not necessarily high impact failures but these events are a source of information that is not investigated sufficiently.

However, communication of the performance of a water systems performance is complex especially as the performance depends on multiple variables which are often difficult to visualise correctly. Also, long terms trends are often neglected especially since the Excel spreadsheet is able to handle a finite number of analyses and is not suitable for "big data" Water Utilities are only starting to investigate the potential utilisation of data science techniques to investigate, visualise and predict outcomes using techniques which are not standard practice or Excel based. Some work has been done in Australia where Coliban Water has started using data science techniques to visualise and generate data for its board members and its consumers (Prevos, 2015). In Figure 1-1, the data from various data sources is used to graphically represent the water quality from catchment to consumer based on various important factors (catchment protection, barrier effectiveness, network protection, regulatory compliance, and customer perception) to provide a useful tool to determine the risk areas in the water supply network.

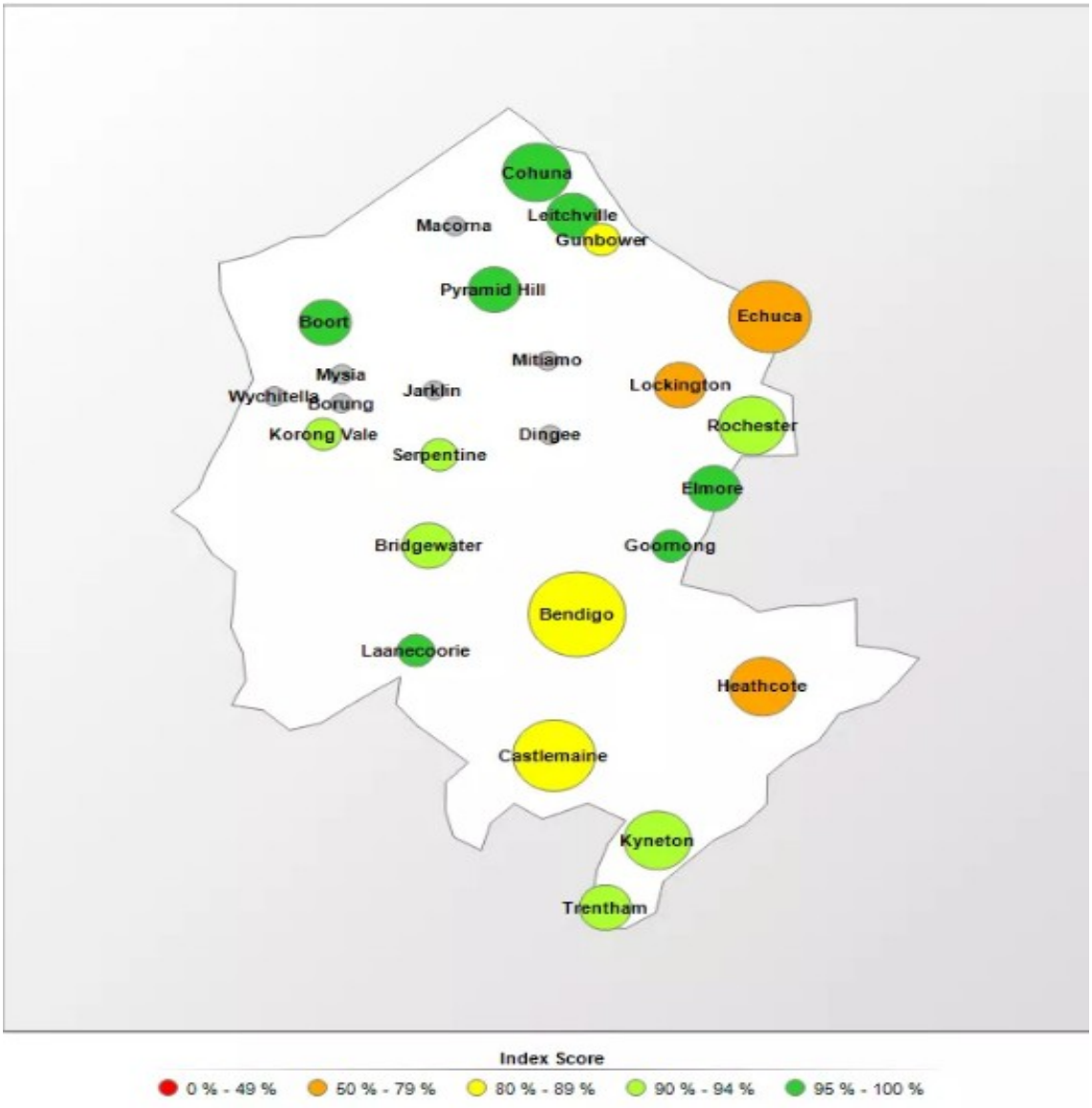

Figure 1-1: The Water System Performance Index from Coliban Water, Australia. Source: Prevos (2015).

Many of the water utilities yearn for a more systematic approach to creating value from data. As such our water utilities have not moved into the data science realm like Google, Amazon and NASA where data science is already part of the way the enterprise is being operated. Data science is a new field for water utilities and it has not been applied in South Africa before. Datasets from water service providers, can be used to introduce the water sector to data science. Data science allows us to use techniques to recognise patterns and make predictions to allow us to find new and different ways of gaining knowledge of our water systems. Much more than previously, the ascent of data science and “big data” into the world of our water business will allow us to grow in ways that we cannot imagine.

1.4 RATIONALE AND OBJECTIVES OF THIS STUDY

WSPs generate large amounts of water quality data which is generally used for quality control and quality assurance purposes on the final product. However, this data may contain some information which is not immediately apparent from the pass/fail type of test but is available to the enquiring mind. Therefore, the open source R language (free to download and use by anyone) with its packages and statistical computing and graphics along with a myriad of packages designed by the users for the users in the worlds of science, environment, marketing, finance, health and research, has been chosen as the most appropriate tool for the exploration of the large datasets of water quality data used in this report. Water quality data for a period of twenty years was obtained from one of the largest WSPs in the country and a nearby municipality. This data pool consists of several million individual test results from more than 650 locations.

With this software, the aim was to introduce the water sector in South Africa to the power of R software and to do this by using R, R Markdown and R Studio to achieve the following aims:

1. What is the statistical relevance of rare microbiological or chemical exceedances and the associated water quality data (free and total chlorine/ pH/ turbidity, etc.) in treated water samples?
2. What can appropriate statistical analysis techniques tell us about the relationships between out of range microbiological data (total coliforms, Heterotrophic plate counts, *E. coli*.)
3. Develop a decision support tool in the form of this manual to assist with the investigation and verification of future microbiological data
4. Find and develop appropriate data presentation and visualisation techniques to allow statistical information to be visualised in a way that can easily be understood by technical and non-technical staff alike.

1.5 SO WHY IS R THE RIGHT CHOICE?

- The R software is open source – which means it is free and that anyone is able to use it. As part of the perks of using R it has a number of functions available such as data manipulation statistics and graphics for visualisation of the data.
- R is also available for all types of hardware and software – it can be used on almost any machine; Windows, Linux and Mac.
- R is extendable – in other words – it can be developed by R Users and there is a large population of R users who extend the software and these improvements are generally freely available There is a large collection of freely available data packages which can be added on; to provide enormous versatility in applying data science.
- R users have a growing community of researchers and data scientists who are engaged via various platforms. R users participate in social networks such as Twitter (www.twitter.com/search/rstats) and Use R! Conferences are held every year.

CHAPTER 2: DATA SCIENCE USING R

2.1 INTRODUCTION

The approach (methodology and practice) to data analysis has been changed dramatically in recent years with the development of open source software; notably R and its many packages. The R software has been recognised by the Royal Statistical Society as one of the key breakthroughs in statistics in the last 100 years. R is the most commonly used statistical analysis package, applied by organisations such as Google and NASA but also the insurance industry, medical profession, consultants & academia. Its extensive use by PhD students has helped disseminate the use of R through industry.

New advances in statistical analysis have nearly all been made using open source software or are quickly available for use through R; usually through an R package. The strength of R comes from the world-wide community that maintain, use and develop it; this community has grown exponentially during the last 5 years. R is an open source free software environment for statistical computing and graphics. It has an integrated software suite which allows for data manipulation, calculations and graphical display. It includes:

- An effective data handling and storage facility
- A suite of operators for calculations on arrays, in particular matrices
- A large collection of intermediate tools for data analysis
- Graphical facilities for data analysis and display either on screen or as hardcopy
- A well-developed simple and effective programming language including user defined functions

R Open Source Software (and its associated Packages and Graphical User Interface (GUIs)) can help with the assessment of routine water quality monitoring and interpretation. Moreover, it offers distinct advantages over proprietary software, namely:

- R and associated software are free to use; it is supported by a huge online community. Not only will this online community help teach users how to use R it will more than likely have solved or have a method of solving an existing statistical determination problem. Proprietary software can also have online communities but will be limited. When compared to that of R it will limit the statistical analysis available.
- Proprietary software is costly to purchase: R is freely available to everyone in the organisation proprietary software would be limited to a few licences
- Proprietary software will also require training in it use but the limited licences make an organisation vulnerable to change of staff (i.e. through promotion)
- New employees are likely to have had exposer/experience of R from university teaching: this helps develop and further the skill and approach to data analysis within the organisation.
- R can access proprietary databases (usually directly) and then apply statistical analysis and produce a report directly into Word or PDF.

The initial use of R and understanding of how it can be applied usually needs to be kick started in an organisation. This research project will, in part, address this aspect of the uptake of R in the water industry.
A significant motivation for the use of R and related open source software is the cost of proprietary packages, for example SPSS® and SAS®: SPSS Standard 1,210 USD per user; SPSS Professional 5,400 USD per user; and, SAS starting price of 10,000 USD per user. The use of proprietary packages in any organisation is a clear cost inhibition to the uptake of techniques which are becoming an essential tool in extracting the maximum value from legacy data.

2.2 A QUICK INTRODUCTION TO R AND R RESOURCES

2.2.1 R

R is an open-source programming language and software environment for statistical computing and graphics that is supported by the R Foundation for Statistical Computing (R Core Team, 2018). The R language is widely used among statisticians/data scientists for developing statistical software and data analysis. The latest version of R (R is regularly updated) can be downloaded from CRAN, the comprehensive **R** archive network http://www.cran.r-project.org/. CRAN consists of a set of mirror servers distributed around the world and is used to distribute R and R packages. The mirror servers are a replica of contents of the main server and are used to reduce the load on one server by having multiple synced copies. You can choose a mirror (server) where you get low latency, this allows for faster downloads. The R project is maintained on, and is available at, the following URLs in South Africa:

- http://r.adu.org.za hosted by the University of Cape Town; and
- http://cran.mirror.ac.za hosted by TENET, Johannesburg

In order to get started using R, you will need to download and install R, RStudio, and a collection of R packages. R Packages are the fundamental units of reproducible R code, and they include reusable functions, the documentation that describes how to use them, and sample data.

2.2.2 RStudio

RStudio is an integrated development environment (IDE) for R, i.e. the platform for using R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management. The latest version of RStudio can be downloaded from:
https://www.rstudio.com/products/rstudio/#Desktop

2.2.3 R Packages

Currently, the CRAN package repository features 13707 available packages. Much of the ability of R is derived from its packages; these add real value to the data science being applied. The packages are developed by a wide variety of contributors and are most commonly provided as a General Public Licence (GPL). A list of all the available packages is available at https://cloud.r-project.org/. Here are some examples of R packages;

- readxl makes it easy to get data out of Excel and into R http://r4ds.had.co.nz/data-import.html
- tidyverse includes packages that you're likely to use in every day data analyses
- leaflet for interactive maps, https://rstudio.github.io/leaflet/
- StreamMetabolism for sunrise and sunset times (Sefick Jr., 2016)
- gridExtra arranging multiple graphs on a page, https://cran.r-project.org/web/packages/gridExtra/vignettes/arrangeGrob.html
- Bookdown package for citations (Xie, 2018)

2.2.3.1 Example 1 – tidyverse

The **tidyverse** package is a collection of packages useful to data science (e.g. loading/manipulating data and plotting graphs) fully described at https://www.tidyverse.org/ and includes packages such as;

- ggplot2 creating graphics http://ggplot2.tidyverse.org/
- dplyr data manipulation http://r4ds.had.co.nz/transform.html

- tidyr tidy data: each variable is in a column, each observation is a row http://r4ds.had.co.nz/ tidy-data.html
- lubridate handling dates and times including daylight saving https://www.jstatsoft.org/article/view/v040i03/v40i03.pdf

How to apply the **tidyverse** packages is also illustrated in the *R for Data Science Book* by Garrett Grolemund & Hadley Wickham the whole book is published online http://r4ds.had.co.nz/index.html.

2.2.3.2 Example 2 – R Markdown

R Markdown provides an authoring framework for data science. You can use a single R Markdown file to both
- save and execute code
- generate high quality reports that can be shared with an audience

R Markdown documents are fully reproducible and support dozens of static and dynamic output formats. It can transform your analyses into high quality documents, reports, presentations and dashboards. R Markdown is integrated into RStudio, a full introduction to R Markdown can be found: http://R Markdown. rstudio.com/lesson-1.html.

2.3 USING R FOR DATA SCIENCE

2.3.1 Overview of the data science approach

With the modern computing power now available along with advanced algorithms we are able to:
- Identify hidden trends in large datasets
- Leverage trends to make predictions
- Compute the probability of each possible outcome
- Obtain accurate results quickly

These techniques allow us to analyse, visualise and leverage data in very different ways. A schematic representation of the methodology for Data Science is presented in Figure 2-1.

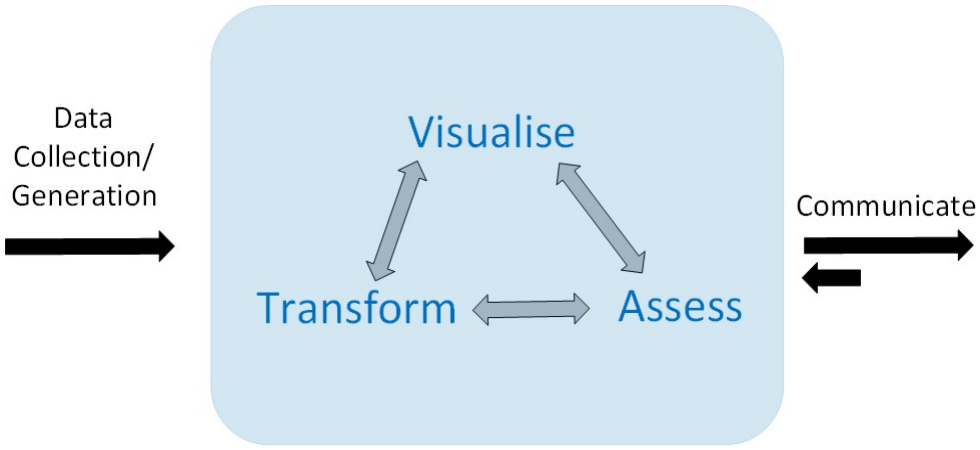

Figure 2-1: Data Science Schematic

Combined with machine learning statistics and mathematics it allows us to recognise patterns within the dataset and enhances its predictive capabilities. These techniques are used in business for example to provide product recommendations systems on online websites to enhance product sales, providing a higher turnover for the seller and showing customised products for willing consumers.

Many of us are guilty of collecting data without always extracting the full value from this data. Although some of us apply our substantive expertise and maths and statistical knowledge in traditional research we do not add machine learning to our skill set – and this is where the powerful field of data science resides. Therefore, data science provides a method for us as humans to acknowledge that our experience and knowledge is limited and coloured by our own experience which influences our decision-making abilities and conclusions that we draw from our data (Figure 2-2).

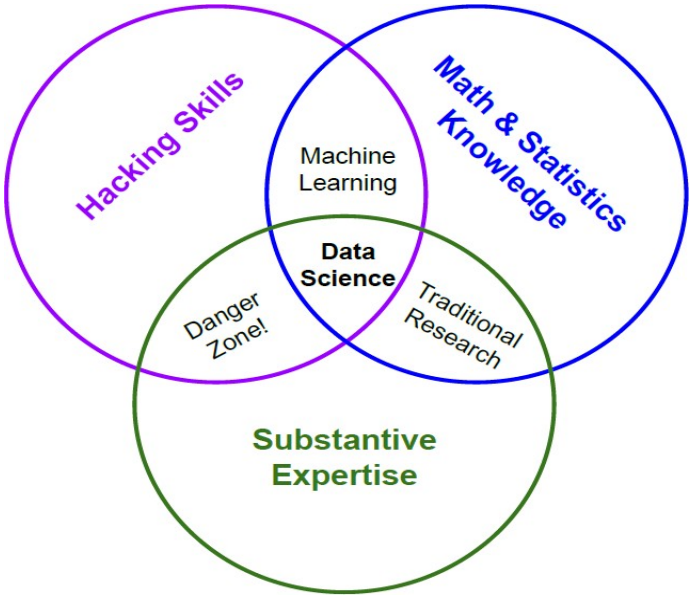

Figure 2-2: Conway's Data Science Venn diagram
(Source: http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram)

2.3.2 Data Management

It is often said that 80% of the time spent on data analysis is spent on the data clean up and preparation process (see Figure 2-3) (Dasu and Johnson; 2003; Wickham, 2014). This step also needs to be repeated frequently when additional data is collected or when a problem comes to light during the course of the data analysis when the analysis shows unexpected errors or outcomes. It may surprise you then to learn that there is an international and South African standard for Data Elements and interchange formats – information interchange – Representation of dates and times (ISO 8601:2004 /SANS 8601:2009). This standard describes the specifications for a numeric representation of information regarding the time of day and the date as well as the format for this information. This becomes important when we are looking at data which has seasonal or circadian rhythms (such as algal growth or seasonal impoundment turnover). Data parsing is composed of checking outlier data, data parsing and missing value imputation (Wickham et al., 2018). Once the data is tidied and there is a standard way of storing the data the cleaning process becomes far easier.

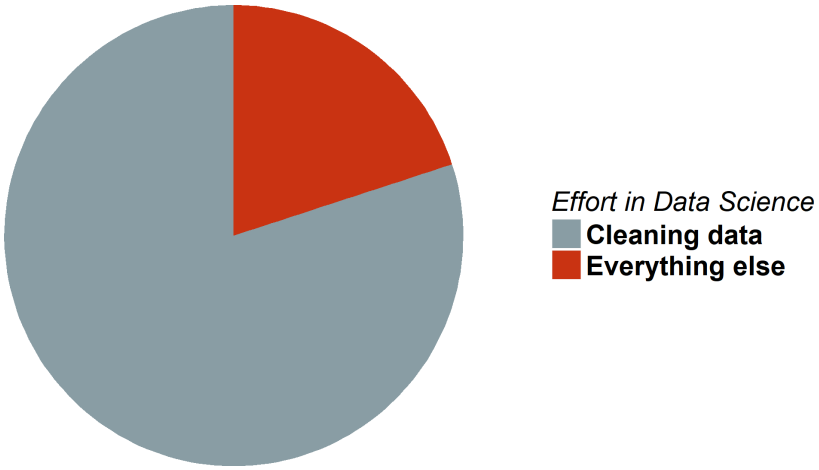

Figure 2-3: Time spent on cleaning data and everything else

2.3.3 What is Tidy Data?

It is no surprise that a key part of Data Science is the data; data is not always in an intelligible form. Table 2-1 shows the World Health Organisation (WHO) data for cases of tuberculosis (TB) recorded throughout the world (http://apps.who.int/gho/data/view.main.57016ALL?lang=en). The data format is messy (untidy) with five variables spread across the columns. If you look at Table 2-2, it highlights the 5 variables spread across the untidy data: Country Code (two letters); Year; Gender; Cases; and, Age.

Table 2-1: WHO data format for cases of tuberculosis

country	year	m014	m1524	m2534	m3544	m4554	m5564	m65	mu	f014
AD	2000	0	0	1	0	0	0	0	—	—
AE	2000	2	4	4	6	5	12	10	—	3
AF	2000	52	228	183	149	129	94	80	—	93
AG	2000	0	0	0	0	0	0	1	—	1
AL	2000	2	19	21	14	24	19	16	—	3
AM	2000	2	152	130	131	63	26	21	—	1
AN	2000	0	0	1	2	0	0	0	—	0
AO	2000	186	999	1003	912	482	312	194	—	247
AR	2000	97	278	594	402	419	368	330	—	121
AS	2000	—	—	—	—	1	1	—	—	—

Table 2-2: The 5 variables across an untidy data

Year Gender Age

country	year	m014	m1524	m2534	m3544	m4554	m5564	m65	mu	f014
AD	2000	0	0	1	0	0	0	0	—	—
AE	2000	2	4	4	6	5	12	10	—	3
AF	2000	52	228	183	149	129	94	80	—	93
AG	2000	0	0	0	0	0	0	1	—	1
AL	2000	2	19	21	14	24	19	16	—	3
AM	2000	2	152	130	131	63	26	21	—	1
AN	2000	0	0	1	2	0	0	0	—	0
AO	2000	186	999	1003	912	482	312	194	—	247
AR	2000	97	278	594	402	419	368	330	—	121
AS	2000	—	—	—	—	1	1	—	—	—

Two letter country code Cases

Tidy Data has a specific format as presented in Figure 2-4.

- every variable is in its own column
- individual observations, or case, is in its own row
- each value goes in a cell

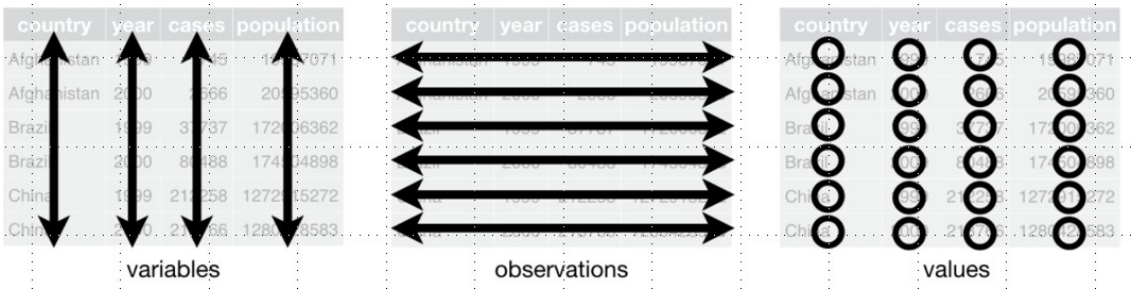

Figure 2-4: Example Tidy Data

Figuring out which is an observation and which is a variable is not quite as easy as one would think! Every value belongs to a variable and an observation. A variable contains all values that measure the same underlying attribute (such as turbidity) and an observation contains all the values measured on the same unit across attributes (such as a sample point) (Wickham, 2014).

The WHO data for cases of tuberculosis is shown in a Tidy format in Table 2-3; the 5 variables are in their own column with each observation for the age/gender/year/country/cases in a row. When the data is in this format it can be visualised more readily; as shown in world map of cases in Figure 2-5.

Table 2-3: WHO TB Tidy Data

country	year	sex	age	cases
AD	2000	m	0-14	0
AD	2000	m	15-24	0
AD	2000	m	25-34	1
AD	2000	m	35-44	0
AD	2000	m	45-54	0
AD	2000	m	55-64	0
AD	2000	m	65+	0
AE	2000	m	0-14	2
AE	2000	m	15-24	4
AE	2000	m	25-34	4
AE	2000	m	35-44	6
AE	2000	m	45-54	5
AE	2000	m	55-64	12
AE	2000	m	65+	10
AE	2000	f	0-14	3

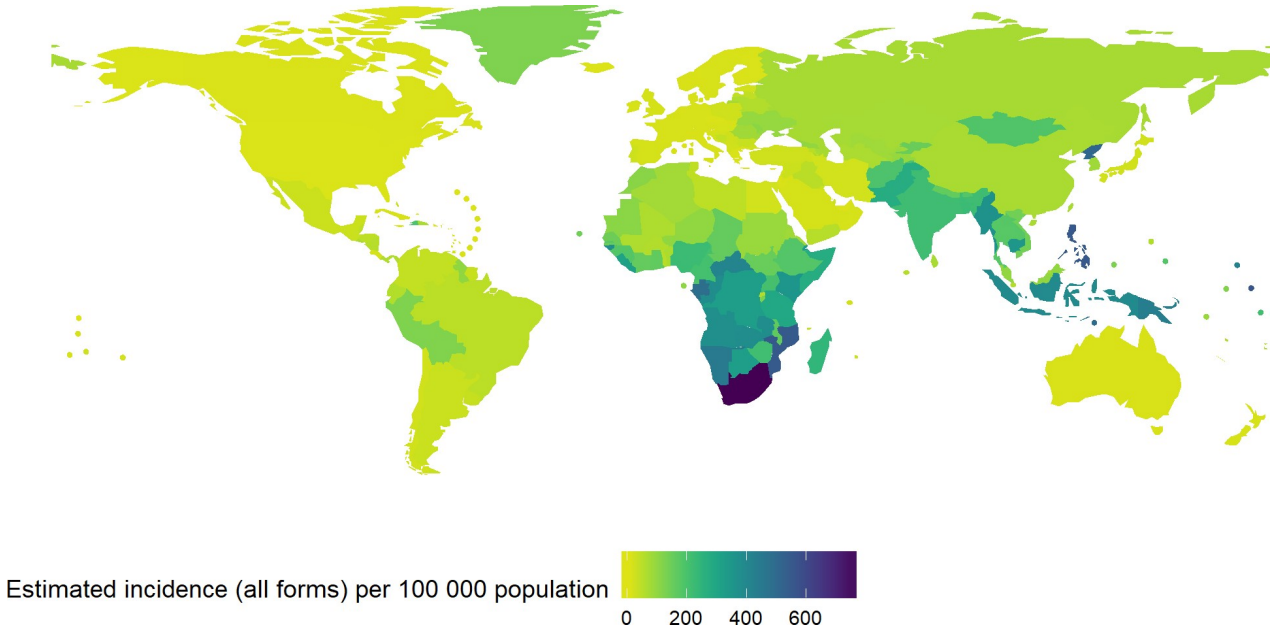

Figure 2-5: World map of TB Cases from Tidy Data

2.3.4 Data Identifier Issues

In a laboratory environment, especially when the database is a large one stretching over several years, one of the most important issues is the correct identification of the data points in the dataset. This is a very important consideration because it allows the data analyst to correct the field identifier (column name) to ensure that all the data is taken into account when the software is applied. For example:

- the equivalence of field identifiers such as pH / PH/ pH @25;
- are the names of parameters the same? e.g. *E coli* versus EC; MPN CFU;
- the incidence of stray parenthesis "," in field identifiers (IUPAC naming) which can hinder transfer file generation as CSVs; and
- data formats (1,2 versus 1.2).

2.3.5 What Additional Data is Desirable?

The R software is able to handle all the data available and therefore it is important that the data is not "sanitised" or restricted by the exclusion of metadata. Some of the data may appear to be unimportant to the dataset but this assumption should not be used to edit the dataset. For example, there may be simple true/false statements available about the location of a raw water sampling point (bank side or bridge collection point). This may be a relevant differentiating factor during data analysis, but if it is not included in the meta data for the sample the relationship, or lack of one, will never be established in the evaluation.

2.3.6 Multiple Data Source Issues

Two (or more) sources of data can generally be combined if there is common field (i.e. a variable that is the same or can be manipulated to be the same). Without a common field, information cannot be combined they are just apples and pears. Date and time are variables that lend themselves to combining data. If more than one data source is utilised (for example LIMS dataset, operational process control data and weather datasets) the compatibility of the datasets entering the data pool needs to be checked and managed carefully to ensure that the datasets are compatible. Adequate processing tools (such as *tidyverse*) should be in place to enable

the examination of chosen relationships. Some of the key issues which will need to be accommodated are included in the nonexclusive list below:

- non-overlapping time frames from multiple sources – for example January to December in set 1 and June to December in set 2 for the same year;
- differing data provision rates – for example weekly water quality data but daily temperature and monthly rain fall;
- differing expressions of spatial coordinates (GPS reference data) – for example Degrees, Minutes and Seconds or Decimal units;
- differing reporting units – for example atmospheric pressure: Pascal, Bar, or Torr.

Processes and techniques which can be deployed for the management of these issues are centred around the R package *tidyverse*; a collection of R packages designed for data science, all the packages share and underlying philosophy and common application programming interface (API). For comparison and analysis of data from different source databases there should be agreement on:

- the column names
- the units that the sample is measured in
- measurements below detection limits and how we handle these

2.3.7 Is the Data Combination Meaningful?

At this stage, the data should be reviewed to ensure that the time line comparisons and correlations make sense. Some data combinations may be appropriate for one task but not for others. For example, using average monthly rainfall data and four suspended solids results for a river for the same months may show a meaningful correlation. However, using rainfall data for four sampling days in the same month may present a relationship which is masked by the variance in such a small data sub set. Therefore a 'data test'[2] should be done before the optimum "mixing relationship" is utilised in the output report.

It is important that all data combinations used are documented and clearly presented in the output report to maintain the integrity and transparency of the data processing. For example, it may be necessary to restrict the use of data sub sets to those with more than 10 000 results, especially when dealing with long time periods. This would automatically exclude locations that have only been operational for a short period. Such a data use choice must be clearly stated in the report. However, if the goal is to study the relationships between two or more parameters then the inclusion of large and small sub sets may be mathematically relevant. An example of this is the examination of seasonal trends using box plots. These choices should be documented and presented carefully so that the report user is aware that for location "A" the seasonal assessment has been carried out with 10 years of data whereas for location "B" only six years' results are available.

2.3.8 Validation and Applicability of Meta Data

A review and validation of metadata is an important step because certain qualifications applicable to the dataset may add value to the interpretation of the results generated. For example, knowledge of when the disinfection regime at a water treatment works changed from sodium hypochlorite to chloramination is an important factor when reviewing the time series data. Having the ability to differentiate a single time series data sub set with this information by simply changing the marker colour on a scatter plot can highlight the relationships which need to be explored. Similar approaches can be applied to other events of seemingly low

[2] Note: A data test is a series of trial plots using differing combinations of manipulated data to determine the processes which do not mask or distort the presentation.

significance but which may be important when examining long time lines. Examples of such information which should be assimilated are dates of change of:

- laboratory methods;
- limits of detection;
- sample preservative;
- temperature of sample transport;
- rounding policy for reporting;
- sampler;
- sample container (glass v plastic);
- relocation of rain gauge;
- sampling technique (e.g. spot sample or auto sampler period average);
- application of daylight-saving time; and
- maintenance schedules or emergency repairs.

The data manager therefore needs to be alert to the potential for these and many other possible meta data 'switches' which could have an influence on the interpretation applied to the final report. In the case of two or more data sources which have different names for a particular analysis (for example *E coli* versus *E. coli*) they can be combined if the date and time variables fall within the same period as long as the user is convinced that the data is the same.

2.3.9 Stoichiometric Units

Biochemical interactions such as disinfection are reliant on the action of a chemical on an active site; therefore, stoichiometric relationships prevail in terms of responses of living organisms to chemical concentration. Conversion of chemical components to molar concentrations will often enhance the visual presentation of correlations and allow more confident interpretations of the relationships examined. The same applies where direct chemical interactions are being studied, for example pH and iron concentrations. In this example, it is important not to double transform the pH data since this is already a value derived from the negative logarithm of the molar hydrogen ion concentration, i.e. a plot of moles of iron against pH would be appropriate.

2.3.10 An Example of Consultation Feed-back Which Lead to Iterations of a First Pass Report

The following is an example of the type of feedback required from the consultative step. A step-change the presentation of coliform data is evident in the time series plot. Such steps should indicate to the data manger that two treatments of seasonal assessments using box plots may be required to assess the impact associated with the step change. Figure 2-6 shows coliform time series for a Water treatment plant (Raw water) case study.

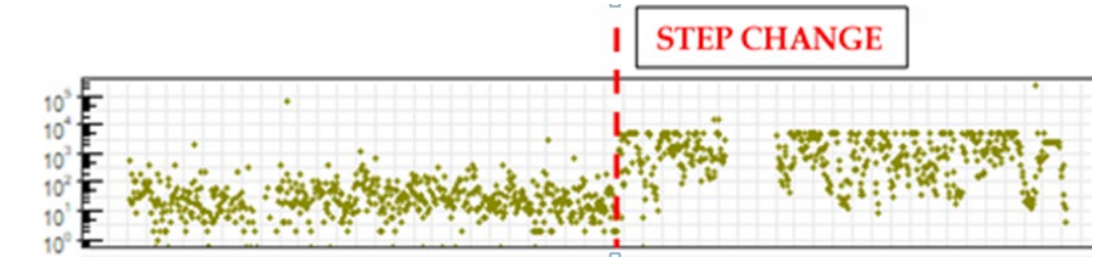

Figure 2-6: Consultative feedback leads to iterations of the initial report

This information needs to be used with caution because of the influence of external variables which might cause a step change in time series data. Before decisions are made with respect to the segregation of data by time blocks it is recommended that a review of available meta data is undertaken to attempt to identify all possible external factors which could relate to the step change. Such a review will assist in confirming the influence of a single external factor.

The type of metadata which could be included, but not exclusively, in the review are:

- Changes in demand on the facility – differences in flow and load.
- Has an additional raw water input been brought on line?
- Has there been a change in sensitivity of the test method?
- Has there been a change in upper test limit reporting policy?
- Have there been any changes in pre-treatment regimes such as the decommissioning of plant?
- It is only when a review of this type is carried out, can the decision to fragment the data be justified and the interpretation of the resultant sub sets correctly documented.

Once completed the first assessment of the type described will inevitably lead to discussions about data management within an organisation to facilitate an easier more streamlined process for future and ongoing use of the data pool.

2.4 ONLINE RESOURCES FOR LEARNING R

This code is not routinely taught in a formal manner and therefore it is up to the user to spend time and resources if required to upskill them. Most of the users are self-taught and it is the usual way to learn this skill. Many practitioners are skilled in one area of focus (finance marketing, etc.) and therefore within the specialists, there are those who particularly focus on a particular set of data which provides certain standard tests and graphics which are predominantly used.

2.4.1 Webinars to Watch

The RStudio website has a large collection of webinars https://www.rstudio.com/resources/webinars/. The following webinars are recommended from those listed under *The Essentials of Data Science*:

- Getting your data into R https://www.rstudio.com/resources/webinars/getting-your-data-into-r/
- What's new with readxl (get data from Excel) https://www.rstudio.com/resources/webinars/ whats-new-with-readxl/
- Data wrangling with R and RStudio https://www.rstudio.com/resources/webinars/data-wrangling-with-r-and-rstudio/
- Getting started with R Markdown https://www.rstudio.com/resources/webinars/getting-started-with-r-markdown/
- The Ecosystem of R Markdown https://www.rstudio.com/resources/webinars/the-ecosystem-of-r-markdown/

There are numerous videos on YouTube, of varying quality, covering all aspects of R/RStudio/R Markdown.

2.4.2 Books and other online help documents

- In addition to the R for Data Science Book http://r4ds.had.co.nz/index.html, there are many other relevant books (whole books) available at https://bookdown.org/.
- RStudio, and other contributions, have produced a series of useful Cheat Sheet references on key aspects of R/Studio/R Markdown https://www.rstudio.com/resources/cheatsheets/.

- All packages have a vignette providing explanatory text and example code for all the functions they contain. For some packages there are also web pages and PDFs which are available that give a more informative and practical guidance on their use.

2.4.3 What to do when you get stuck?

Google the question (as a sentence in English) and the answer is likely to appear in stack overflow https://stackoverflow.com/. Other good sources for answers (not an exhaustive list):

- Quick – R https://www.statmethods.net/
- R Tutorial http://www.r-tutor.com/r-introduction

To keep up to date, with coding ideas and new packages, sign up to R-bloggers https://www.r-bloggers.com/

CHAPTER 3: DATA MANAGEMENT

3.1 DATABASE SOURCES FOR THIS PROJECT

Water quality data for a period of 20 years was provided by WSP 1 and WSP 2. This data was provided in a text format as a .csv file. R and R Markdown are able to draw data directly from a LIMS database (csv format) and also can utilise data from a Microsoft Excel package. The real advantage of R is that it can read from multiple sources, for example: LIMS database/weather station data/GPS data) and publish data, code and text all together in one document from R Markdown. This R Markdown report can be updated and knitted (produces an html/ pdf/ word) document to provide the user with a reproducible and transparent document producing process.

3.2 INITIAL DATA ASSESSMENT

3.2.1 WSP 1 data

The first data extraction yielded over 1.7 million results from WSP 1 resulting in a csv file size of 97MB. There were 151 parameters measured with a group of these parameters measured regularly for operational purpose, the remainder for other reasons. This comprised of 62 'locations' (identified by long descriptor name) and 58 'sample points'. Table 3-1 shows a list of the parameters that were not available for visualisation because they were lost in parsing, a consequence of receiving the data as a csv file. This occurs because comma delimitation creates columns with zero entries and it is not possible to confidently assign the associated data thereafter.

Table 3-1: Systematic names versus transferred information into R

Systematic Name	Data Field Identifier
2,4-Dinitrophenylhydrazine	2,4 – dnp
2,4,6-Trichlorophenol	2,4,6 – tcp
2,4-Dichlorophenol	2,4 – dcp
2,4-dimethylphenol	2, 4 – dmp

As an example, "2,4,6 – tcp" presents as 3 columns with headings as follows:

2	4	6 – tcp

This results in confusion because, if all the information is contained in a column with the heading "2" as a default then; when that column is called, which is the way R works, it is not evident to which of the original filed identifiers the data relates to. It will simply appear in an assessment as a treatment of parameter "2" which could contain any, or all, of similarly named comma delaminated identifiers:

- 2,4-Dinitrophenylhydrazine
- 2,4,6-Trichlorophenol
- 2,4-Dichlorophenol
- 2,4-dimethylphenol

It is likely that corrections to the data field identifier would rectify this problem (see Table 3-1). Alternatively, transmission of the information in a different file format may eliminate the problem.

3.2.2 WSP 2 data

At WSP 2, about 151 parameters are measured; however, they are not synchronous with those monitored by WSP 1. For example, microbiological parameter monitoring differs, see Table 3-2. There are 389 'locations' described as a long name but 391 'sample points'. This implies that two sample locations may have duplicate names but with different datasets.

Table 3-2: Example of data monitoring differences between the two WSP databases

Field identifier for Microbiological Parameter	Dataset
HPC 37	WSP 1
HPC 21	WSP 1
Total Organisms	WSP 2

3.2.3 Comments on WSP 1 and 2 data

For WSP 2, some reservoir and distribution data were omitted from the initial review. For WSP 1, Heterotrophic Plate Count at 37˚C (HPC 37) and Heterotrophic Plate Count at 21˚C (HPC 21) were not included in the initial review. The reason for this exclusion was identified as being non-recognition of the test abbreviation in the original interrogation call by the R Software. This is a coding issue compounded by confusion centring on different field identifiers, for example; between the two datasets, three different field identifiers are relevant for heterotrophic plate counts See Table 3-3.

Table 3-3: Examples of data field assignments in different WSPs

Field identifier for Microbiological Parameter	Dataset
HPC 37	WSP 1
HPC 21	WSP 1
Total Organisms	WSP 2

It is apparent that the WSP 1 data differentiates between the different incubation temperatures of 21˚C and 37˚C. However, it is not descriptive of the actual parameter as the field identifier '*E coli'* is. The WSP 2 data is even more obscure as that the there is no incubation temperature identified in the primary field identifier 'Total Organisms', it appears that this naming was intuitive. After consultation, it was disclosed that this identifier refers to the measurement of HPC at 37˚C. It has been suggested, and accepted by the team, that the water quality abbreviations presented in Section 4.3 of SANS 241-1:2015 are to be used as default descriptors for reporting purposes, however this will require some refinement since the standard itself does not differentiate HPC at the two different incubation temperatures in the table referenced.

The case studies undertaken during this research allowed the following advice to be developed:
It is recommended that a team is constituted to investigate the value of and requirement for a Laboratory Data Management Policy to guide decision-making regarding laboratory data. This may include, amongst others, issues related to reporting limit changes, documentation of analytical method changes, issues related to

decimal point reporting, data validation, etc. It is further recommended that this team is led by the Quality Team from Laboratory Services, but staff from other water company departments will also need to contribute. This recommendation allows for an inclusive approach to many, or all, of the data issues encountered and ensures that the outcome of any decisions reached is fully integrated in to the reporting system consistent with laboratory and any sampling accreditation process in place.

3.3 PRELIMINARY TECHNICAL INTERPRETATION OF INITIAL DATA ASSESSMENTS

3.3.1 Data clean-up

An enormous amount of time was spent during this project to clean up the dataset provided from the water utilities. Dates are also an important anchor in data; without a date (and ideally the time) the information is redundant. A date expressed as 6.11.10 has six different possibilities: 6th Nov 2010 / 10th Nov 2006 / 11th Jun 2010 / 10th Jun 2011 / 6th Oct 2011 / 11th Oct 2006 – so what date is it? Tidy data should always express date and time in the correct format following the principles of ISO 8601:2004 /SANS: yyyymmdd hh:mm:ss. Local time should be used stating the Time Zone reference (e.g. Africa/Johannesburg) as this can allow for daylight saving. The water quality monitoring database used for visualisation was in a tidy format, example data is shown in Table 3-4, however, the data could be improved by:

- rationalising (standardising) the determinand names
- adding a variable column that states the method used for the analysis (e.g. the SCA blue book reference)
- including metadata such as the Time Zone and co-ordinates of the sample locations.

Table 3-4: Example – Tidy format of Water quality data

Sample_Point	Description	Sample_Date	Determinand	Result	Unit
TDH007	WTW1 final 1	1991-01-29	Al (S)	36	µg/L
TDH007	WTW1 final 1	1990-01-11	CC 21	0	mL
TDH007	WTW1 final 1	1990-01-11	CO2 (F)	.92	mg CO2 / L
TDV006	WTW3 WW final	2016-03-21	Coliforms 51	0	-
TDH010	WTW1 final 3	1990-06-14	F	34.9	µg/L
TAM010	WTW2 final	1990-06-15	F.strep	0	100 ml
TAM010	WTW2 final	2016-04-08	Geosmin measured	<0.005	µg/L
TDV006	WTW3 WW final	2015-08-06	NH2Cl	2.40	mg N/L
TDV006	WTW3 WW final	2016-05-05	Plate 9 count	0	-
TDH008	WTW1 final 2	2013-05-22	U	0.05	µg/L

As mentioned earlier, most of the time in data science is spent wrangling data. A rough estimate is that 80% of time doing data science is spent wrangling the data; see Figure 2-3. This also includes transforming and cleaning data into a tidy format, with each variable in a column; (see Figure 2-4) or merging datasets together such as the sample location co-ordinates and the monitoring results. In this project, a tool/manual is developed to make this task and the rest of data science method easier.

3.3.2 Visualisation: WSP 1 1990 to 2017

Visualisation of the data is influenced by the reporting limits of the results, for example hundreds of chlorine results reported to one decimal place result in a straight line. The implication is that this practice may be biasing the data and thus reducing the sensitivity of any subsequent future data correlation assessment. However, confirmation of apparent trends in data is still achievable using a non-parametric trend evaluation technique such as Locally Weighted Scatterplot Smoothing (LOWESS). An example of the potential loss of

sensitivity within the data can be seen for the time series plot of Turbidity for “WTW2 final” samples where the reporting limit criteria changed notably after mid-2014. Results thereafter appear to be reported as one of four options rather than as true data, i.e. rounding bias has been introduced which is likely’ in time’ to significantly influence any seasonal assessment of the data from that date forward for that treatment works. The same applies for “WTW final 1” and “WTW final 2” samples from 2015 onwards.

The box plot evaluations of sample temperatures give a very clear indication of when regrowth risk is more likely based on seasonal changes and could be developed to provide location specific advanced warnings of any need to boost disinfection. For example, May to October is a period when sample temperatures are below 20˚C at WTW 1; however, this would need to be investigated with parameters indicative of chlorine demand, such as turbidity, before any final operational tools can be derived.

Turbidity records exhibit notable differences between sources; this could be explored as a tool indicative of increased biofilm formation potential and infrastructure sedimentation risk relative to potential increases in dirty water complaints. Seasonal evaluations of turbidity are therefore worth exploring further.

3.3.3 Visualisation: WSP 2 1996 to 2017; Mixed Treatment Works and Distribution

A distribution reservoir within the area of supply shows periodicity in the time line for conductivity data which is not evident as long-term seasonal trends – the significance of this relative to changes in other water quality parameters is worthy of investigation. The pH record pattern for another distribution reservoir − shows a converse trend in reporting limits compared to that exhibited in the WSP 1 data referenced below, i.e. the data appears to present a greater resolution of information post 2010. This pattern of reporting is also present in pH data for another distribution reservoir

3.4 CONCLUSIONS FROM THE INITIAL DATA REVIEW

There is an issue with the lack of standardisation of field identifiers between the two datasets. It is anticipated that this problem will multiply when a greater number of datasets are submitted for a standardised review. Some data can be safely combined for initial visualisation purposes for example in the WSP 2 dataset various field identifiers for free chlorine could potentially be combined into one time series. Preliminary inspection of the time series plots of the data has already highlighted reporting limit policy changes. Over time these may significantly introduce bias and reduce the interpretive power of recently collected data, i.e. the results for the data recorded after mid-2015 is likely to have introduced a significant bias in that proper randomization of the data cannot be achieved thereby ensuring that data evaluated is not representative of the population intended to be assessed.

Compression of “y” axis on time lines is an endemic issue in both datasets and there is no realistic opportunity to verify the veracity of the causal result inputs. The phenomena appear to affect turbidity records more than other components of the datasets. There may therefore be a correlation with other parameters such as iron. This requires further exploration. The disparity between the number of ‘sample points’ and ‘locations’ in the both datasets suggests that some ‘locations’ are incorrect entries. It is anticipated that such disparities, if confirmed, can be addressed by coding and cross referencing to corrective look up tables in subsequent assessment runs.

3.5 RECOMMENDATIONS FOR WATER QUALITY DATA MANAGEMENT

The following points are relevant to all laboratories dealing with water quality data and databases:

1. **Review of Reporting Limit Policy** – It is recommended that data donors review their reporting limit policy to restore interpretive power of the long-term records.

2. **Data Field Identifiers** – It is recommended that data managers agree on a common format for field identifiers to allow easier collation of data prior to production of review reports.

3. **Rogue Data Rejection Protocol** – There is a need for a WSP laboratory to agree a data rejection protocol for rogue results based on a statistical assessment of the likelihood of such an input to be an outlier. Correlations of turbidity with iron and HPC values should be investigated as part of the development of the required protocol.

4. **Sample Point Location Identifiers** – It is recommended that data donors review the naming of sample point locations and advise the team of the disparities identified. No corrections to the primary data should be made without consultation with the team.

5. **Comma Inclusive Systematic Naming** – Data should be kept in a way that does not eliminate information due to the comma delamination of systematic chemical names in the filed identifiers. This is important because at present valuable information is being lost in the file transfer process.

6. **Provision of Meta Data** – It is recommended that data donors provide metadata associated with the water quality information in order that greater use can be made of the information created. For example, more value is generated from data when collating: weather observations with surface water quality; and water quality failures to geographical points within a distribution network. Parameters known to be of value from other data reviews include:
   - GPS locations of the sample points; and
   - Weather information: maximum and minimum temperature, Rainfall, Humidity, and Sunshine

7. **Data Unit Inconsistency** – It is recommended that in order to deal with ‘data unit’ inconsistency between sets, for example metals results expressed as µg/l and mg/l the default treatment to be followed will be that defined in section 4.3 of SANS 241-1:2015 or the WHO guidelines for drinking water where the SANS standard is not sufficient.

CHAPTER 4: DATA TRANSFORMATION

4.1 INTRODUCTION

Using the datasets from the two WSPs, this Chapter illustrates how R-codes can be used to transform data into a useable block; this block of data is a sub set of information from the database that can be used to create:

- summary statistics
- tables of information
- Time series graphs, etc.

This is part of the **Transform** process (see Figure 2-1). The sub set of information (the block of data) is created by:

- filtering the database for a single determinand or sample location
- selecting data between a date range

4.2 WATER QUALITY DATABASE STRUCTURE

4.2.1 Typical Water Quality Data

To illustrate the format of the data used for transformation, the first 20 rows of the water quality database is shown in Table 4-1. The water quality database has 1,712,175 records and covers the period from 1 January 1990 to 30 April 2017.

Table 4-1: Water quality database records

Sample_Point	Sample_Date	Determinand	Result	Units
TDH008	1990-01-06	Coliforms	0	MPN/100mL
TDH008	1990-01-06	Colour	<1	°H
TDH008	1990-01-06	*E. coli*	0	MPN/100mL
TWG010	1990-01-06	Coliforms	0	MPN/100mL
TWG010	1990-01-06	Colour	<1	°H
TWG010	1990-01-06	*E. coli*	0	MPN/100mL
TAM010	1990-01-07	Coliforms	0	MPN/100mL
TAM010	1990-01-07	Colour	<1	°H
TAM010	1990-01-07	*E. coli*	0	MPN/100mL
TUR007	1990-01-08	Coliforms	0	MPN/100mL
TUR007	1990-01-08	Colour	1	°H
TUR007	1990-01-08	*E. coli*	0	MPN/100mL
TDH007	1990-01-10	Coliforms	0	MPN/100mL
TDH007	1990-01-10	Colour	<1	°H
TDH007	1990-01-10	*E. coli*	0	MPN/100mL
TMM031	1990-01-12	Coliforms	0	MPN/100mL
TMM031	1990-01-12	Colour	4	°H
TMM031	1990-01-12	*E. coli*	0	MPN/100mL
TMM030	1990-01-15	Coliforms	0	MPN/100mL
TMM030	1990-01-15	Colour	<1	°H

4.2.2 Determinands Measured

The total number of determinands measured is 151;
Table 4-2 lists the determinands ranked on the number of occasions during the period 1 Jan 1990 to 30 Apr 2017.

Table 4-2: List of Determinands measured with more than 20 000 results

Determinands	Number	Units
Turbidity	121764	NTU
Free Chlorine	116674	mg Cl2/L
Total Chlorine	116515	mg Cl2/L
E. coli	115891	MPN/100mL
Coliforms	115540	MPN/100mL
Temperature OS	105901	°C
Conductivity	95704	mS/m
HPC 37	94534	mL
HPC 21	78103	mL
Colour	75711	°H
pH OS	48024	-
NLF	45570	100 ml
Turbidity OS	33882	NTU
Odour	32978	-
Odour 60	32277	-
Taste	31389	-
Fe	28580	mg/L
Mn	28578	mg/L
NH3	21817	mg N/L

4.3 FILTERING DATA

4.3.1 Overview

The water quality databases contain measurements:

- at various locations
- for a range of determinands
- over periods of time

To create summary statistics and visualisation for individual (or groups of) locations, determinands and periods of time we *filter* the database so only the required information is left.

4.3.2 Filtering Data by Date

It is useful to filter data by date; for example, all the monitoring recorded after 30 Jun 2010. The R package lubridate (part of the tidyverse) has made it easier and more flexible to handle dates. More on lubridate can be found in the dates and times in the R for Data Science chapter http://r4ds.had.co.nz/dates-and-times.html; there is also a cheat sheet on lubridate that can be printed out https://www.rstudio.com/resources/cheatsheets/. Box 4-1 shows the series for steps for filtering data by date and Table 4-3 is an example sample of the data filtered by date.

Box 4-1: Example filtering data by date

```
#Make a variable `AfterJul20` that is the date 30/06/2017.
AfterJun30 <- lubridate::dmy("30 Jun 2010") # note the order of dmy
#Some alternative formats to make the variable
AfterJun30 <- lubridate::dmy("30/06/2010") # note the order of dmy
AfterJun30 <- lubridate::dmy("30-06-2010")
AfterJun30 <- lubridate::ymd("20100630") # note the order of ymd
# We put the filtered data into a new data.frame DataAfterJul20
DataAfterJul20 <- filtered_data %>%
# DataAfterJul20 is filter for dates after "30/06/2017"
filter(Sample_Date > AfterJun30)
```

Table 4-3: Filtered Data all results after 30/03/2010 (sample)

Sample_Point	Sample_Date	Determinand	Result	Units
TAM010	2016-06-03	Conductivity	26.20	mS/m
TAM010	2013-03-04	Conductivity	19.92	mS/m
TAM010	2013-10-30	Conductivity	25.50	mS/m
TAM010	2013-03-20	Conductivity	20.10	mS/m
TAM010	2013-09-04	Conductivity	24.50	mS/m
TAM010	2012-06-27	Conductivity	21.40	mS/m
TAM010	2015-12-04	Conductivity	28.60	mS/m
TAM010	2016-06-29	Conductivity	23.00	mS/m
TAM010	2013-05-24	Conductivity	20.80	mS/m

4.3.3 Filtering Data to a Date Range

To filter for a data range, e.g. greater than 30 Jun 2010 but less than 30 June 2012, the same filter function is used filter (Sample_Date > afterDate & Sample_Date < beforeDate); in words this says only results after 30 June 2010 and before 30 June 2012 (see Box 4-2).

Box 4-2: Example filtering data to a date range

```
#Date range
afterDate<-lubridate::ymd("20100630")#30June2010
beforeDate<-lubridate::ymd("20120630")#30June2012
# We put the filtered data into a new data.frame Data_range
Data_range <- filtered_data %>%
        # Data_range is filtered for > 30 Jun 2010 but < 30 June 2012
        filter(Sample_Date > afterDate & Sample_Date < beforeDate)
```

A sample of the filtered data, by the date range, after 30 June 2010 and before 30 June 2012, is shown in Table 4-4. The data filtered by the date range 30/06/2010 to 30/6/2012 is shown in **Error! Reference source not found.**. The grey shaded area shows the 95% confidence level interval for predictions of the linear regression applied. For background information see https://en.wikipedia.org/wiki/Linear_regression.

Table 4-4: Filtered data 30/06/2010 to 30/06/2012 (sample)

Sample Point	Sample Date	Determinand	Result	Units
TAM010	2011-05-23	Conductivity	22.2	mS/m
TAM010	2011-08-08	Conductivity	23.1	mS/m
TAM010	2010-09-08	Conductivity	21.5	mS/m
TAM010	2012-03-09	Conductivity	24.4	mS/m
TAM010	2010-12-31	Conductivity	21.0	mS/m
TAM010	2011-09-09	Conductivity	26.3	mS/m
TAM010	2011-05-27	Conductivity	22.0	mS/m
TAM010	2010-11-03	Conductivity	21.8	mS/m
TAM010	2010-12-03	Conductivity	22.0	mS/m
TAM010	2010-11-01	Conductivity	20.3	mS/m

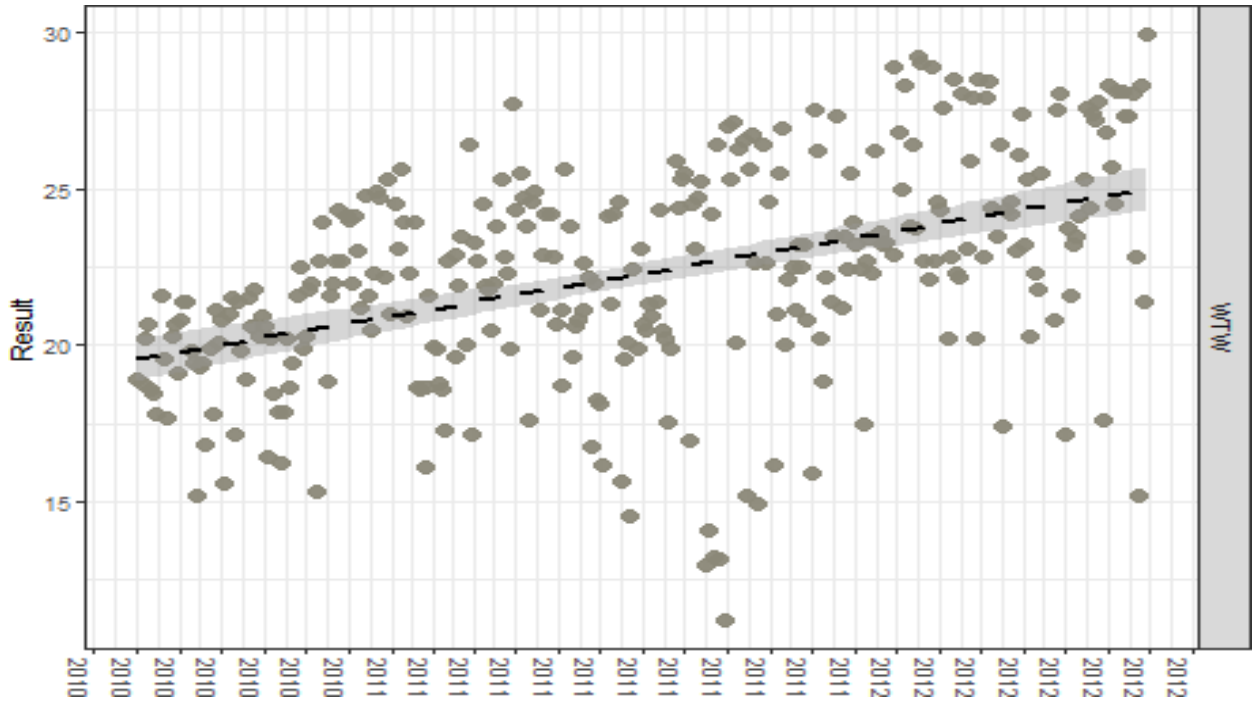

Figure 4-1: Linear regression analysis of WTW Conductivity 30/6/2010 to 30/6/12

4.3.4 Filtering Data by Determinand

Box 4-3 below is an example of filtering the database by **Conductivity** determinand at a water treatment works; sample point TAM010

Box 4-3: Example filtering data by determinand (conductivity)

```
#We put the filtered data into a new data.frame filtered_data
filtered_data <- U_Data %>%
# The variable Sample_Point is filtered for TAM010
# at the same time the variable Determinand is filtered Conductivity
filter(Sample_Point == "TAM010", Determinand == "Conductivity")
```

Details of this data wrangling process and use of the pipe operator **%>%** see: http://r4ds.had.co.nz/transform.html. The filtered database filtered_data we can see that there are now 6,878 observations from a total of 1,712,175 in the original database U_Data. All the filtered_data is shown in Table 4-5 below.

Table 4-5: Summary Statistics for Conductivity at a WTW (sample)

Sample_Point	Sample_Date	Determinand	Result	Units
TAM010	2012-06-22	Conductivity	15.23	mS/m
TAM010	2012-04-26	Conductivity	28.00	mS/m
TAM010	1996-01-12	Conductivity	15.9	mS/m
TAM010	1991-08-03	Conductivity	13.1	mS/m
TAM010	1990-02-21	Conductivity	14.1	mS/m
TAM010	2009-07-01	Conductivity	22.60	mS/m
TAM010	2015-03-02	Conductivity	28.1	mS/m
TAM010	1993-03-18	Conductivity	14.9	mS/m
TAM010	1995-10-19	Conductivity	18.3	mS/m
TAM010	2010-10-01	Conductivity	20.60	mS/m

4.3.5 Summary statistics

The list of data in Table 4-5 above is not very informative (if only because it is difficult to see all the data at one time); to gain a better understanding of the Conductivity data, summary statistics can be produced. The code below uses the mosaic package to calculate some favourite stats using the function favstats; the summary statistics for the Conductivity for a WTW is shown in Table 4-6.

Box 4-4: Example generating conductivity summary statistics

```
#The summary statistics are saved into a variable sumstats
sumstats <- favstats(~Result, data = filtered_data)
#sumstats is made into a the table (using the code below)
knitr::kable(sumstats,
t_output,
caption='SummaryStatistics for Conductivity for a WTW',
booktabs = TRUE,
longtable = TRUE,
align = c("l", rep("r",9)))
```

Table 4-6: Summary Statistics for conductivity at a WTW (sample)

min	Q1	median	Q3	max	mean	sd	n	missing
2.18	13.8	15.48	18.7	37.9	16.98729	4.573197	6878	0

4.4 CENSORED VALUES

A censored value is a condition in which the value of a measurement or observation is only partially known, by convention it is shown by a greater than > or less than < symbol. Many results in water quality monitoring are censored; a sample from a typical water company's water quality database is shown in Table 4-7.

Table 4-7: Range of determinands with censored values (sample)

Determinand	Result	Units
CN (F)	<10.0	µg/L
F	<76	µg/L
Co	<10.0	µg Co/L
As	<2	µg/L
Fe	<0.02	mg/L
THM	<40.0	µg/L
Mn	<.01	mg/L
Al (T)	<25.00	µg/L
Zn	<0.030	mg Zn/L
NO3 + NO2	<0.49	mg N/L

The analysis of censored values is not undertaken, this is currently outside the scope of work for this project however information on how to analyse censored data can be found in the *NADA* package https://cran.r-project.org/web/ packages/NADA/NADA.pdf. In this report censored values are just substituted for that value; by removing the greater or less than symbol and making the value numeric; using the function *MakeCenNumeric*. The results with censoring removed are shown in the Table 4-8 below.

Table 4-8: Censored values substituted with the value recorded

Determinand	Result	Units
CN (F)	10.00	µg/L
F	76.00	µg/L
Co	10.00	µg Co/L
As	2.00	µg/L
Fe	0.02	mg/L
THM	40.00	µg/L
Mn	0.01	mg/L
Al (T)	25.00	µg/L
Zn	0.03	mg Zn/L
NO3 + NO2	0.49	mg N/L

CHAPTER 5: DATA VISUALISATION

5.1 INTRODUCTION

A key aspect of data science is visualisation; see Figure 2-1. This Chapter builds on the filtering data techniques acquired in Chapter 4 and applies the functions used for generating standard graphs, so the data can be explored visually. R Markdown is able to generate a HTML version of a document and this can be used interactively. For example, if the conductivity for a WTW is drawn as a times series plot; the graph is also interactive so it can be zoomed to show specific time periods and the mouse can hover on individual values. Applying the R code to blocks of data to create the graphics is described in details in workshop tutorial sections (Chapters 6, 7 and 8).

5.2 USING STANDARD GRAPHS TO DESCRIBE HIDDEN RELATIONSHIPS IN WATER QUALITY DATASETS

5.2.1 Overview

This section is aimed at illustrating the range of graphs that can help describe the hidden relationships in the drinking water quality data. Applying the R code to blocks of data to create the graphics is described in Section 5.5.

5.2.2 Determinand on a Log and Linear Scale

An example determinand time line using a log scale is shown in
Figure 5-1 for *E.coli* at (Water Treatment Works Final) from 2008 to 2016. By convention, and clarity, microbiological determinands are displayed on a log scale.

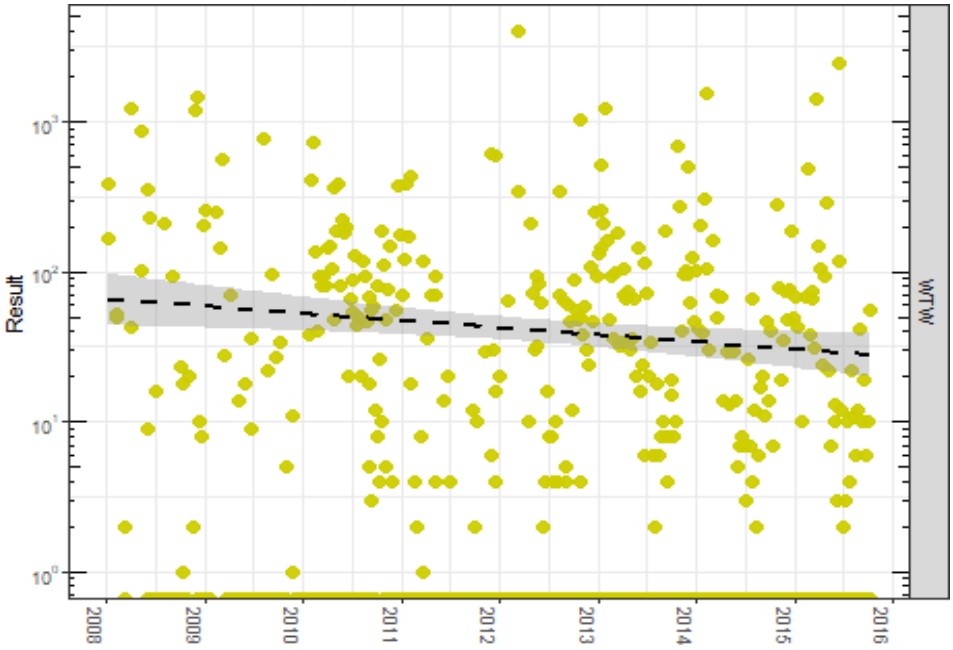

Figure 5-1: Example Log plot with trend line

The graph below (Figure 5-2) shows a trend line for a simple linear regression applied to the *E.coli* data plotted above in Figure 5-1. The grey shaded area shows the 95% confidence level interval for predictions of the linear regression applied. For background information see https://en.wikipedia.org/wiki/Linear_regression.

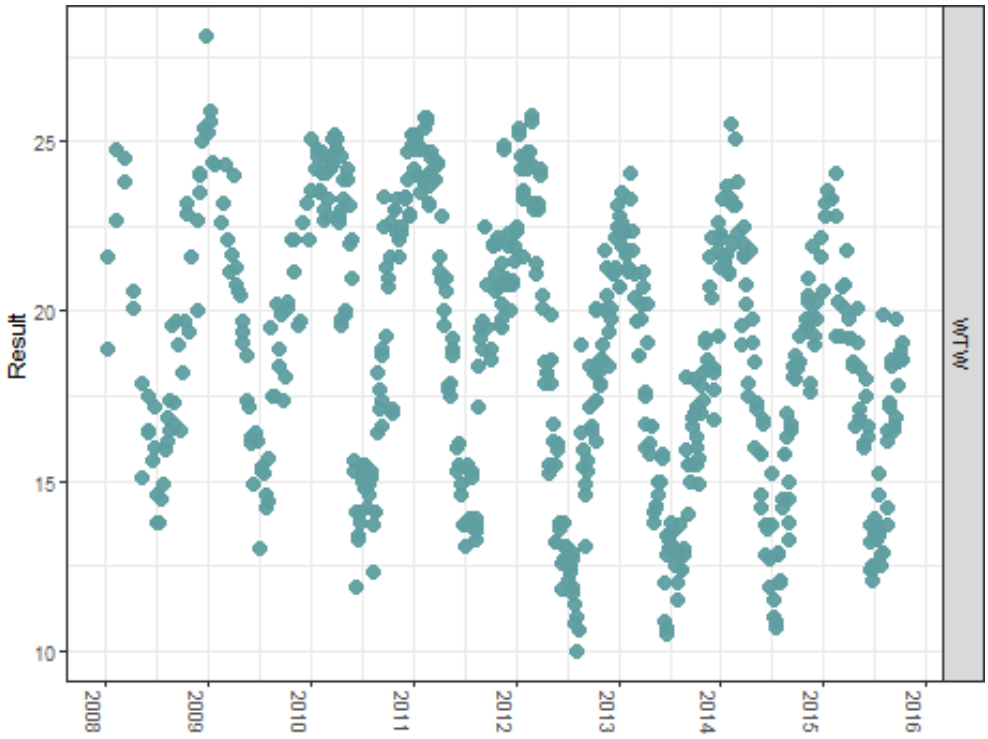

Figure 5-2: Example Linear Plot

Other determinands that would generally be plotted on a linear scale are:

- Temperature
- Ammonia
- Chlorine
- Colour
- Conductivity
- Metals (Fe, Mn, etc.…)

5.2.3 Cumulative Sum Plot of Determinand

Cumulative sum plots are a technique that can be used to identify changes not otherwise evident in noisy data. Figure 5-3 shows a cumulative sum plot for *E. coli* over time; the slope of the graph line is useful in illustrating water quality:

- a horizontal line indicating no *E. coli* present
- the steepness of line indicating the level of *E. coli* present

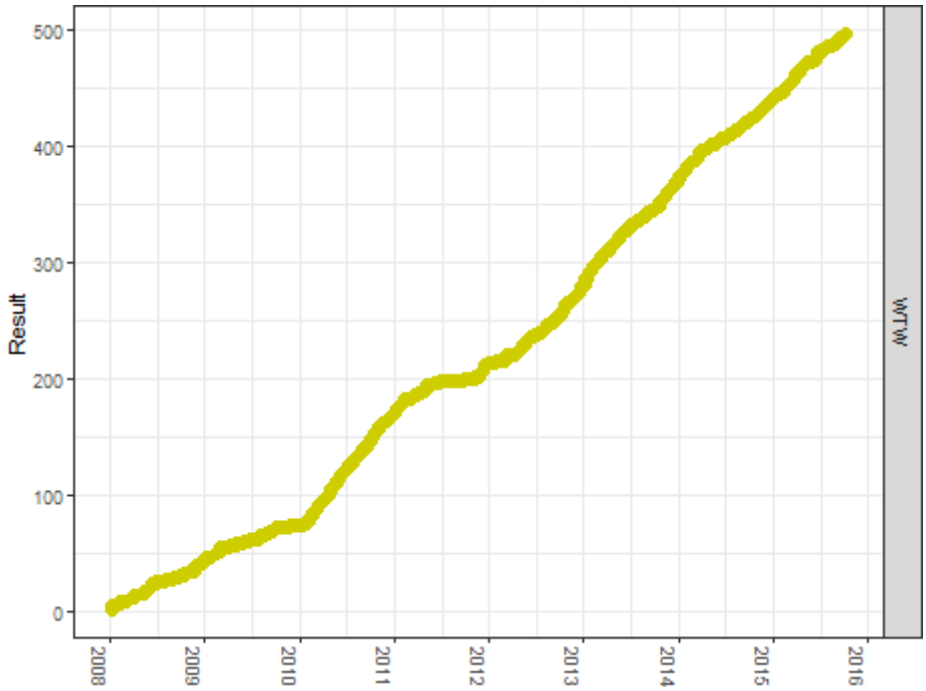

Figure 5-3: Example of a cumulative sum time line plot for *E.coli*

5.2.4 Boxplots of Determinands for a Period of Time

A boxplot is a visual method of showing a distribution of values; it is popular among statisticians and is a useful way to explore data. Figure 5-4 below shows the essential outline of the boxplot. Each boxplot consists of a box that stretches from the 25th percentile (Q1) of the distribution to the 75th percentile (Q3), a distance known as the interquartile range (IQR). In the middle of the box is a line that displays the median, i.e. 50th percentile, of the distribution. These three lines give you a sense of the spread of the distribution and whether or not the distribution is symmetric about the median or skewed to one side. The dots at the end of the boxplot represent outliers There are a number of different rules for determining if a point is an outlier, but the method that R (R Core Team, 2018) and ggplot (Wickham et al., 2018) use is the 1.5 rule.

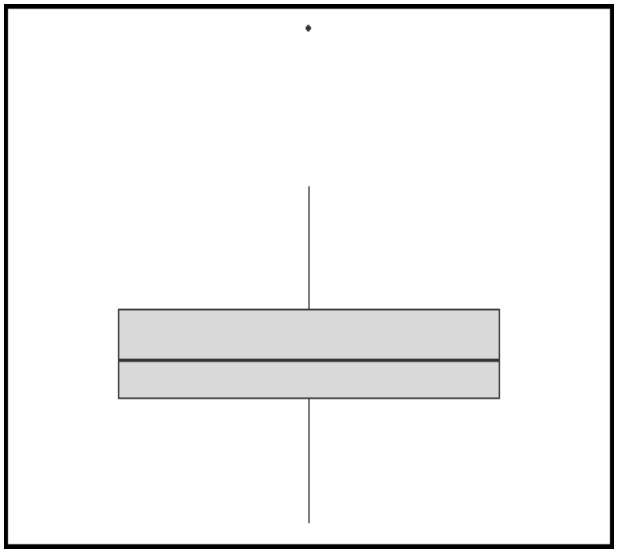

Figure 5-4: Example of a boxplot with whiskers and an outlier

If a data point is:

- less than Q1 − 1.5 ∗ IQR
- greater than Q3 + 1.5 ∗ IQR

Then that point is classed as an outlier. IQR is Inter Quartile Range. The whiskers are defined as:

- upper whisker = $min(max(x), Q_3 + 1.5 * IQR)$
- lower whisker = $max(min(x), Q_1 - 1.5 * IQR)$

where IQR = Q3 − Q1, the box length. So, the upper whisker is located at the smaller of the maximum x value and Q3 + 1.5 ∗ IQR, whereas the lower whisker is located at the larger of the smallest x value and Q1 − 1.5 ∗ IQR.

Additional information about boxplots can be found here: https://en.wikipedia.org/wiki/Box_plot

5.2.4.1 Monthly Variation

Figure 5-5 below shows a boxplot for *E.coli* for each month of the year.

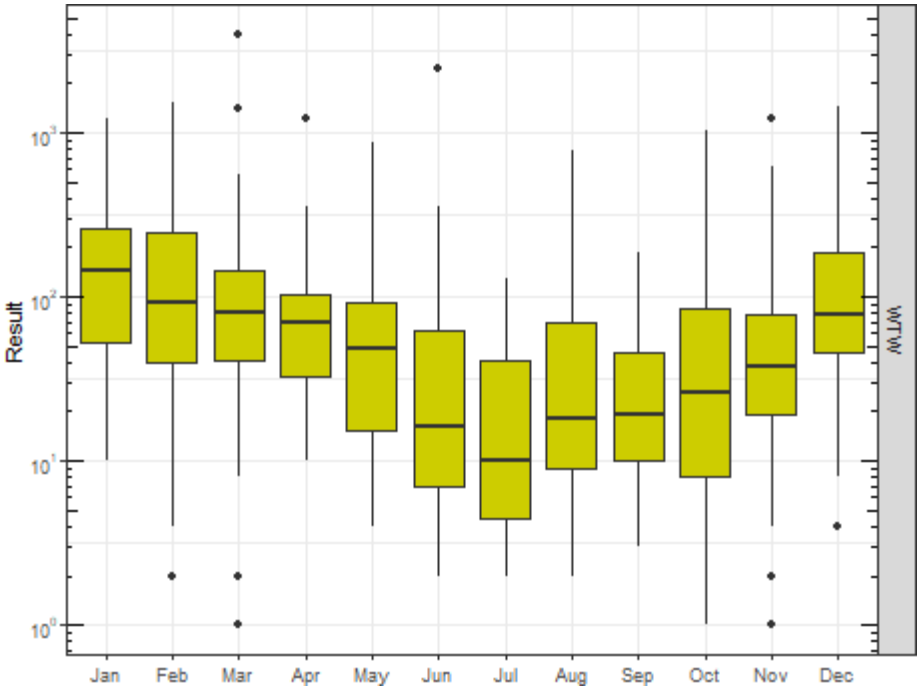

Figure 5-5: Example of a series of monthly box plots

5.2.4.2 Annual Variation

Figure 5-6 shows a boxplot for annual temperatures.

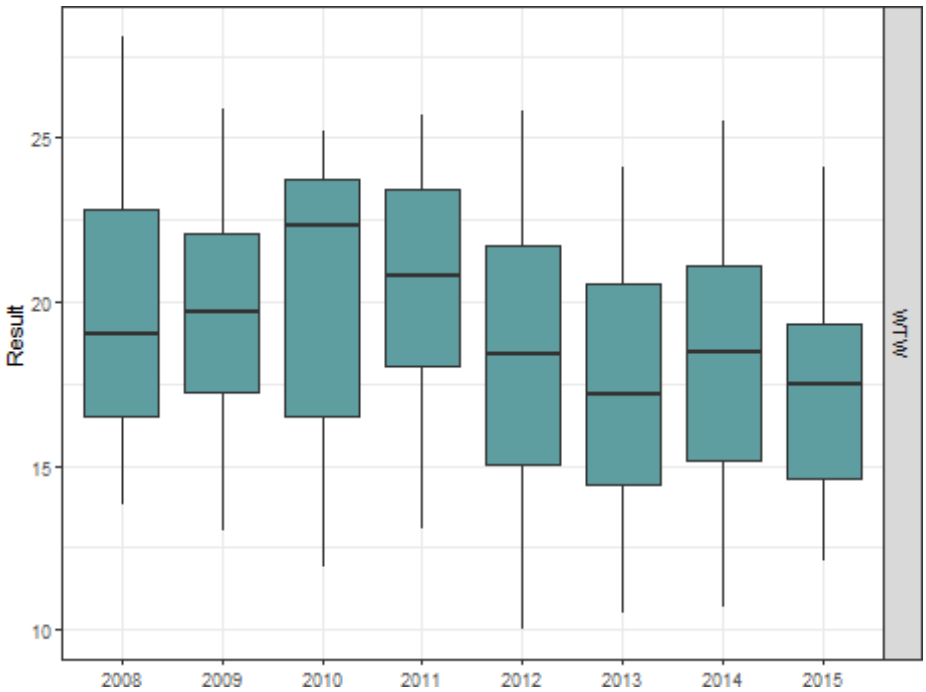

Figure 5-6: Example series of annual boxplots

5.2.5 Seasonal Variation

Figure 5-7 shows a boxplot for seasonal turbidity.

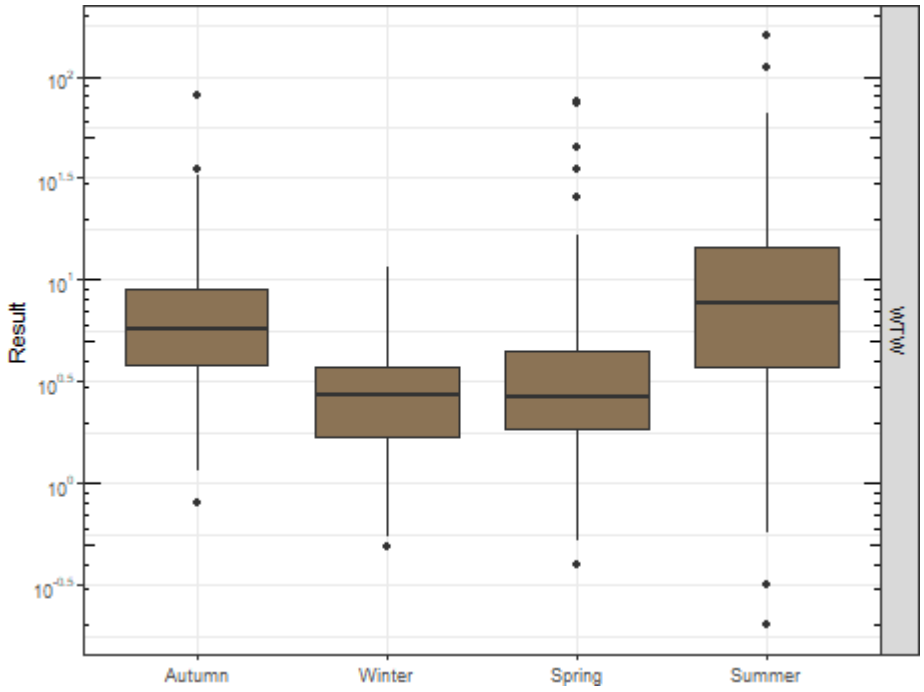

Figure 5-7: Example of a series of seasonal turbidity

5.3 VISUALISATION CODING FOR SCATTER PLOTS OF DETERMINANDS AGAINST TIME

5.3.1 Loading the database into R

First the water quality database has to be loaded into R; because of the large size of the databases these have been converted to a binary format so that the computer can load them quickly. The code that loads *Water Company* quality database into an R dataframe *U-data* is shown in Box 5-1.

Box 5-1: Example loading a database into an R dataframe

```
#Read feather file - faster to read
path <- "..//CSVwater//Data//Sample_data.feather"
#data.frame U-data
U_Data <- read_feather(path)
```

Note: – the Water Company monitoring database csv files were converted to binary formats using the feather package. Further information on the use of feather can be found https://blog.rstudio.com/2016/03/29/ feather/.

The *glimpse* function (see Box 5-2) provides the structure of the data loaded for example:

- the number of columns and rows of data (e.g. 1,712,175 observations. of 6 variables)
- the name of the columns (from the database, e.g. $ Determinand)
- the data type
- *chr* character
- *date* date

As shown much data as possible is shown for each column

Box 5-2: Example information on the structure of the data using the glimpse function

```
## Observations: 636,064
## Variables: 7
## $ Sample_Point <chr> "TDH008", "TDH008", "TDH008", "TAM010", "TAM010",...
## $ Description <chr> "WTW1 final 2", "WTW1 final 2", "WTW1 final 2", "...
## $ Sample_Date <date> 1990-01-06, 1990-01-06, 1990-01-06, 1990-01-07, ..
## $ Determinand <chr>"Coliforms","Colour","E.coli","Coliforms","Co...
##$Result <chr>"0","<1","0","0","<1","0","0","<1","0","...
## $ Unit  <chr> "MPN/100mL", "°H", "MPN/100mL", "MPN/100mL", "°H"...
##$SelectWTW  <lgl>TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,T...
```

Note: – the result column is shown as a character <chr> because of the censored values; these are converted to numeric values using the function *MakeCenNumeric*.

The function *MakeCenNumeric* is a bespoke function for the project and is one of a collection of functions in the file magic_functions.R. The Results column is now a data type dbl numeric (*double precision value*) (see Box 5-3).

Box 5-3: Example using the *MakeCenNumeric* function

```
## Observations: 636,064
## Variables: 7
##$ Sample_Point<chr>"TDH008","TDH008","TDH008","TAM010","TAM010",...
## $Description<chr>"WTW1final2","WTW1final2","WTW1final2","...
##$Sample_Date <date>1990-01-06,1990-01-06,1990-01-06,1990-01-07,...
## $Determinand<chr>"Coliforms","Colour","E.coli","Coliforms","Co...
##$ Result <dbl>0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 2...
## $ Unit   <chr> "MPN/100mL", "°H", "MPN/100mL", "MPN/100mL", "°H"...
##$SelectWTW   <lgl>TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,TRUE,T...
```

5.3.2 Filter Data for Determinand and Location

Box 5-4 and 5-5 show examples of filtering data for a determinand and location.

Box 5-4: Example for filtering data for WTW 1 Final 3

```
WTW1_colour <- U_Data %>%
filter(Sample_Point == "TDH010", Determinand == "Colour")
```

Box 5-5: Example for filtering data for WTW2 Final

```
WTW2_colour <- U_Data %>%
filter(Sample_Point == "TAM010", Determinand == "Colour")
```

5.3.3 Plot function Magic_TimeLine

The function Magic_TimeLine plots a time line scatter plot and is one of the functions in the file magic_functions.R (see Box 5-6). For the Magic_TimeLine function to plot the graph the following information is required:

- PlotData – a dataframe of filtered data, e.g. Durban_colour
- *S_date* – start date for the plot time line to begin
- *E_date* – end date for the plot time line to finish
- *TimeBreak* – how often the date is displayed on the x-axis, e.g. ("1 month", "1 week")
- *ParaColour* – colour of the graph point
- *Gtitle* – title of graph as text
- *Gcaption* – caption that appears bottom left of graph as text
- *Units* – parameter units of measurement as text
- *LinearScale* – TRUE the y-axis is linear; FALSE the y-axis is logarithmic
- *Regression* – FALSE no trend line is shown; TRUE trend line is shown
- *Alpha* – transparency of the plotted point, e.g. 0.0 invisible 1.0opaque
- *Size* – point size, e.g. 1 small, 6 large

Note*:* – a list of colours is available at http://www.stat.columbia.edu/~tzheng/files/Rcolor.pdf

Box 5-6: Example for plotting a scatter plot using the Magic_TimeLine function

```
Magic_TimeLine(PlotData,
   S_date,
   E_date,
   TimeBreak = "1 year",
   ParaColour ="darkorange3",
   Gtitle = "Conductivity",
   Gcaption="Examplegraph",
   Units = "mS/m",
   LinearScale=TRUE,
   Regression =FALSE,
   Alpha = 0.95,
   Size = 3)
```

The completed Magic_TimeLine function to plot the colour at WTW2 final over time for sample points TDH010 and TAM010 is shown in Figure 5-8 below. Note the graph is saved to a variable `g2`.

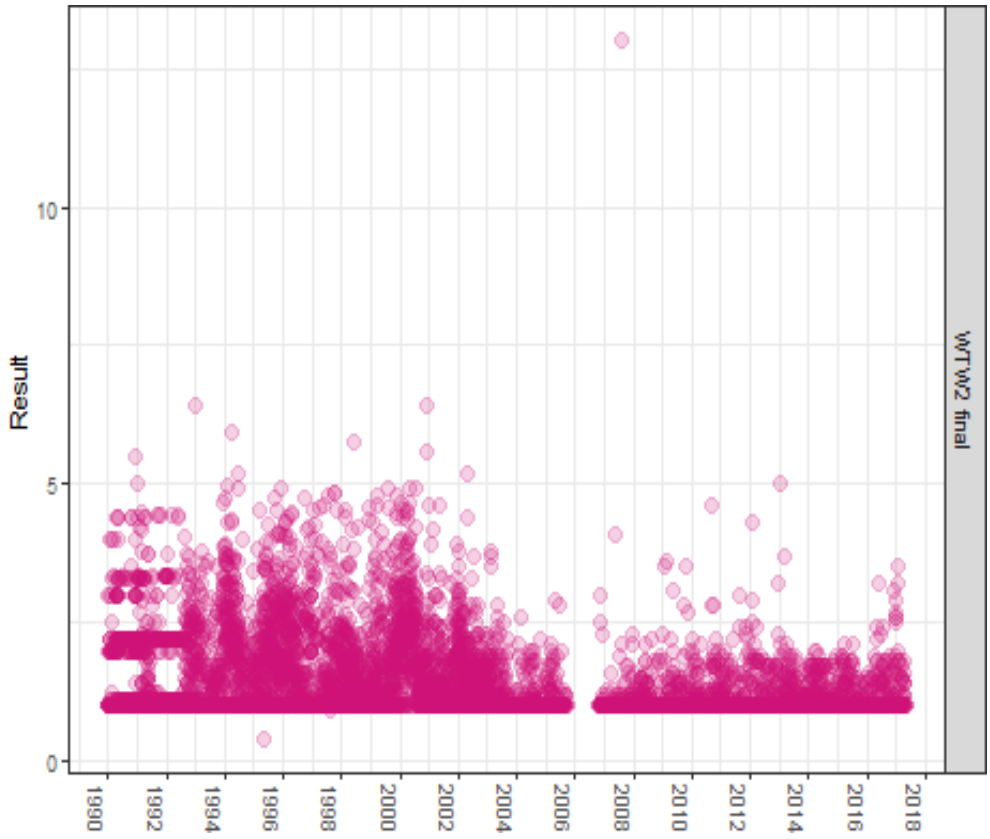

Figure 5-8: Colour WTW2 final

5.3.4 Arranging Graphs on a Page

Box 5-7 shows the graph produced by Magic_TimeLine going to a variable g1 and then to a function gridExtra::grid.arrange to plot the graph; this enables plots to be arranged on a page.

Box 5-7: Example for plotting a scatter plot using the gridExtra::grid.arrange function

```
#find start date of data
S_date <- min(WTW1_colour$Sample_Date)
#find start date of data
E_date <- max(WTW1_colour$Sample_Date)
#Graph uses the variable name SampleDate
#Must be date only
WTW1_colour$SampleDate <-lubridate::as_date(WTW1_colour$Sample_Date)
g1 <- Magic_TimeLine(WTW1_colour,
S_date,
E_date,
TimeBreak = "2 year",
ParaColour ="darkorange3",
Gtitle = "Colour: "WTW1 final 3",
Gcaption="Note the transparency of the points with Alpha = 0.3", Units = "H",
LinearScale=TRUE, Regression =FALSE, Alpha =0.3,
Size = 3)
gridExtra::grid.arrange(g1)
#find start date of data
S_date <- min(WTW2_colour$Sample_Date)
#find start date of data
E_date <- max(WTW2_colour$Sample_Date)
#Graph uses the variable name SampleDate
#Must be date only
WTW2_colour$SampleDate <-lubridate::as_date(WTW2_colour$Sample_Date)
g2 <- Magic_TimeLine(WTW2_colour,
S_date, E_date,
TimeBreak = "2 year", ParaColour = "deeppink3", Gtitle="WTW2Final:Colour",
Gcaption="NotethetransparencyofthepointswithAlpha=0.2", Units = "H",
LinearScale=TRUE, Regression =FALSE, Alpha =0.2,
Size = 3)
gridExtra::grid.arrange(g2)
```

To assist with the graphical analysis, it is often useful to display two (or more) graphs side by side or above each other; the package *gridExtra* makes this possible. Figure 5-9 and 5-10 show typical results when the *gridExtra* package is used. In Figure 5-9, the graphs appear side by side (ncol = 2). In Figure 5-10, one graph is above the other (nrow = 2).

For Figure 5-9:

```
gridExtra::grid.arrange(g2,g5,ncol=2)
```

For Figure 5-10:

```
gridExtra::grid.arrange(g2,g5,nrow=2)
```

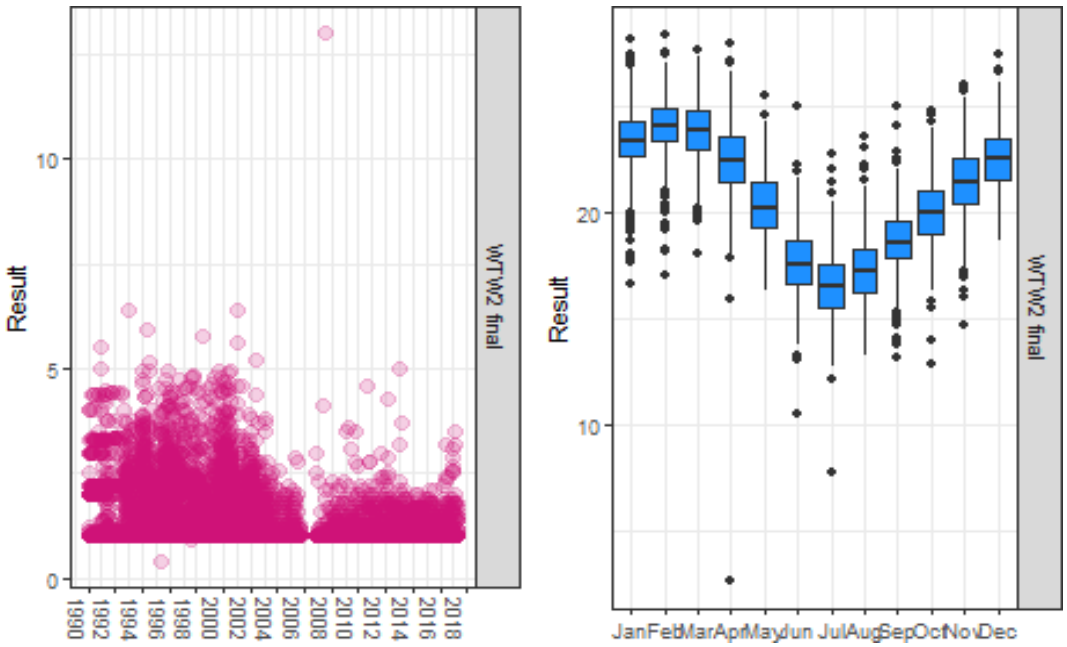

Figure 5-9: Arranging figures into a grid using the gridExtra function

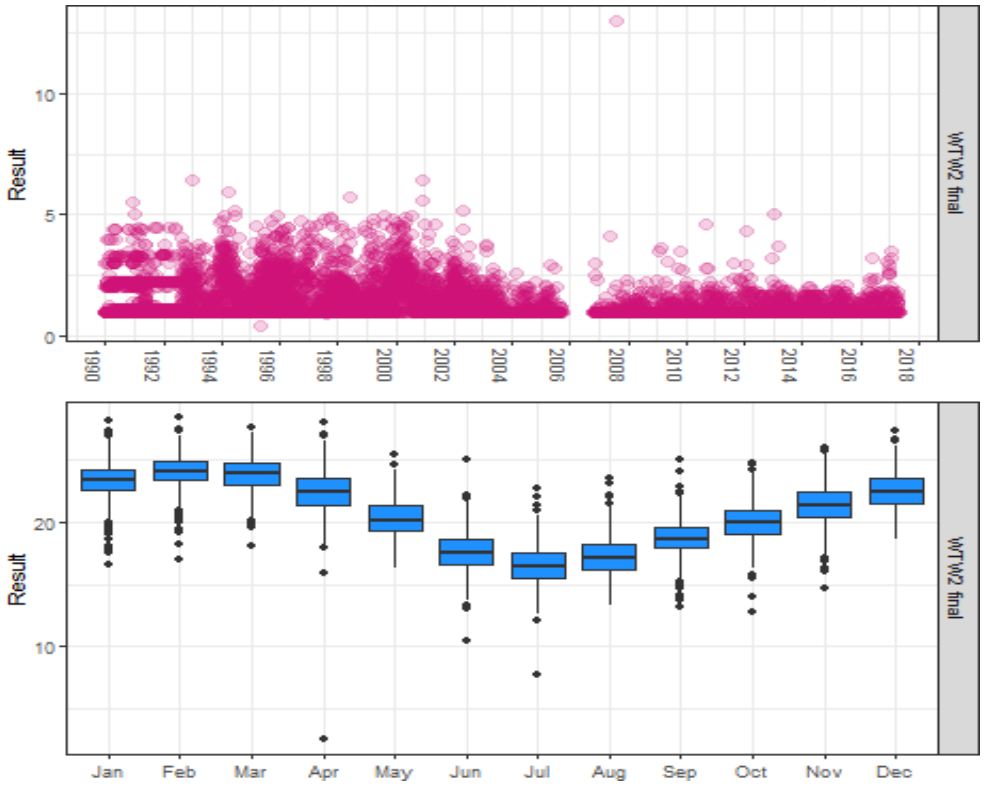

Figure 5-10: Arranging figures into columns

The web link https://cran.r-project.org/web/packages/gridExtra/vignettes/arrangeGrob.html details how *gridExtra* can display other, more complex, layouts.

5.3.5 Scatter plots for Two or More Locations

It is useful to look at the same determinand for two or more locations, the code below filters the U_Data for Coliforms at three different locations (Box 5-8), note the use of the log scale y-axis, LinearScale = FALSE, it is conventional to use a log scale with microbiological counts. Figure 5-11 shows the scatter plots generated by using the code.

Box 5-8: Example for filtering the U_Data for Coliforms at three locations

```
D_A_coliforms <- U_Data %>% filter(Sample_Point=="TDH010"|
Sample_Point=="TAM010"| Sample_Point=="TDV006", Determinand =="Coliforms")
#find start date of data
S_date <- min(D_A_coliforms$Sample_Date)
#find start date of data
E_date<-max(D_A_coliforms$Sample_Date)
#Graph uses the variable name SampleDate
#Must be date only
D_A_coliforms$SampleDate <-lubridate::as_date(D_A_coliforms$Sample_Date)
g3 <- Magic_TimeLine(D_A_coliforms,
S_date, E_date,
TimeBreak = "2 year", ParaColour ="aquamarine4", Gtitle ="Coliforms",
Gcaption="Notelogscaleandzerovalues", Units = "MPN/100mL",
LinearScale=FALSE, Regression=FALSE, Alpha = 0.2,
Size = 3)
gridExtra::grid.arrange(g3)
```

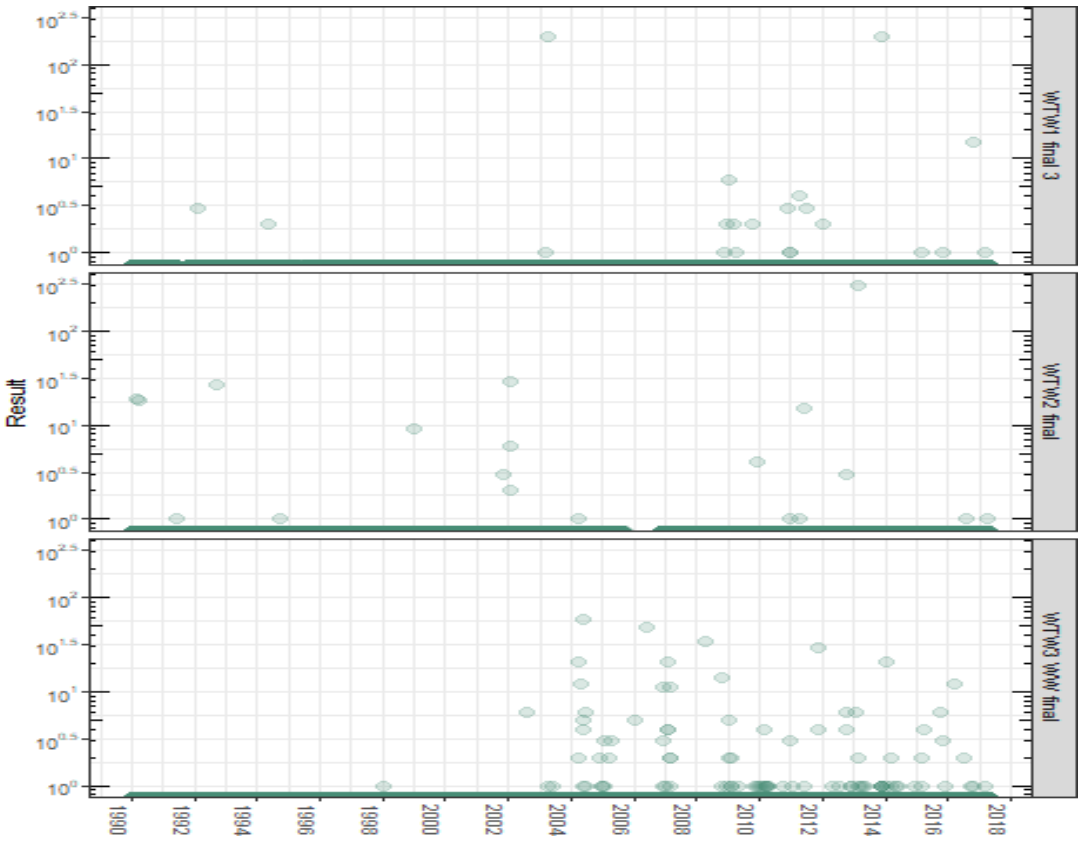

Figure 5-11: Coliforms three WTW

5.3.6 Trend Lines

A linear trend line applied to a scatter plot, see Box 5-9 and Figure 5-12. It is a simple way of visualising whether values are increasing or decreasing over time.

Box 5-9: Example for applying a trend line to a scatter plot

```
Size = 3)
gridExtra::grid.arrange(g4)
#Filter
D_turbidity <- U_Data %>% filter(Sample_Point == "TDV006",
Determinand == "Turbidity OS", Sample_Date>lubridate::ymd("20100630"))
#find start date of data
S_date <- min(D_turbidity$Sample_Date)
#find start date of data
E_date <- max(D_turbidity$Sample_Date)
#Graph uses the variable name SampleDate
#Must be date only
D_turbidity$SampleDate <-lubridate::as_date(D_turbidity$Sample_Date)
g4 <- Magic_TimeLine(D_turbidity,
S_date, E_date,
TimeBreak = "1 year", ParaColour ="aquamarine4", Gtitle = "Turbidity", Gcaption = "",
Units = "ntu", LinearScale=TRUE, Regression=TRUE, Alpha = 0.2,
```

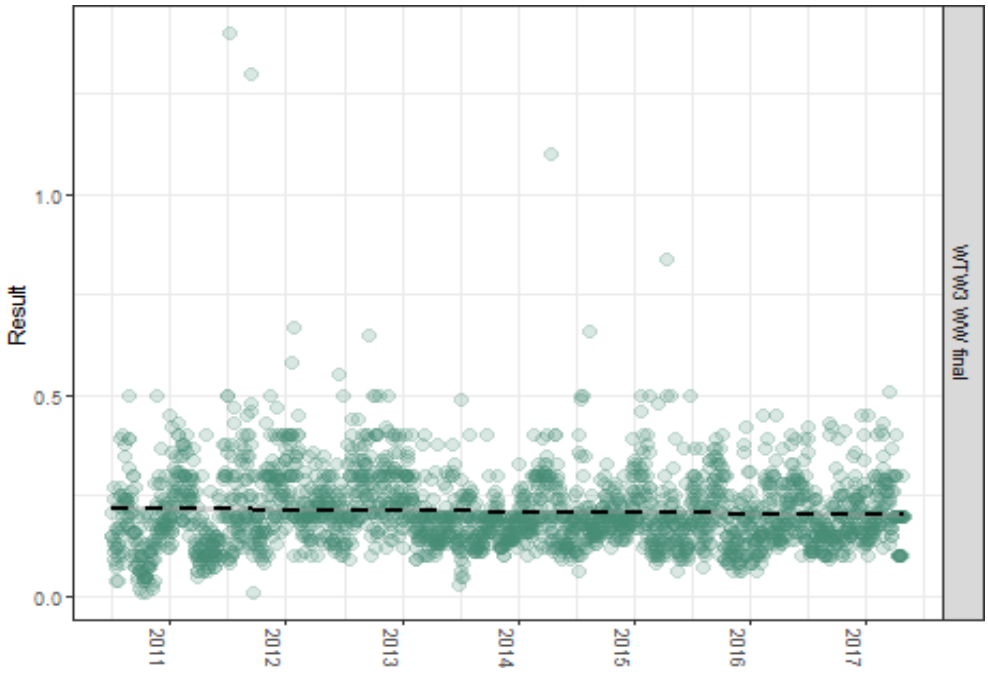

Figure 5-12: Turbidity trend at WTW3 final

5.3.7 Determinand Grouped for a Period of Time

Another way of showing trends is to group the data for a specific determinand into a period of time (e.g. month, season or year) and to illustrate the information as a boxplot (see Box 5-10). The lubridate package can be used to create a month or year variable from the U_Data$Sample_Date variable, using the functions month and year respectively. The code below makes Month and MonthFull and Year variables from U_Data$Sample_Date; the results can be seen in Table 5-1, a random selection of observations from U_Data.

Box 5-10: Example for grouping determinands for a period of time

```
#Create Month
U_Data$Month <- lubridate::month(U_Data$Sample_Date, label = TRUE)
#Create Month as full name
U_Data$MonthFull<-lubridate::month(U_Data$Sample_Date,label=TRUE,abbr=FALSE)
#Create Year
U_Data$Year <- lubridate::year(U_Data$Sample_Date)
```

Table 5-1: Data selected at random showing sample date, month and year

Determinand	Sample_Date	Month	MonthFull	Year
Cl2 (F) OS	1999-08-15	Aug	August	1999
E.coli	2017-03-27	Mar	March	2017
CC 21	1990-04-10	Apr	April	1990
Appearance	2004-06-27	Jun	June	2004
Cl2 (T) OS	2013-07-17	Jul	July	2013
Turbidity	1997-12-04	Dec	December	1997
CC 21	1994-07-27	Jul	July	1994
Turbidity	2011-11-12	Nov	November	2011
Temperature OS	2001-05-17	May	May	2001
E.coli	1991-03-31	Mar	March	1991

When plotting months, it is easier to understand the graph if the months run consecutively as they do with time; the months follow a set order. To do this for the U_Data$Month variable, it is made a factor and given a set of month_levels as they would normally occur in the year; as shown in the code below (Box 5-11). Box 5-12 shows use of the function Magic_BoxMonth to produce the boxplot in Figure 5-11 and shows the WTW2 Final Temperature (°C) monthly.

Box 5-11: Example for grouping determinands by months

```
# Month and Year Factors
month_levels<-c("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec")
U_Data$Month <- factor(U_Data$Month, levels = month_levels)
# Similarly for the fully named month
month_levelsFull <- c("January", "February", "March", "April", "May", "June", "July", "August", "Septem
U_Data$MonthFull <- factor(U_Data$MonthFull, levels = month_levelsFull)
```

Box 5-12: Example for showing temperature monthly

```
Temp <- U_Data %>%
filter(Sample_Point == "TAM010", Determinand == "Temperature OS")
g5 <- Magic_BoxMonth(Temp,
ParaColour = "dodgerblue1",
Gtitle="WTW2FinalTemperature", Gcaption = "" ,
Units=expression("Temperature"*~degree*C), LinearScale = T)
gridExtra::grid.arrange(g5)
```

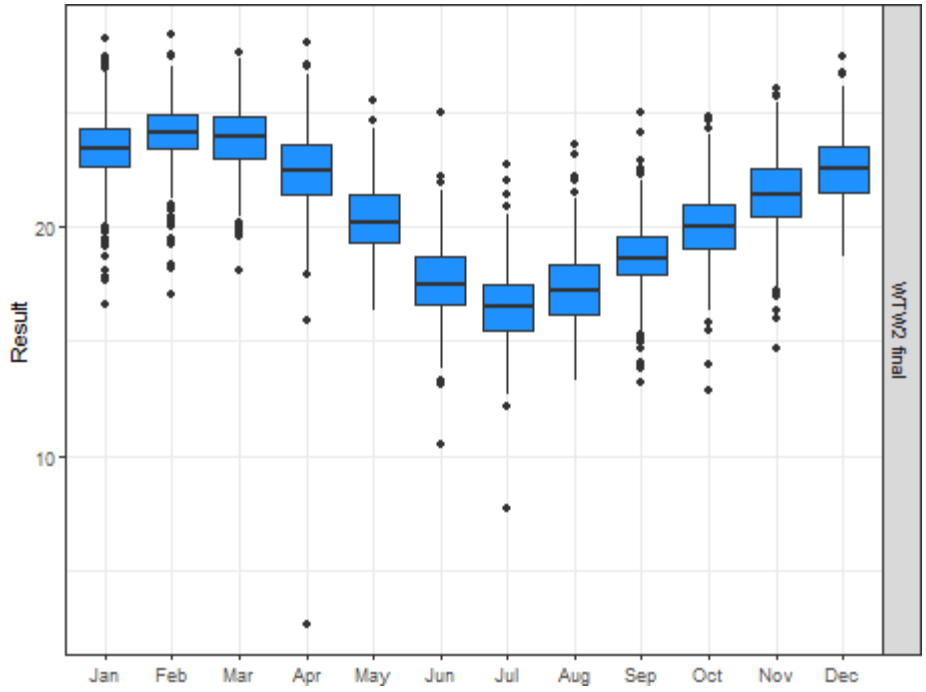

Figure 5-13: WTW2 Final Temperature

5.3.8 Interactive Time Series Graphs

Figure 5-14 shows an interactive time series graph that can be zoomed to show specific time periods and where the mouse can hover on individual values (only in the HTML version of the document). The graphs are created from the dygraphs package and can be used to explore water quality monitoring data and are also embedded within R Markdown documents and published to an html report such as this manual; detailed information on dygraphs can be found at https://rstudio.github.io/dygraphs/.

Dygraphs automatically plots xts time series objects (or any object convertible to xts); so first of all an xtsobject is created from the U_Data by selecting a determinand, for example Fe, the Sample_Date and filtering for the Results. The filtered data is then made into two vectors the Date and Result using the functions; DateVector and ResultVector respectively; these are combined by xts into a time series object FeTS. FeTS is the time series object created above. The Dygraphs code that creates Figure 5-14 is shown below (Box 5-13).

Box 5-13: Example for creating Dygraphs

```
#filter Fe data from WTW2 Final
filtered_data <- U_Data %>%
        filter(Sample_Point == "TAM010", Determinand == "Fe")
# Make two vectors
TheDate <-DateVector(filtered_data)
Result<-ResultVector(filtered_data)
#combined two vectors into a time series object
FeTS <- xts(x=Result, order.by=TheDate)
```

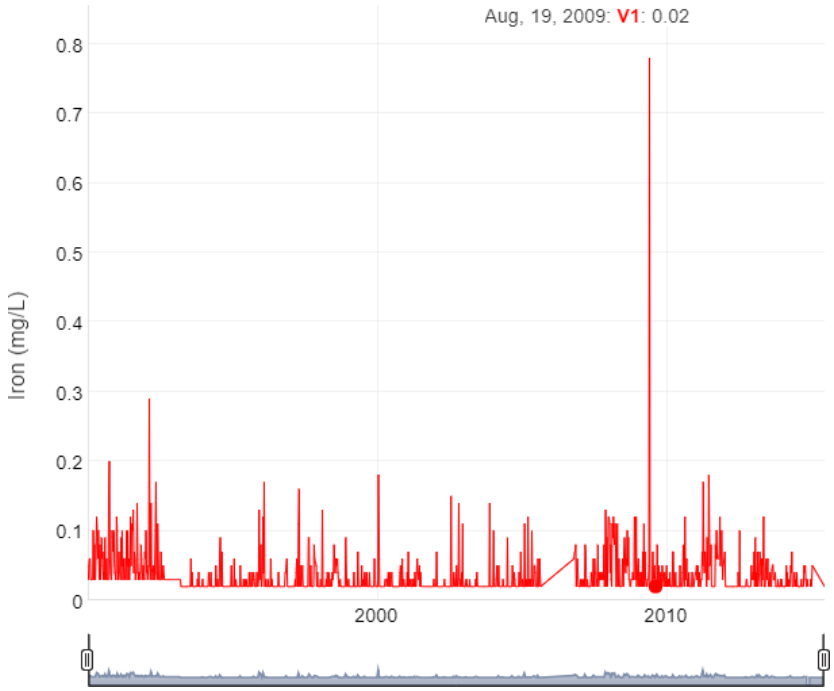

Figure 5-14: Interactive time series of Iron for WTW2

CHAPTER 6: DATA MAGIC WORKSHOP TUTORIAL 1

6.1 INTRODUCTION

This chapter provides a description of data science and the content and code for the workshops presented in Durban on the 23rd October 2018 and Pretoria on the 25th October 2018. Please refer to Annexure A for workshop tutorial material.

6.2 DATA SCIENCE REVOLUTION

The discipline of Data Science has a *beginning* with the space shuttle *Challenger* disaster of 28th January 1986. The day before the launch, 27th January, Morton Thiokol who supplied solid rocket motors to NASA for the space shuttle, recommended that NASA delay the launch due to concerns that the cold weather forecast for the next day's launch would endanger the rubber O-rings that held the rockets together. A conference call of over two hours ensued between Morton Thiokol and NASA. The descriptive results for the performance of O-rings on previous launches were discussed with the aid of diagrams (Figure 6-1) **none of which were graphical**; two of the more informative *charts* are reproduced in Figure 6-2. Due to a lack of persuasive evidence, the Morton Thiokol engineers' recommendation was overruled by NASA; the launch proceeded on schedule. The O-rings failed as the engineers had dreaded, 73 seconds after launch; all seven astronauts on board died.

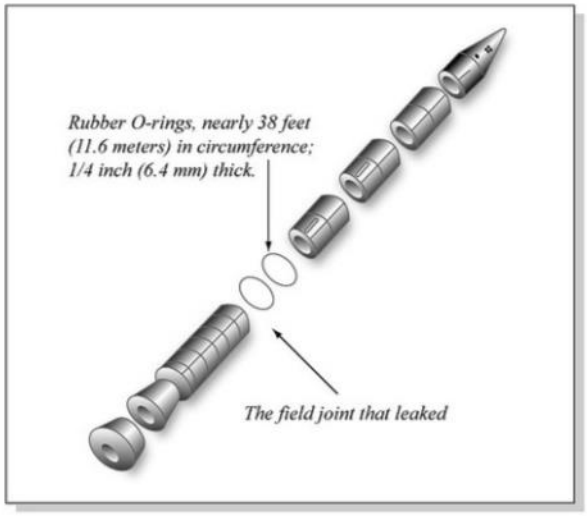

Figure 6-1: Solid Rocket Motors and O-ring

HISTORY OF O-RING TEMPERATURES (DEGREES - F)

MOTOR	MBT	AMB	O-RING	WIND
DM-1	68	36	47	10 MPH
DM-2	76	45	52	10 MPH
QM-3	72.5	40	48	10 MPH
QM-4	76	48	51	10 MPH
SRM-15	52	64	53	10 MPH
SRM-22	77	78	75	10 MPH
SRM-25	55	26	29 27	10 MPH 25 MPH

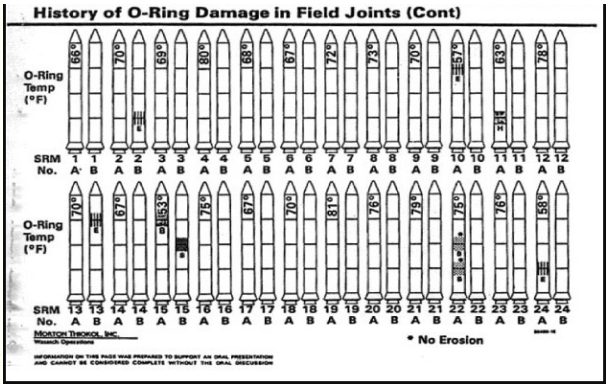

Figure 6-2: Two charts from the conference call

Morton Thiokol, the O-ring engineers had correctly interpreted the data that the rubber O-rings became brittle in low temperatures; their failure – one with horrific consequences – was not to present the data to NASA in a convincing manner. A conclusion reached by the Rogers Commission set up to study the disaster: https://spaceflight.nasa.gov/outreach/SignificantIncidents/assets/rogers_commission_report.pdf.

Figure 6.3 shows the graphic not produced for the conference call between Morton Thiokol and NASA; a graphical representation of the number of O-rings damaged (y-axis) and temperature °C (x-axis), for all the previous 23 successful launches, the temperature forecast for the next day's launch is also shown.

Figure 6-3 has been adapted from one produced by Edward Tufte for the Rogers Commission. The graph clearly shows the relationship of O-ring damage and temperature.

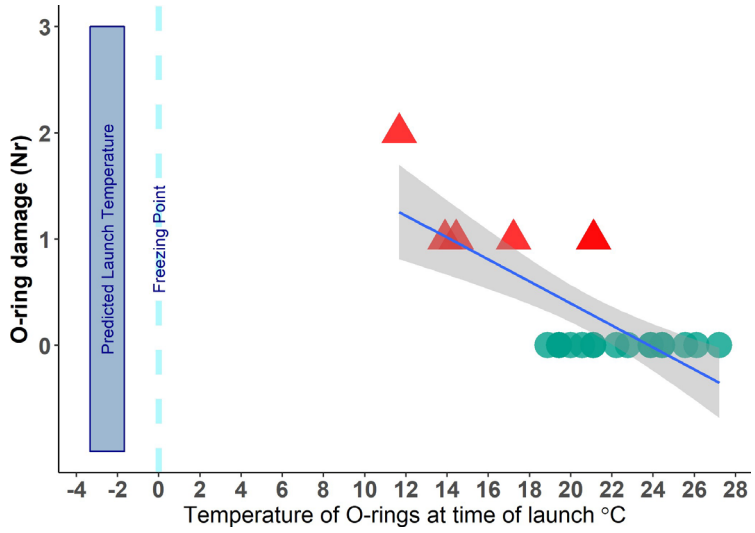

Figure 6-3: The Graphic NOT produced for the Conference Call

NASA and the O-ring engineers (Morton-Thiokol) did not apply **Data Science**. Data science is more than just statistics and encompasses:

- visualisation
- computer science
- information science
- data mining
- machine learning

and is best described as a graphic reproduced in Figure 2-2. This document is aimed at exploring how to apply *Data Science* to water quality monitoring data.

6.3 TOOLS FOR DATA SCIENCE

There are many tools to undertake *data science* Excel is familiar, others are less well known, such as *Matlab* and *SAS*; Figure 6-4 helps to put these and other tools into perspective by assessing their ease of use (y-axis) and capability (x-axis). Excel is friendly to use but not very capable at applying the skills to do data science; SAS is capable but less user friendly than Excel and a lot more expensive; R is very capable but less user friendly than Excel although it is open source (free). R is the most common choice by businesses as the tool to undertake data science; refer to http://www.business-science.io/business/2017/12/ 27/six-reasons-to-use-R-for-business.html.

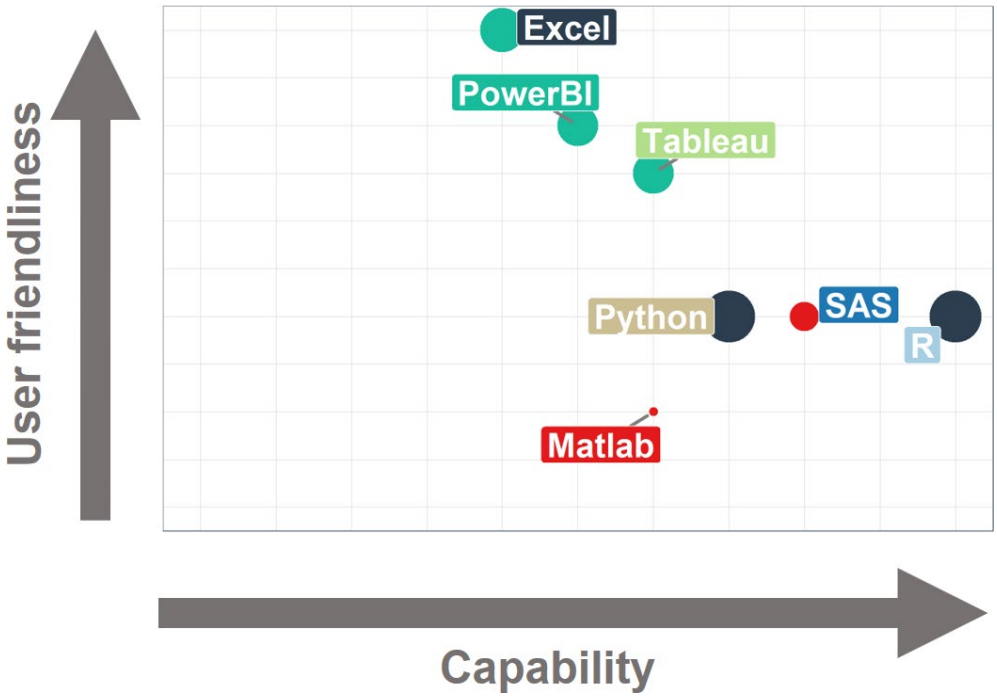

Figure 6-4: Tools for Data Science

6.4 THE RATIONALE FOR CHOOSING R

R is a very capable business orientated software, an open-source programming language and software environment for statistical computing and graphics that is supported by the R Foundation for Statistical Computing. Others advantages of R include;

- Open source (free)
- Operates on Windows/Mac/Linux
- RStudio (open source version) has made R flexible and capable
- Many users: NASA, Google, Microsoft and most of academia
- Opens up the analysis of data
- data wrangling
- visualisation
- ability to handle censored data
- Allows access to new techniques
- machine learning (e.g. Caret & e1071 packages)
- deep learning (e.g. Keras package)
- Makes data analysis reproducible through R Markdown

Much of the ability of R is derived from its packages (14,000+); these add real value to the data science being applied. The packages are developed by a wide variety of contributors and are most commonly provided as a General Public Licence (GPL).

The tidyverse package is a collection of packages useful to data science (e.g. loading/manipulating data and plotting graphs) fully described at https://www.tidyverse.org/. How to apply the tidyverse packages is also illustrated in the "R for Data Science Book" by Garrett Grolemund & Hadley Wickham the whole book is published online. http://r4ds.had.co.nz/index.html. Making data analysis reproducible and transparent is one of the key aspects of R and is best achieved through using R Markdown with RStudio. The workflow of producing reports and documents is hindered by the process of copying and pasting between different software; a process recognised by the UK Government https://dataingovernment.blog.gov.uk/ and illustrated in Figure 6-5. Workflow is greatly improved with the use of R Markdown which removes the need to copy and paste between software; the process recognised by the UK Government https://dataingovernment.blog.gov.uk/ and illustrated in Figure 6-6.

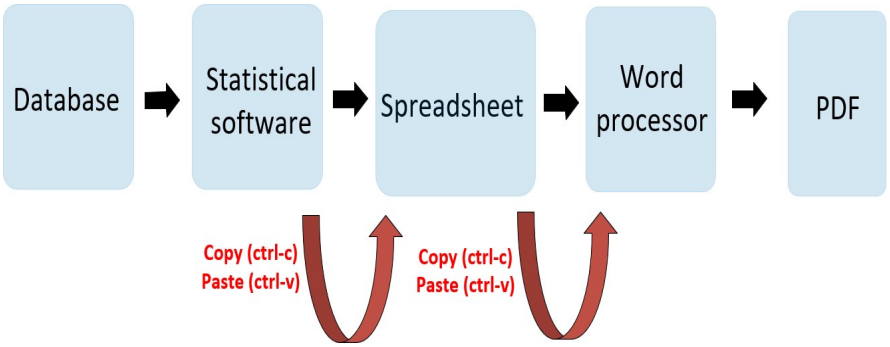

Figure 6-5: Conventional Workflow

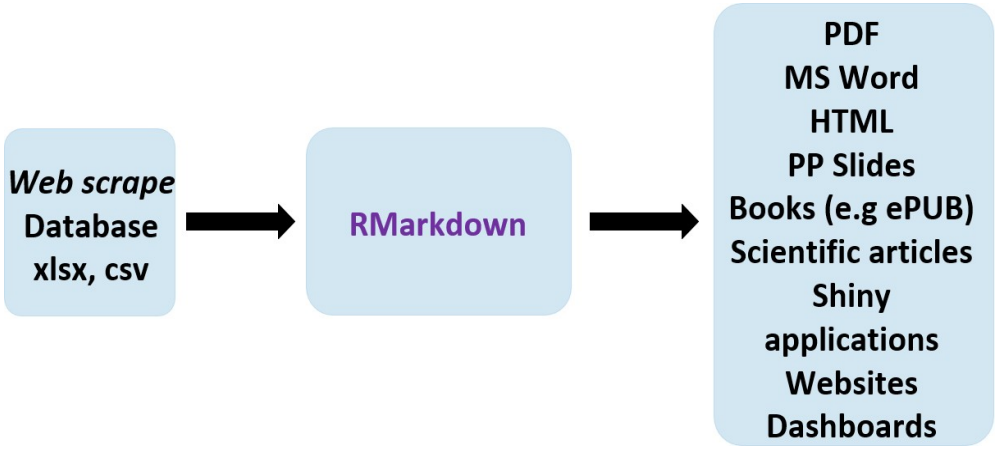

Figure 6-6: Workflow with R Markdown

This R Markdown workflow process is called literate programming (the term was coined by Donald Knuth https://www-cs-faculty.stanford.edu/~knuth/lp.html) and is a way of combining code and text into one document so it is readable; this manual was written in R Markdown. R Markdown can output to various formats including html, pdf or Word. Using R Markdown is recommended for undertaking Data Science with water quality monitoring data. Section 6.4 below uses an R Markdown template and provides introductory guidance; so new users can get started in applying Data Science to the water quality monitoring data.

6.5 MAGIC DOCUMENTS WORKSHOP

This section details the preparation required before attending the magic workshop, or undertaking the workshop without attending. It also describes how to get started with R/RStudio/R Markdown by looking at water quality monitoring data; an example dataset magic_WTW.csv is provided to practise on. The code can then be applied to the tidy example dataset.

6.5.1 Step One

a. Download and install the latest version of R (R is regularly updated) https://cran.r-project.org/bin/windows/base/. Download the version for your operating system; R can be downloaded for Windows, Mac & Linux.
b. Use R through RStudio. Download and install the latest version of RStudio from their web page https://www.rstudio.com/products/rstudio/#Desktop. Download and install the free desktop version.

6.5.2 Step Two

a. Create a folder, in a suitable place on your computer and give it a name you understand and can find again, e.g. magic-workshop-Oct-18.
b. Inside the folder created in (1) create a new folder called Data.
c. Copy (or save) the csv data file magic_WTW.csv into the Data folder.
d. Copy (or save) the Rmd file magic_template.Rmd into the folder created in (1) (not the Data folder).
e. Copy (or save) the R file magic_functions.R into the folder created in (1) (not the Data folder).

6.5.3 Step Three

a. On your computer desktop click on  and run RStudio. In the RStudio menu select file\Open File and navigate to and select the magic_template.Rmd located in the folder created in Step Two (1); press open (find this in the bottom right of the window).
b. Click the 

button – a document should generate (when run for the first time it will take some time to load all the packages!!); it should eventually produce a file like the one shown in Figure 6.14 and you are now ready to look at water quality monitoring data (if an error occurs or nothing happens re-check the steps above).

6.5.4 Notes on the magic_template.Rmd File

a. The `magic_template.Rmd` is the R Markdown template to help new users to get started on looking at water quality monitoring data.
b. The first text in the R Markdown document is the YAML header, it directs the output file `html` (it can also output to Word / PDF/ html) suggest html and how the documents looks (e.g. table of contents). The YAML begins and ends with `---` and always on its own line and looks something like this:

```
---
title:"MagicTemplate" author:"Marquis&Lord" date: "27 March 2018" output:
html_document: number_sections:yes
toc: yes
---
```

When you click the  button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document.
In magic_template.Rmd you can see embedded R code chunks like this:

```
```{r}
#Read the magic_WTW.csv file from the magic_data folder
WTW_Data <- readr::read_csv("..//magic_data//magic_WTW.csv", col_names= TRUE)
```
```

R code chunks can change how the output looks; here the two graphs (g1 & g2) are plotted to a bigger window with fig.width=9, fig.height=6.

```
```{r fig.width=9, fig.height=6}
gridExtra::grid.arrange(g1, g2, ncol=2)
```
```

Outside the code chunks there is a set of conventions for formatting plain text; the markdown part. Mark- down can be used to indicate

- bold and italic text
- lists
- headers (e.g., section titles)
- hyperlinks and much more

For more information click help on the RStudio menu and select Markdown Quick Reference There are excellent references on line at https://R Markdown.rstudio.com/ and http://r4ds.had.co.nz/r-markdown.html.

When magic_template.Rmd is knitted (by clicking the  button) the .Rmd file is rendered to a html; the beginning of the html output for magic_template.Rmd is shown in Figure 6-7.
The magic_template.Rmd can be amended (and rewritten) within RStudio using the examples from the workshop chapters of this document; for example changing the determinand to Colour. At first it is recommended that one thing at a time is changed then re-knitted
The files required to produce the Magic Documents:-

- The R Markdown template **magic_template.Rmd**
- R functions magic_functions.R
- sample csv data file called **magic_WTW.csv**

all can be found in the `magic_documents_data` folder within the Magic Manual folder.
This template should produce the output in the Figure 6-7 below when it is knitted – to do this, press the  button.

Magic Template

Marquis & Lord

27 March 2018

1 Magic WTW

1.1 Table of Determinands Measured

Show 5 entries Search:

Determinands Measured at Magic WTW

	Determinand	n	Units
1	Turbidity	7674	ntu
2	E.coli	6887	MPN/100mL
3	Cl2 (F) OS	6885	mg Cl2/L
4	Cl2 (T) OS	6882	mg Cl2/L
5	Conductivity	6878	mS/m

Showing 1 to 5 of 156 entries

Previous 1 2 3 4 5 ... 32 Next

1.2 Graph of Coliforms

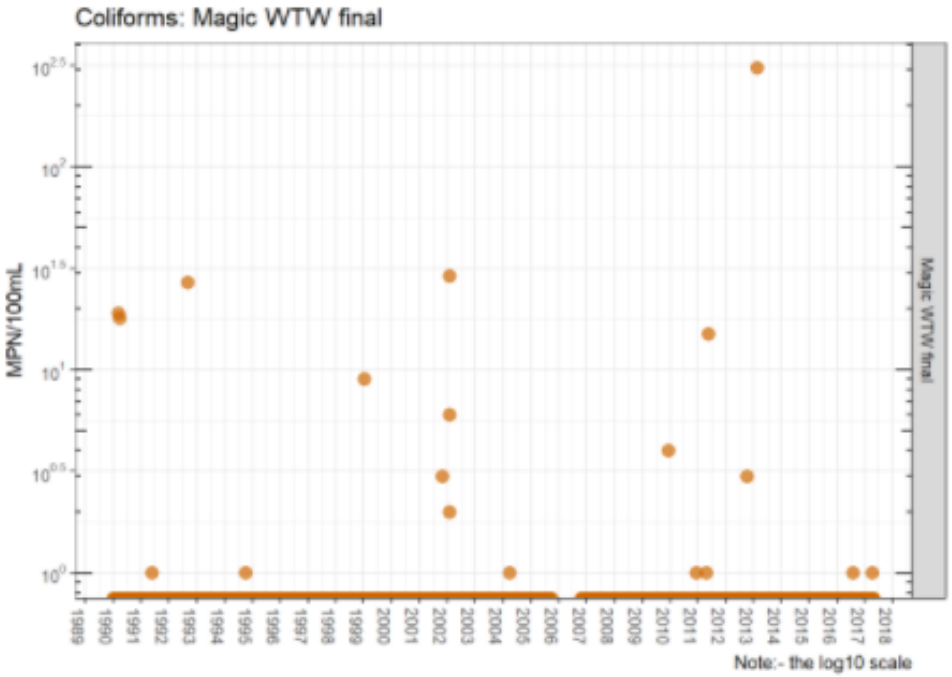

Figure 6-7: Knitted output of the magic template Rmd

CHAPTER 7: DATA MAGIC WORKSHOP TUTORIAL 2

7.1 INTRODUCTION TO R AND R MARKDOWN

This interactive Data Magic Workshop was presented in Durban, 23rd October 2018, and Pretoria 25th October 2018. Please refer to Annexure A for workshop tutorial material. The introduction covered the following;

- Download and install the latest version of R (R is regularly updated) https://cran.r-project.org/bin/windows/base/. Download the version for your operating system; R can be downloaded for Windows, Mac & Linux.
- Use R through RStudio. Download and install the latest version of RStudio from their web page https://www.rstudio.com/products/rstudio/#Desktop. Download the free desktop version.
- Go to the online version of R for Data Science by Garrett Grolemund & Hadley Wickham; the whole book is available for free. It's recommended to work through the book from the Introduction; it will introduce the `tidyverse` a collection of packages with all the essential tools for data science.
- Explore R Markdown; a full introduction to R Markdown can be found here http://RMarkdown.rstudio.com/lesson-1.html.

7.2 WORKSHOP BASICS

- On your computer desktop Click on  and run RStudio.
- From the menu choose `File\New File\R Markdown...` provide a Title, `e.g. Workshop` and Author, `i.e. your name`; leave remaining options as default.
- Save the file (e.g. with a name `Workshop.Rmd`) (in the same folder as the `watminder.csv` is located)
- In the bottom right window find the tab `packages` and select `install`; type `tidyverse` and click the install button (tidyverse – includes the packages readr,dplyr,ggplot2,tidyr andlubridate.
- Below is mono-type text on a shaded background this is R-code designed to run in `r-chunks` in the in R Markdown document, e.g. `Workshop.Rmd`.
- To create an `r-chunk`; in the top right window select the tab `insert` and select `R`.
- Type or Copy the R-code below into the r-chunk; it is recommended at first to do this in the order set out below.
- To run the code in an r-chunk click the green > in the top right corner or alternatively highlight the code and press Ctrl-enter; a section of code or a variable can be run using Ctrl-enter.
- Packages that are used in this example include:

```
library(readr)
library(dplyr)
library(ggplot2)
library(tidyr)
library(lubridate)
```

As part of this workshop and this document, a watminder dataset is included. This data is already tidy and is available for the reader to follow this tutorial step by step by using the cleaned dataset and the instructions as

provided in Workshop 1 2 and 3. First you will need to put the watminder dataset into the Rstudio environment. We need to read in the data. We do this with a function called **readr.** Type in the code as follows:

```
watminder<-readr::read_csv("watminder.csv",col_names=T)
```

This loads the watminder dataset into the Rstudio environment. Now if we select a determinand (Conductivity) we will filter the dataset for all determinands called Conductivity. We allocate the entire dataset to a new file name graph_data and then filter for conductivity as in the code below (See Figure 7-1):

```
graph_data <- watminder %>%
        filter (Determinand == "Conductivity")
```

Now we add a function called lubridate and we use the month of analysis to separate the data (**`lubridate::month variable)`** on a standard scatter graph called geom_point in Rstudio.

```
graph_data <- watminder %>%
        filter (Determinand == "Conductivity") %>%
        mutate(Month=lubridate::month(Date,label=TRUE))
        ggplot(graph_data,aes(x=Date,y=Result))+
        geom_point()
```

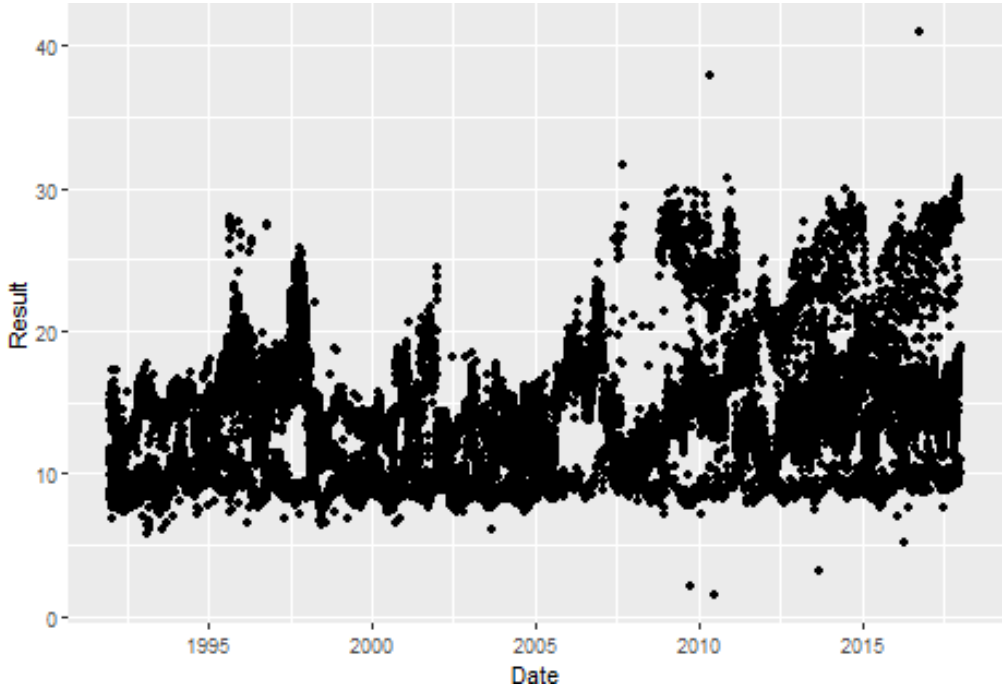

Figure 7-1: Scatter plot of conductivity

If you use the facet_wrap(~Location) function the data will separate out into different graphs (facets) based on the location (sampling point) as per the code and Figure 7-2 below.

```
ggplot(graph_data,aes(x=Date,y=Result,colour=Location))+
        geom_point() +
        facet_wrap(~Location)
```

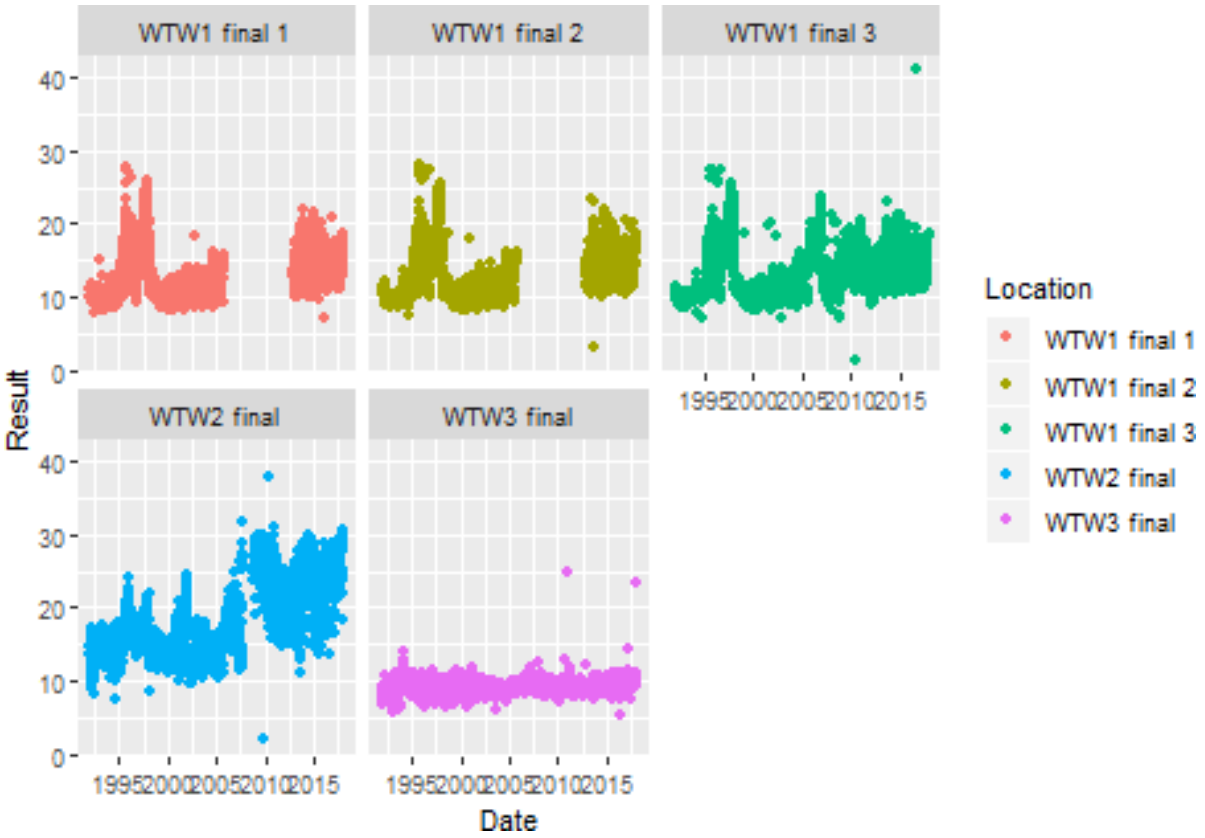

Figure 7-2: Using the facet wrap to separate the locations within the dataset

As can be seen from the figure above (Figure 7-2), the dates are not visible and the legend is unnecessary. This can be changed by allocating a month separation on the x axis and to see the data better we will change the plot from a scatter plot to a geometric box plot as follows:

So x = Month and geom_boxplot code is written as follows:

```
ggplot(graph_data,aes(x=Month,y=Result,colour=Location))+
        geom_boxplot() +
        facet_wrap(~Year)
```

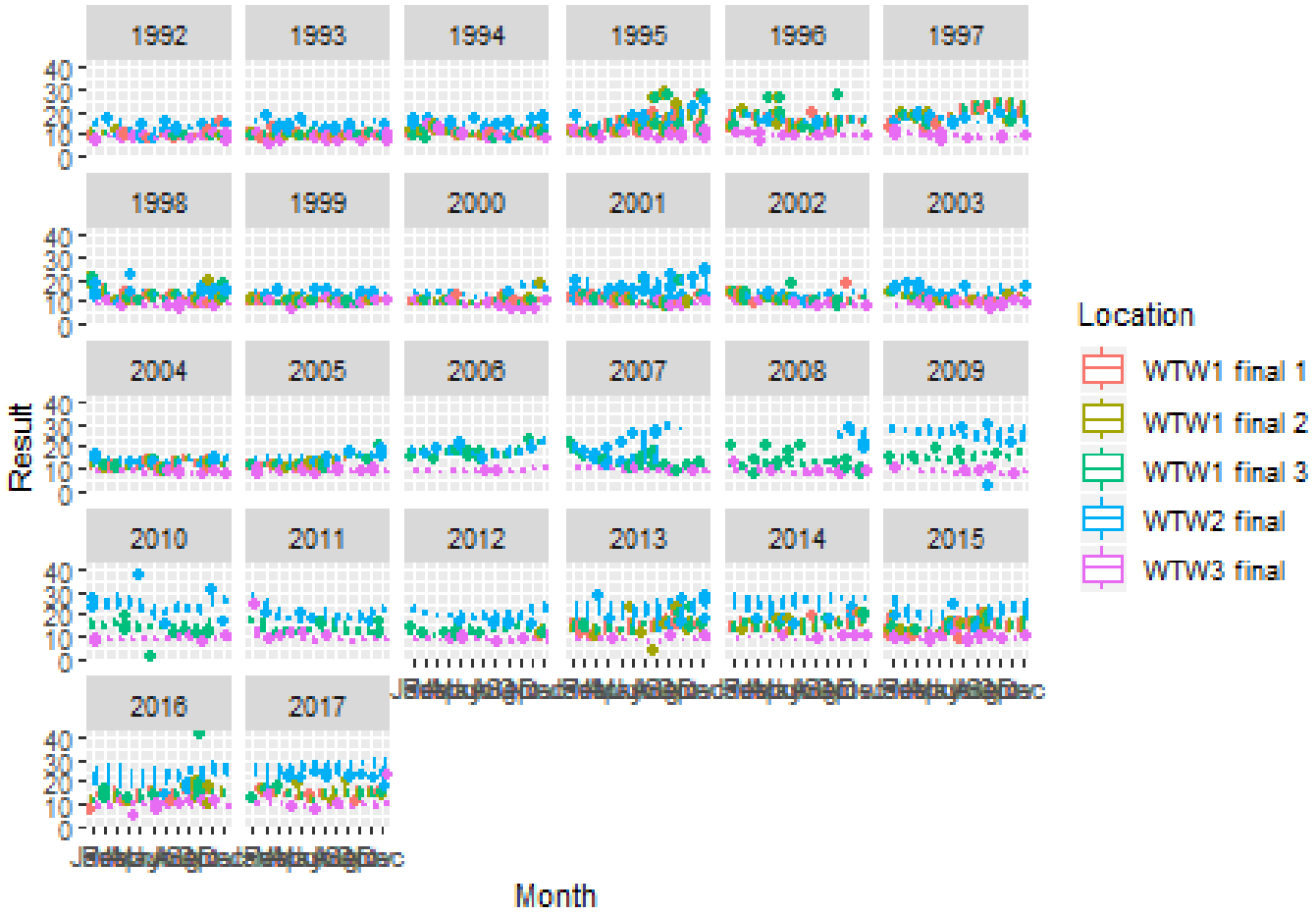

Figure 7-3: Facet wrap based on year and colours for location

We can use the `theme(..)` instruction to remove the legend and rotate axis text as in the code below:

```
ggplot(graph_data,aes(x=Month,y=Result,colour=Location))+
        geom_boxplot() +
facet_wrap(~Location) +
        theme(legend.position ="none",
axis.text.x = element_text(angle =90))
```

Note that the legend has now been removed and the text of the x axis has been turned 90° in Figure 7-4 below.

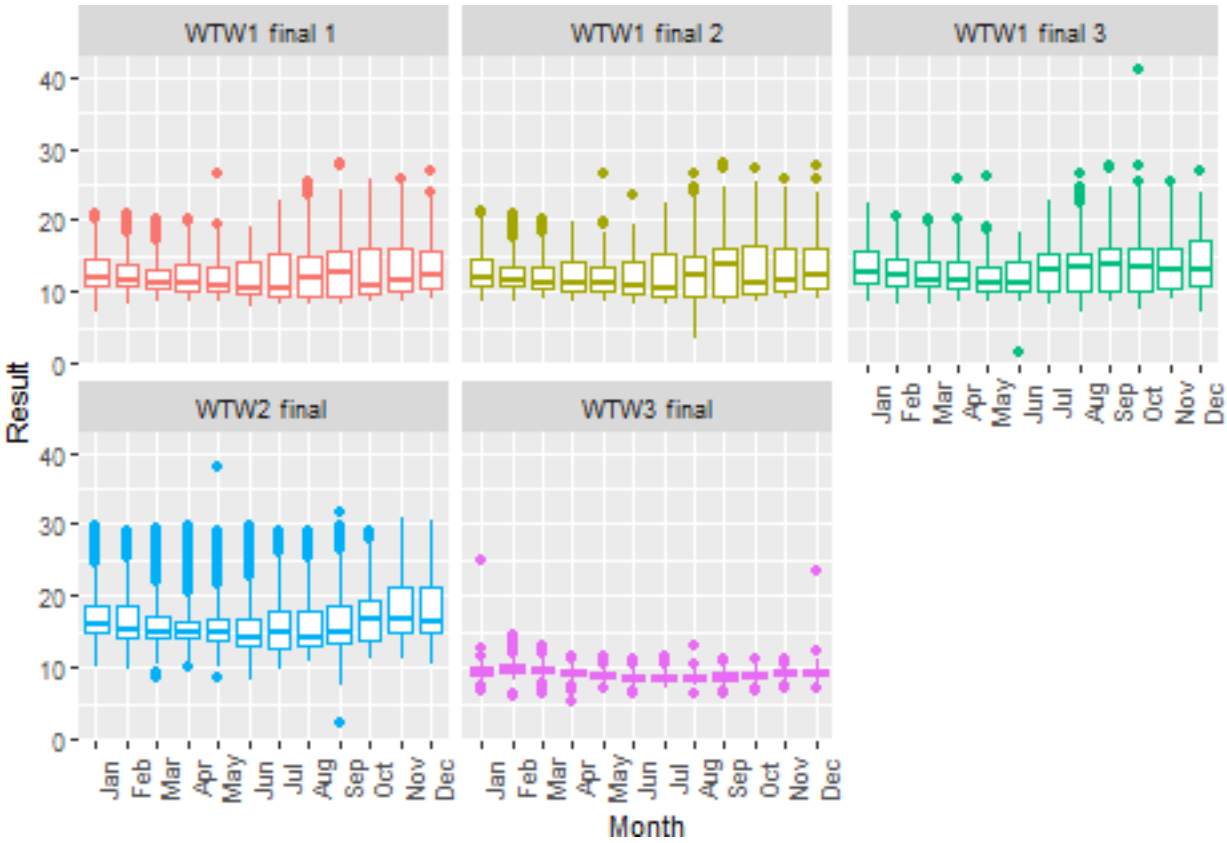

Figure 7-4: Facet wrapped box plots grouped by location

7.3 BASIC TABLES FOR A SELECTED DETERMINAND

Note that these are very basic tables and other code is available to draw more advanced tables. However, as a basic first step to allow us to draw a table for a certain determinand, such as turbidity, we first need to filter for this determinand. We do this by filtering the entire watminder dataset and then calculate the median with the following code, giving us a single result of the median for the entire turbidity set of 0.2 NTU.

```
watminder %>%
        filter (Determinand=="Turbidity")%>%
        summarise(ParmMedian =median(Result))
```

```
###A tibble: 1 x 1
##ParmMedian
##   <dbl>
##1    0.2
```

For us to see the median for turbidity it may be more helpful to group_by(Year) as follows:

```
watminder %>%
        filter(Determinand=="Turbidity")%>% group_by(Year) %>%
        #summarise(Percent95 = quantile(Result,probs=0.95))
        summarise(ParmMedian = median(Result))
```

This produces the following basic table in Rstudio:

###A tibble: 26 x 2

##Year ParmMedian

##<dbl> <dbl>

##	1	1992	0.16
##	2	1993	0.19
##	3	1994	0.16
##	4	1995	0.2
##	5	1996	0.16
##	6	1997	0.155
##	7	1998	0.24
##	8	1999	0.2
##	9	2000	0.21
##	10	2001	0.2

... with 16 morerows

To have a better look at the data we decide to group it by year and location simultaneously, using the group_by(Year, Location) code as shown below:

```
watminder %>%
        filter(Determinand=="Turbidity")%>%
        group_by(Year, Location) %>%
        summarise(ParmMedian =median(Result))
```

This produces a table that looks like this:

###A tibble: 130 x 3

###Groups:Year[?]

YearLocation ParmMedian

##<dbl><chr><dbl>

##	1	1992	WTW1	final	1	0.14
##	2	1992	WTW1	final	2	0.14
##	3	1992	WTW1	final	3	0.12
##	4	1992	WTW2	final		0.16
##	5	1992	WTW3	final		0.23
##	6	1993	WTW1	final	1	0.2
##	7	1993	WTW1	final	2	0.18
##	8	1993	WTW1	final	3	0.16
##	9	1993	WTW2	final		0.18
##	10	1993	WTW3	final		0.23

###...with 120 morerows

In order to visualise the data better we need to spread the data over the page and we spread it based on the Location (sample point) and the statistic (`spread(Location, Statistic)`).

```
watminder %>%
        filter(Determinand=="Turbidity")%>%
        group_by(Year, Location) %>%
        summarise(Statistic=median(Result))%>%
        spread(Location, Statistic)
```

```
## # A tibble: 26 x6
###Groups:Year [26]
##    Year `   WTW1 final 1`  `WTW1 final 2`  `WTW1 final 3`  ` WTW2 final`
```

##	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
##	1 1992	0.14	0.14	0.12	0.16
##	2 1993	0.2	0.18	0.16	0.18
##	3 1994	0.16	0.15	0.13	0.22
##	4 1995	0.18	0.18	0.17	0.17
##	5 1996	0.12	0.13	0.12	0.23
##	6 1997	0.1	0.11	0.12	0.24
##	7 1998	0.21	0.23	0.21	0.25
##	8 1999	0.19	0.18	0.17	0.21
##	9 2000	0.18	0.17	0.17	0.26
##	10 2001	0.17	0.19	0.17	0.24

```
## # ... with 16 more rows, and 1 more variable: `WTW3 final` <dbl>
```

Now the watminder dataset is assigned to Table_Data and the knitr function is used to generate a table of the relevant data knitr::kable(Table_Data)>. We filter for the determinand Colour (Determinand == "Colour"> and turbidity. The data is then summarised per year and location and the 95th percentile is summarised using the summarise (Statistic = quantile(Result, probs=0.95)) function as per this code:

```
Table_Data <- watminder %>%
        #filter (Determinand == "Colour") %>%
        filter(Determinand=="Turbidity")%>%
        group_by(Year, Location) %>%
        summarise(Statistic=median(Result))%>%
        #summarise(Statistic = quantile(Result, probs=0.95)) %>%
        spread(Location, Statistic)
knitr::kable(Table_Data)
```

This code then produces a table of 95th percentiles summarised per location and year for colour and turbidity.

Table 7-1: Turbidity and colour for WTW 1 2 and 3

Year	WTW1 final 1	WTW1 final 2	WTW1 final 3	WTW2 final	WTW3 final
1992	0.14	0.140	0.120	0.160	0.230
1993	0.20	0.180	0.160	0.180	0.230
1994	0.16	0.150	0.130	0.220	0.160
1995	0.18	0.180	0.170	0.170	0.270
1996	0.12	0.130	0.120	0.230	0.230
1997	0.10	0.110	0.120	0.240	0.280
1998	0.21	0.230	0.210	0.250	0.330
1999	0.19	0.180	0.170	0.210	0.290
2000	0.18	0.170	0.170	0.260	0.290
2001	0.17	0.190	0.170	0.240	0.220
2002	0.15	0.150	0.140	0.230	0.220
2003	0.14	0.140	0.130	0.270	0.260
2004	0.17	0.170	0.170	0.270	0.260
2005	0.25	0.250	0.240	0.260	0.330
2006	0.21	0.200	0.190	0.210	0.240
2007	0.27	0.260	0.245	0.260	0.260
2008	0.26	0.240	0.230	0.220	0.210
2009	0.28	0.220	0.230	0.220	0.240
2010	0.16	0.110	0.110	0.200	0.155
2011	0.21	0.160	0.160	0.250	0.180

Year	WTW1 final 1	WTW1 final 2	WTW1 final 3	WTW2 final	WTW3 final
2012	0.26	0.205	0.190	0.315	0.240
2013	0.23	0.190	0.170	0.320	0.250
2014	0.22	0.190	0.190	0.270	0.260
2015	0.20	0.170	0.190	0.310	0.180
2016	0.22	0.200	0.200	0.300	0.210
2017	0.30	0.200	0.200	0.300	0.200

Now that we have a table we will plot the year on the x axis (x=Year) and we will only plot the data for WTW2 (y = WTW2 final) and geom_smooth(method = "lm") to obtain a linear trend shading area.
The code and the graph looks like this:

```
ggplot(Table_Data,aes(x=Year,y=`WTW2final`))+
        geom_point() +
        geom_smooth(method = "lm")
```

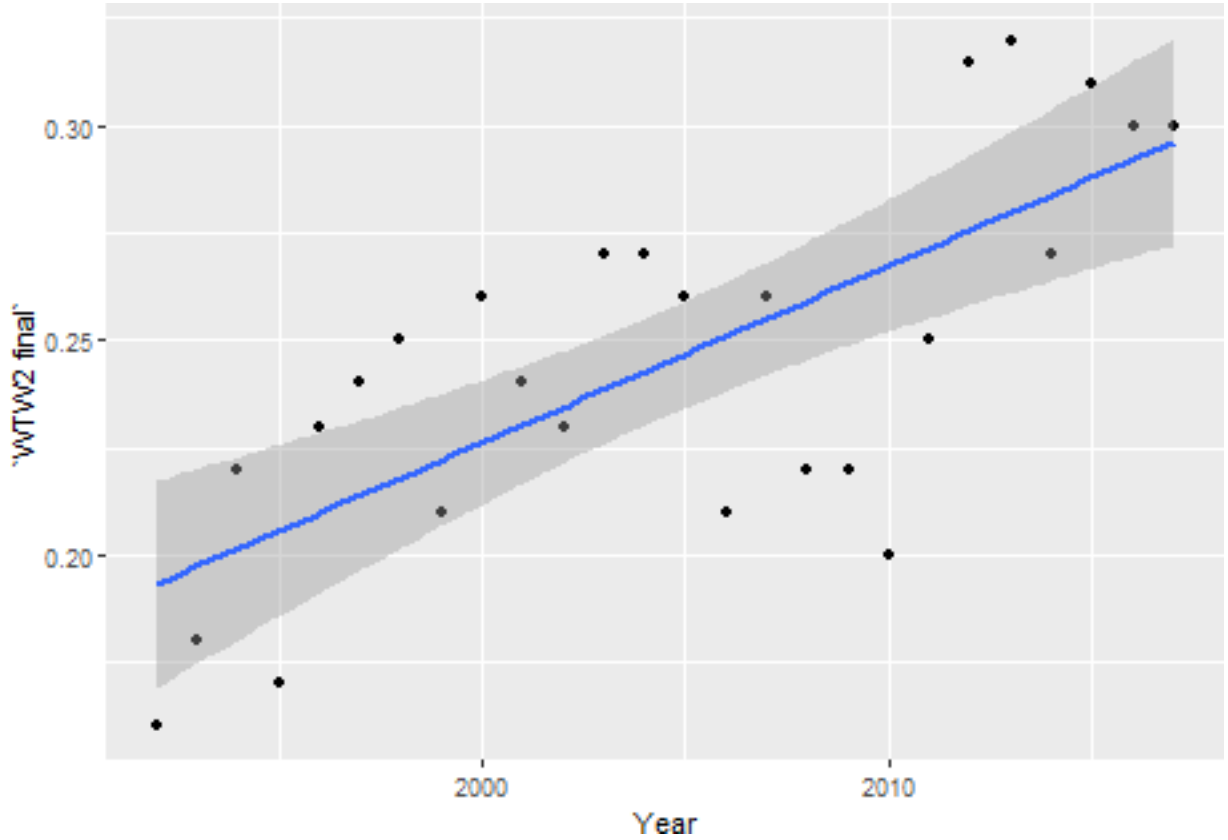

Figure 7-5: Coliform data with trend line

7.4 MICROBIOLOGICAL DATA – COLIFORMS

To look at a specific set of determinands, in this case, coliforms, we must filter the dataset **`watminder`** for the Coliform determinand

```
graph_data <- watminder %>%
        filter (Determinand == "Coliforms")
```

7.5 GRAPH OF COLIFORM DATA

To draw a scatter plot of this selected data, the **`ggplot`** function is used. Since this is microbiological data, the **`scale_y_log10()`** instruction makes the scale a logarithmic scale (log graph).

```
ggplot(graph_data,aes(x=Date,y=Result,colour=Censored))+
        geom_point() +
        scale_y_log10() +
        facet_wrap(~Location)
```

This provides the following output as seen in Figure 7-6. Note the **`facet_wrap`** code which separates the data based on location.

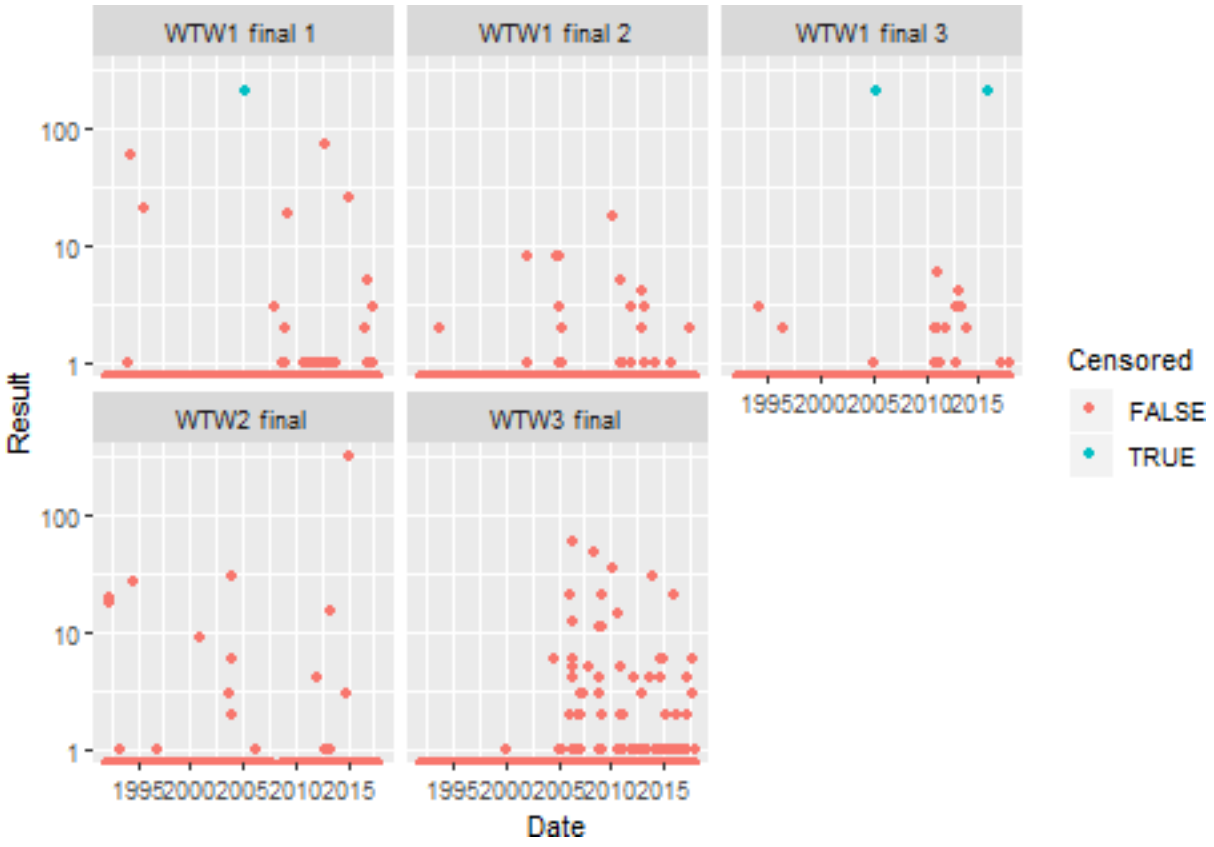

Figure 7-6: Graph of coliform data with colour

7.6 GRAPH OF COLIFORM DATA WITH COLOUR

To draw a graph of the coliform data for WTW3 we filter the watminder dataset for “Coliforms”, “WTW3 final” and Year >= 2002.

```
graph_data <- watminder %>%
        filter(Determinand=="Colour",Location=="WTW3final",Year>=2002)%>%
        mutate(Month = lubridate::month(Date, label = TRUE),
Colour5 = Result < 5)
```

7.7 GRAPHS WITH ANNOTATION

Now we include a line on the graph and colour it red to ensure that the water quality standard appears on the graph to show non-compliance, where appropriate. We do this with the code:
” `geom_hline(yintercept = 5, colour = "red")` as shown below and in Figure 7-7:

```
ggplot(graph_data,aes(x=Month,y=Result))+
        geom_boxplot() +
        geom_hline(yintercept=5,colour="red")+
        facet_wrap(~Year) +
        theme(axis.text.x = element_text(angle =90))
```

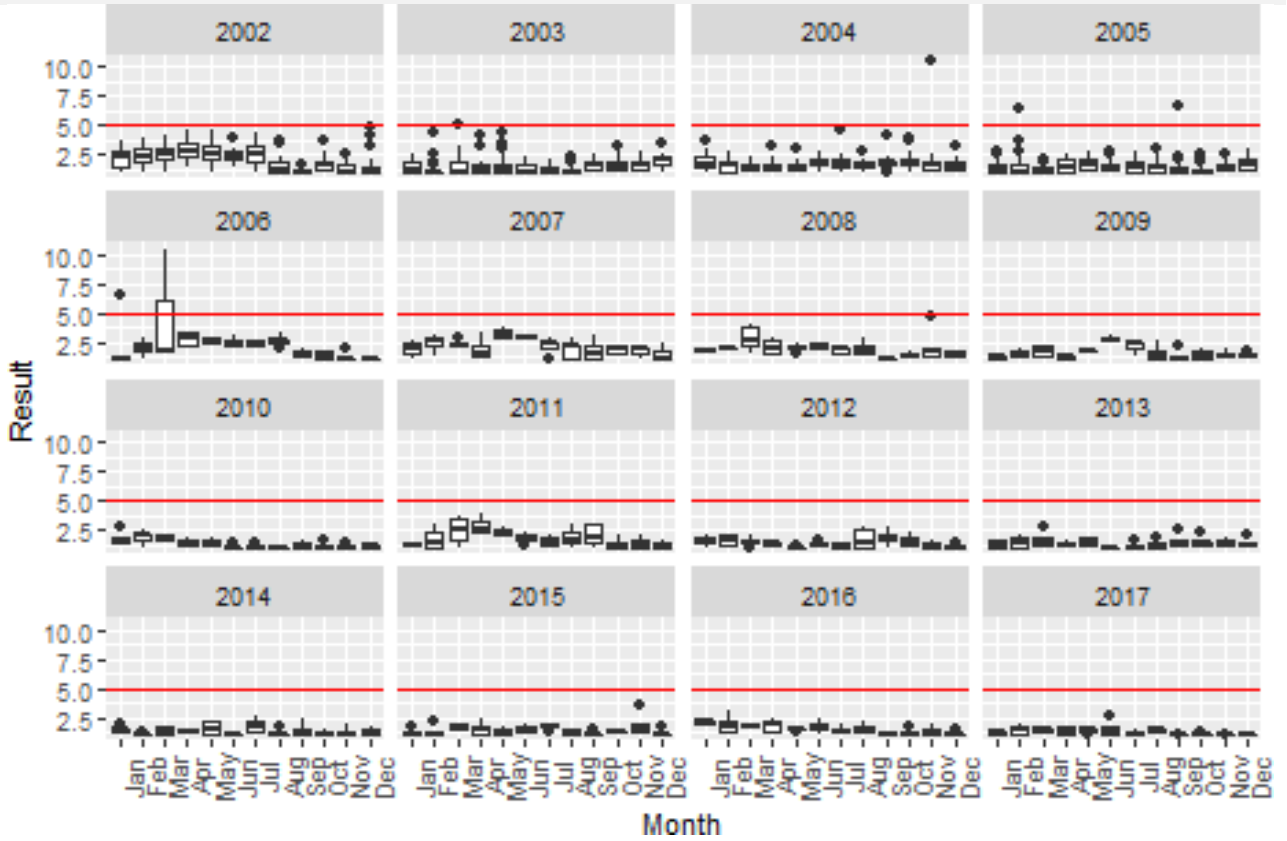

Figure 7-7: Box and whisker plots for coliform data for 15 years

7.8 PRESENTATION GRAPHS

Occasionally the data needs to be presented in a way to show outlier data or non-compliant data. The code library(scales) is required and this creates the x axis on a 1 year scale as follows: scale_x_date(breaks = date_breaks("1year"),labels=date_format("%Y")). See the code and Figure 7-8 below:

```
library(scales)
ggplot(graph_data,aes(x=Date,y=Result,colour=Colour5))+
        geom_point() +
        scale_x_date(breaks=date_breaks("1year"),labels=date_format("%Y"))+
        labs(title = "Colour 2002-2017",
y = "Colour (PtCo units)", colour="StandardAchieved")+
        geom_hline(yintercept = 5, colour = "red") +
        #facet_wrap(~Year) +
        theme_bw() +
        theme(axis.title.x = element_blank(), axis.text.x=element_text(angle=90))
```

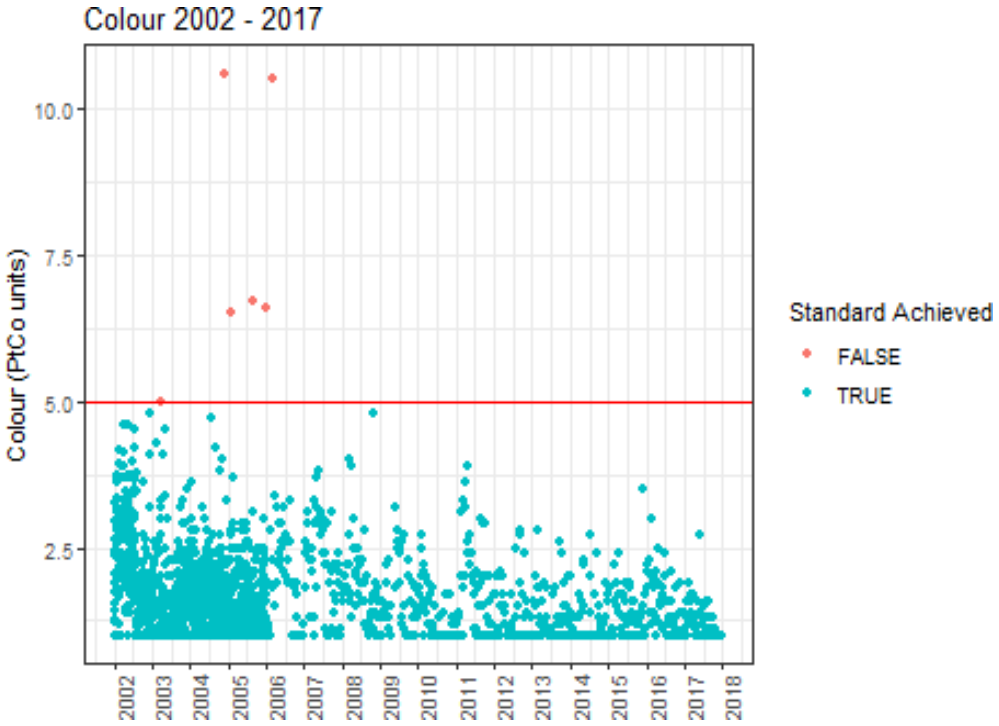

Figure 7-8: Colour for a sample point with non-compliant data points in a different colour

7.9 FURTHER RESOURCES

There are many resources available on the internet and it requires persistence and perseverance as well as a willingness to invest time and energy to follow an online course to obtain the most value from it. The way that we are learning is fundamentally changing. Our children turn to Google while we used the Encyclopaedia Britannica and the library.

7.9.1 DataCamp (Structured Learning)

There are many free resources online to learn R/R Markdown/Data Science, as any web search will show; and it is feasible to learn all you need to know through this unstructured learning. For a more structured approach, one that will go at your pace, DataCamp is recommended. Although there is a monthly/annual fee (£232 per year Sept 2018), a limited number are free, the price is very competitive when compared to a day workshop on learning R and/or Data Science; an extensive range of courses is available covering all aspects of Data Science.

7.9.2 Unstructured Learning

- RStudio has Excellent tutorials, guides, webinars and videos on all aspects of R available through their Resources Tab;
- Online Learning; including links to the free courses on DataCamp. DataCamp's free introduction to R https://www.datacamp.com/courses/free-introduction-to-r webinars; including all the RStudio conference presentations.

7.9.3 Other resources

- Cheatsheets; making it easy to learn about and use some key packages; especially the tidyverse collection.
- stackoverflow; covers more than R but is a good place to go when you have a specific coding problem.
- R-bloggers; keeps you up to date on all thing R; with many code examples.
- twitter; although will often repeat R-bloggers see #rstats, #DataScience

7.9.4 Books Online

- The online book R for Data Science, by Garrett Grolemund & Hadley Wickham, provides an Excellent grounding for using R in data science. There are, however, many other online books (published using bookdown) that show in detail how to apply the many features of R; a select list is provided below:
- R Markdown: The Definitive Guide by Yihui Xie, J. J. Allaire, Garrett Grolemund – provides a comprehensive and accurate reference to the R Markdown ecosystem. With R Markdown, you can easily create reproducible data analysis reports, presentations, dashboards, interactive applications, books, dissertations, websites, and journal articles, while enjoying the simplicity of Markdown and the great power of R and other languages.
- Geocomputation with R by Robin Lovelace, Jakub Nowosad and Jannes Muenchow – It teaches a range of spatial skills, including: reading, writing and manipulating geographic data; making static and interactive maps; applying geocomputation to solve real-world problems; and modelling geographic phenomena.
- Text Mining with R by Julia Silge and David Robinson – A guide to text analysis within the tidy data framework, using the tidytext package and other tidy tools.
- Efficient R programming by Colin Gillespie and Robin Lovelace – Efficient R Programming is about increasing the amount of work you can do with R in a given amount of time, through both computational and programmer efficiency.
- Handling Strings with R by Gaston Sanchez – provides a panoramic perspective of the wide array of string manipulations that you can perform with R. If you are new to R, or lack experience working with character data, this book will help you get started with the basics of handling strings. Likewise, if you are already familiar with R, you will find material that shows you how to do more advanced string and text processing operations.
- An Introduction to Statistical and Data Sciences via R by Chester Ismay and Albert Y. Kim – the book assumes no prerequisites: no algebra, no calculus, and no prior programming/coding experience. The

book aspires to provide a gentle introduction to the practice of analysing data and answering questions using data the way data scientists, statisticians, data journalists, and other researchers would.

- Exploratory Data Analysis with R by Roger D. Peng – This book covers the essential exploratory techniques for summarizing data with R. These techniques are typically applied before formal modelling commences and can help inform the development of more complex statistical models. Exploratory techniques are also important for eliminating or sharpening potential hypotheses about the world that can be addressed by the data you have.
- R Programming for Data Science by Roger D. Peng – The R programming language has become the de facto programming language for data science. Its flexibility, power, sophistication, and expressiveness have made it an invaluable tool for data scientists around the world. This book is about the fundamentals of R programming. You will get started with the basics of the language, learn how to manipulate datasets, how to write functions, and how to debug and optimize code.

7.9.5 R People

A lot goes on in the R community and keeping up with new developments is a challenge; fortunately, some people keep their finger on the pulse particularly on Twitter, see the following twitter handles:

- Mara Averick @dataandme
- Jenny Bryan @JennyBryan
- Mine CetinkayaRundel @minebocek
- Julia Silge @juliasilge
- Yihui Xie @xieyihui
- Hadley Wickham @hadleywickham

CHAPTER 8: DATA MAGIC WORKSHOP TUTORIAL 3

8.1 INTRODUCTION

This Chapter describes a worked example of investigating water quality monitoring data at a Magic WW; it demonstrates the processes shown in Figure 1.1. Please refer to Annexure A for workshop tutorial material. It answers the following questions:

- Is there a statistically significant trend in coliform results at WTW final water?
- Is there a statistically significant difference in coliform results at the three sample sites at WTW?

Note: – The code in this chapter focuses on Coliforms however the same code can be used for other microorganisms in the water quality monitoring data; for example Heterotrophic Plate Count or *E coli* or any other determinand, simply by substituting Coliforms with HPC or any other determinand of your choice.

8.2 MAGIC WW DATA

The water quality monitoring data for a magic WW is stored in a dataframe called magic_WW the first 10 rows of the total number of rows (370,319 rows) is displayed in Table 8-1 below.

Table 8-1: Magic WTW Data showing the first 10 rows of 370,319 rows of data

Sample_Point	Description	Sample_Date	Determinand	Result	Unit
TDH008	WTW1 final 2	1990-01-06	Coliforms	0	MPN/100mL
TDH008	WTW1 final 2	1990-01-06	Colour	1	°H
TDH008	WTW1 final 2	1990-01-06	*E. coli*	0	MPN/100mL
TDH007	WTW1 final 1	1990-01-10	Coliforms	0	MPN/100mL
TDH007	WTW1 final 1	1990-01-10	Colour	1	°H
TDH007	WTW1 final 1	1990-01-10	*E. coli*	0	MPN/100mL
TDH008	WTW1 final 2	1990-01-18	Coliforms	0	MPN/100mL
TDH008	WTW1 final 2	1990-01-18	Colour	1	°H
TDH008	WTW1 final 2	1990-01-18	*E. coli*	0	MPN/100mL
TDH007	WTW1 final 1	1990-01-19	Coliforms	0	MPN/100mL

The code for this looks like this:

```
# Just show first 1000 lines
magic_WW_part <- magic_WW %>%
        slice(1:10)
knitr::kable(magic_WW_part,
caption='MagicWTWData', booktabs = TRUE,
align = c("l", "l", "l", "r", "l"))
```

8.3 QUESTION 1 – IS THERE A STATISTICALLY SIGNIFICANT TREND IN COLIFORM RESULTS AT MAGIC WTW FINAL WATERS?

Transform data into a dataframe so just Coliforms remain.

```
filtered_coliforms<-magic_WW%>%
        filter(Determinand == "Coliforms")
```

Visualise the data with the `Magic_TimeLine`.

```
#find start date of data
S_date <- min(filtered_coliforms$Sample_Date)
#find start date of data
E_date <- max(filtered_coliforms$Sample_Date)
#Graph uses the variable name SampleDate
#Must be date only
filtered_coliforms$SampleDate <-lubridate::as_date(filtered_coliforms$Sample_Date)
g1 <- Magic_TimeLine(filtered_coliforms,
        S_date,
        E_date,
        TimeBreak = "2 year",
        ParaColour ="darkorange3",
Gtitle="Coliformsforthreefinals", Gcaption = "Note Log10 scale",
Units="MPN/100mL", LinearScale =FALSE, Regression = TRUE, Alpha = 0.3,
Size = 3)
gridExtra::grid.arrange(g1)
```

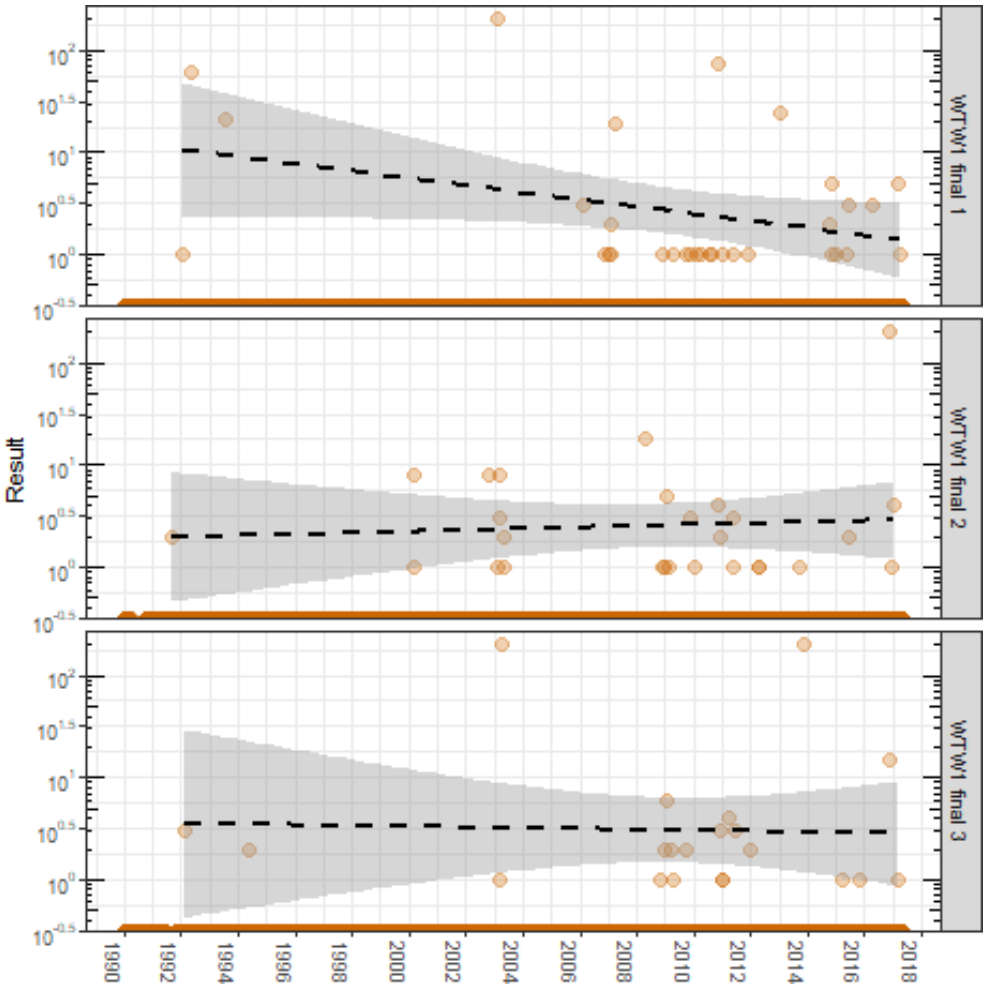

Figure 8-1: Coliform trend for three final waters including censored values (coliforms = 0)

Figure 8-1 reveals that there are a high number of censored values where the coliform count = 0. The trend line on a log scale has been drawn with the coliforms = 0 data being removed (because log(0) is infinite); to demonstrate this, the trend line is redrawn in Figure 8-2 with the data (coliforms = 0) removed. The three trend lines are identical for Figure 8-1 and Figure 8-2.

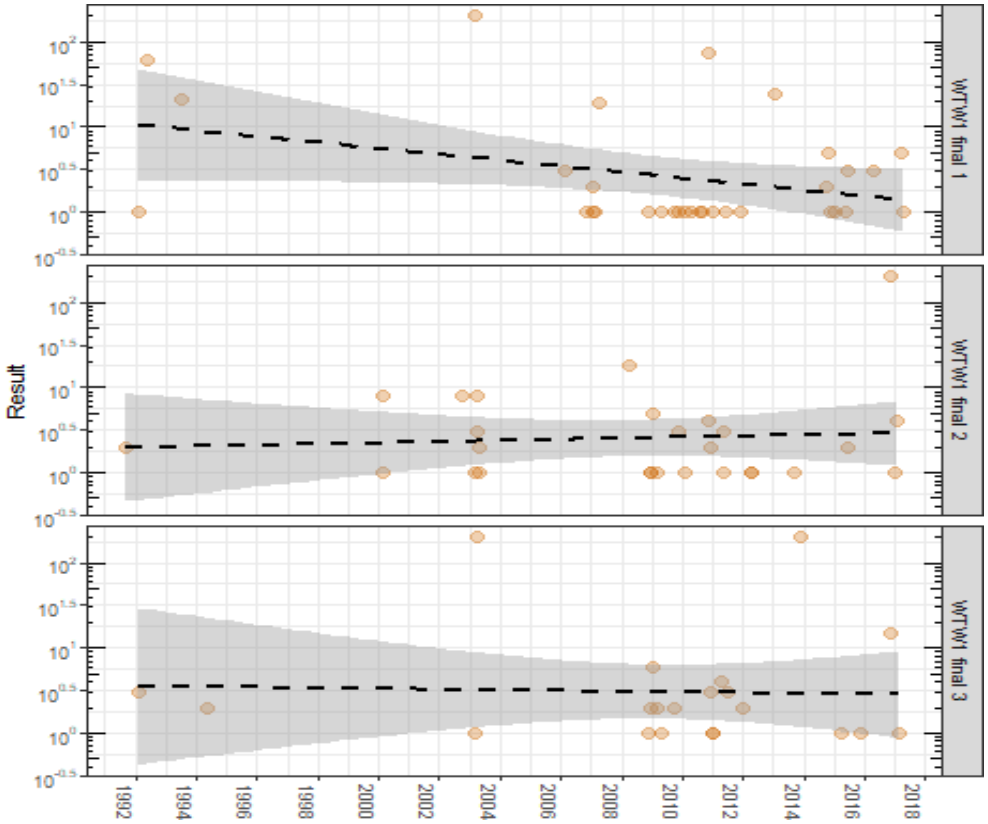

Figure 8-2: Coliform results for three final waters where coliforms = 0 are removed.

The code for this is shown below:

```
filtered_coliforms0<-filtered_coliforms%>%
        filter(Result!=0)#keepallnotequaltozero
#find start date of data
S_date <- min(filtered_coliforms0$Sample_Date)
#find start date of data
E_date <- max(filtered_coliforms0$Sample_Date)
#Graph uses the variable name SampleDate
#Must be date only
filtered_coliforms0$SampleDate <-lubridate::as_date(filtered_coliforms0$Sample_Date)
g1 <- Magic_TimeLine(filtered_coliforms0,
        S_date,
        E_date,
        TimeBreak = "2 year",
        ParaColour ="darkorange3",
        Gtitle="Coliformsforthreefinals",
        Gcaption = "Note Log10 scale",
        Units="MPN/100mL",
        LinearScale=FALSE,
        Regression =TRUE,
        Alpha = 0.3,
        Size = 3)
gridExtra::grid.arrange(g1)
```

To plot a trend of all three *final* results on one plot with the `Magic_TimeLine` function the description is changed to `final` as shown in the R-code chunk below, the plot is shown in Figure 8-3. Table 8-2 shows the coliform summary statistics including the censored values equal to zero.

```
#Copy data.frame to filtered_final
filtered_final <- filtered_coliforms0
#description is changed to `final`
filtered_final$Description <- "final"
#find start date of data
S_date <- min(filtered_final$Sample_Date)
#find start date of data
E_date<-max(filtered_final$Sample_Date)
#Graph uses the variable name SampleDate
 #Must be date only
filtered_final$SampleDate<-lubridate::as_date(filtered_final$Sample_Date)
g1 <- Magic_TimeLine(filtered_final,
S_date, E_date,
TimeBreak = "2 year", ParaColour="darkorange3", Gtitle="ColiformsFinal", Gcaption="NoteLog10scale",
Units = "MPN/100mL", LinearScale = FALSE, Regression =TRUE,
Alpha = 0.3,
Size = 3)
gridExtra::grid.arrange(g1)
```

Table 8-2: Coliform results including the censored values equal to zero

Description	min	Q1	median	Q3	max	mean	sd	n	missing
WTW1 final 1	0	0	0	0	201	0.0449964	2.278424	9823	0
WTW1 final 2	0	0	0	0	201	0.0294726	2.059730	9670	0
WTW1 final 3	0	0	0	0	201	0.0475393	2.914505	9550	0

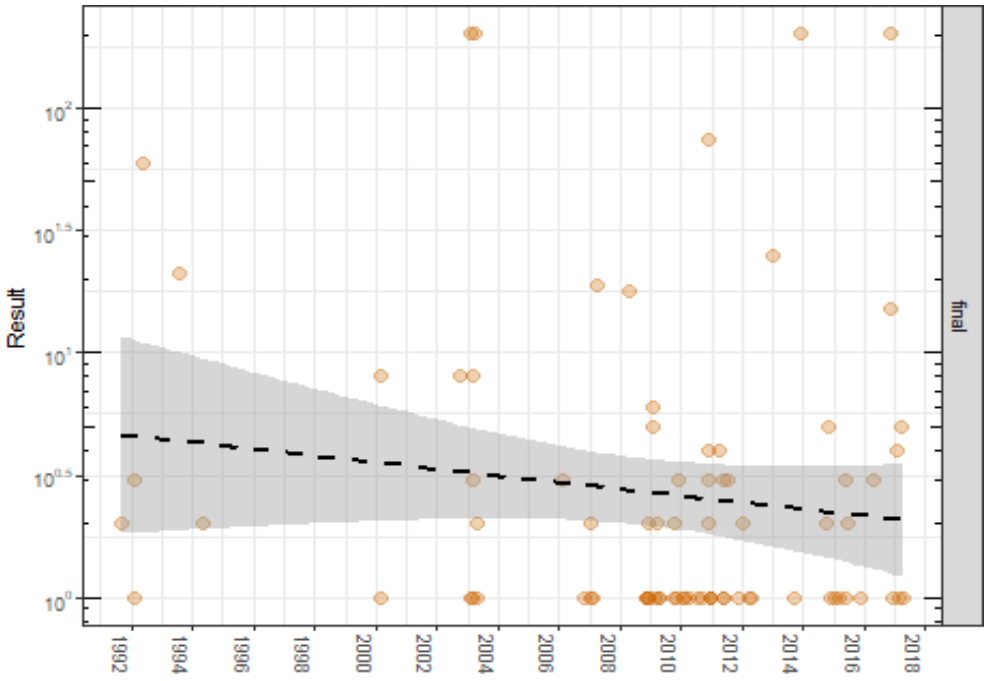

Figure 8-3: Coliforms of all Final Waters

```
#The summary statistics are saved into a variable sumstats
sumstats <- favstats(Result~Description, data = filtered_coliforms)
#sumstats is made into a the table (using the code below)
knitr::kable(sumstats,
caption='Coliformresultsincludingthecensoredvaluesequaltozero', booktabs = TRUE,
align = c("l", rep("r",9)))
```

Table 8-3 shows the coliform summary statistics with censored values equal to zero removed. The code is provided below:

```
#The summary statistics are saved into a variable sumstats
sumstats <- favstats(Result~Description, data = filtered_coliforms0)
#sumstats is made into a the table (using the code below)
knitr::kable(sumstats,
caption='Coliformresultswithcensoredvaluesequaltozeroremoved', booktabs = TRUE,
align = c("l", rep("r",9)))
```

Table 8-3: Coliform results with censored values equal to zero removed

Description	min	Q1	median	Q3	max	mean	sd	n	missing
WTW1 final 1	1	1	1	3.5	201	13.81250	38.05890	32	0
WTW1 final 2	1	1	2	4.0	201	10.55556	38.24047	27	0
WTW1 final 3	1	1	2	3.0	201	21.61905	59.71556	21	0

8.4 QUESTION 2 – IS THERE A STATISTICALLY SIGNIFICANT DIFFERENCE IN COLIFORM RESULTS AT THE THREE SAMPLE SITES AT A MAGIC WTW?

A statistic that would test for this difference is the Kruskal-Wallis rank sum test; a simple non-parametric test to compare the medians of three or more samples (see https://en.wikipedia.org/wiki/Kruskal%E2%80%93Wallis_one-way_analysis_of_variance). For the coliform results at the three sample sites final 1, final 2, final 3:

- **Null hypothesis:** the three sample sites are from distributions with the same median.
- **Alternative hypothesis:** three sample sites are from distributions with a different median.

The data.frame `filtered_coliforms0` has the non-censored results in the `Result` column and the three final water samples in the `Description`; the R code below runs the Kruskal-Wallis test.

Always Visualise the Data!

```
ggplot(filtered_coliforms0,aes(x=factor(Description),y=Result,fill=Description))+
        geom_boxplot() +
        scale_y_log10() +
        labs(title ="Coliforms", y = "MPN/100mL")+
        theme_bw() +
        scale_fill_tableau("Color Blind") +
        theme (legend.position="none") +
        theme (axis.title.x=element_blank())
```

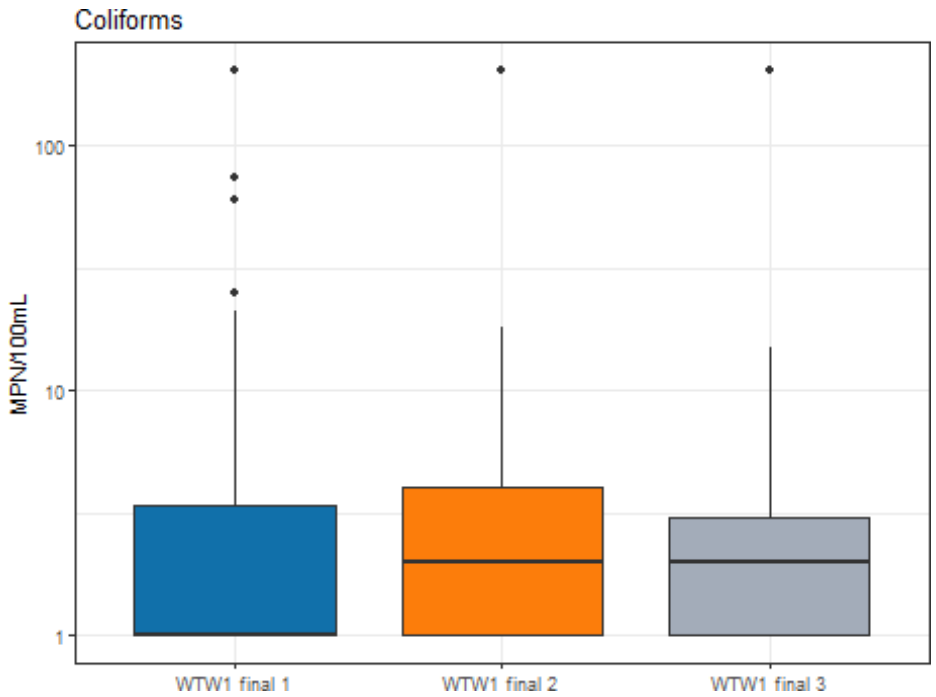

Figure 8-4: Boxplots of coliforms for three sample sites

```
# Description needs to be a factor for the kruskal.test()
filtered_coliforms0$Description <-factor(filtered_coliforms0$Description)
# The p value from the kruskal.test is saved to a variable P.stat to be used
# in the text below
P.stat<-kruskal.test(Result~Description,data=filtered_coliforms0)$p.value
#Used in the text below
K_W.hypo <- "three sample sites"
```

The resulting p-value of 0.6599 is (P >0.05) tells you to not reject the null hypothesis.
There is no difference between the three sample sites; they come from distributions with the same median.

8.5 COLIFORM TREND ANALYSIS

To calculate whether the coliform trend is significant, the results can be averaged for a period (e.g. Year) and then the Mann Kendall Trend Test can be applied. The Mann Kendall Trend Test (sometimes called the M-K test) is used to analyse data collected over time for consistently increasing or decreasing trends (monotonic trends) in Y values. It is a non-parametric test, which means it works for all distributions (i.e. data doesn't have to meet the assumption of normality).

The test can be used to find trends for as few as four samples. However, with only a few data points, the test has a high probability of not finding a trend when one would be present if more points were provided. The more data points you have the more likely the test is going to find a true trend (as opposed to one found by chance). The minimum number of recommended measurements is normally at least 8 to 10.

- The null hypothesis for this test is that there is no monotonic trend in the series.
- The alternate hypothesis is that a trend exists. This trend can be positive, negative, or non-null.

The R code chunk below takes the data.frame `filtered_coliforms0` with the Coliforms equal zero removed; creates a Year variable (from the SampleDate) and calculates the average (mean) `ave_coli` Coliform counts for each year, shown in Table 8-4.

```
#Make a year `factor` variable
filtered_coliforms0$Year <-factor(lubridate::year(filtered_coliforms0$SampleDate))
Year_coliforms<-filtered_coliforms0%>%
        group_by(Year)%>%
        summarise(ave_coli = mean(Result))
knitr::kable(Year_coliforms,
caption='ColiformAnnualMean', booktabs = TRUE,
align = c("l", "r"))
```

Table 8-4: Coliform annual mean

Year	ave_coli
1991	2.000000
1992	21.333333
1993	21.000000
1994	2.000000
2000	4.500000
2002	8.000000
2003	52.250000
2006	2.000000
2007	5.750000
2008	4.000000
2009	2.300000
2010	7.583333
2011	2.142857
2012	1.000000
2013	75.666667
2014	2.666667
2015	1.500000
2016	55.000000
2017	2.750000

The R code below takes Table 8-4 and makes a `ggplot` which is saved to `g1` (so it can be plotted later in Figure 8-5). The `ggplot` function was used to create the `Magic_TimeLine`; `ggplot` is a flexible and powerful visualisation package; for more information see http://r4ds.had.co.nz/data-visualisation.html.

```
#Make Year Numeric so trend will plot
Year_coliforms$Year <- as.numeric(Year_coliforms$Year)
#ggplot is used in `Magic_TimeLine`
g1<-ggplot(Year_coliforms,aes(x=Year,y=ave_coli))+
         geom_point(size = 4) +
        geom_smooth(method="lm") +
        labs(title="Coliforms Annual Mean",
y = "MPN/100mL") +
        theme_bw() +
theme (legend.position = "none",
        axis.text.x=element_text(size=10,angle=0,face="bold"),
        axis.text.y=element_text(size=10,face="bold"),
        strip.text=element_text(size=12,face="bold"))+
        theme (axis.title.x =element_blank())
```

The R code below creates a Mann-Kendall `M_K` function that returns text stating if the trend is significant or not. The text is plotted next to the graph `g1` in Figure 8.4.

```
#This is a Mann-Kendall function that returns text
#The returned text states if the trend is significant or not
M_K <- function(a_vector) {
        res <- MannKendall(a_vector)
        answer<-stringr::str_c("tau=",      paste(round(res[[1]],digits=2)),"\n",
        "p-value=",         paste(round(res[[2]], digits =6)))
        if (res[[2]] < 0.05) {
text<-stringr::str_c("MannKendallTrendTest",answer,"Trendissignificant(p<0.05)",
txt_col <- "darkred"
} else {
text<-stringr::str_c("MannKendallTrendTest",answer,"Trendnotsignificant(p>0.05)",
sep txt_col <- "steelblue"
}
text_grob(text, face = "italic", color = txt_col)
}
#Pull a vector for the Mann-Kendall function
MK_vector<-Year_coliforms%>% pull(ave_coli)
#Call Mann-Kendall function
t_mk1 <- M_K(MK_vector)
 grid.arrange(g1,t_mk1,ncol=2,widths=c(2,1))
```

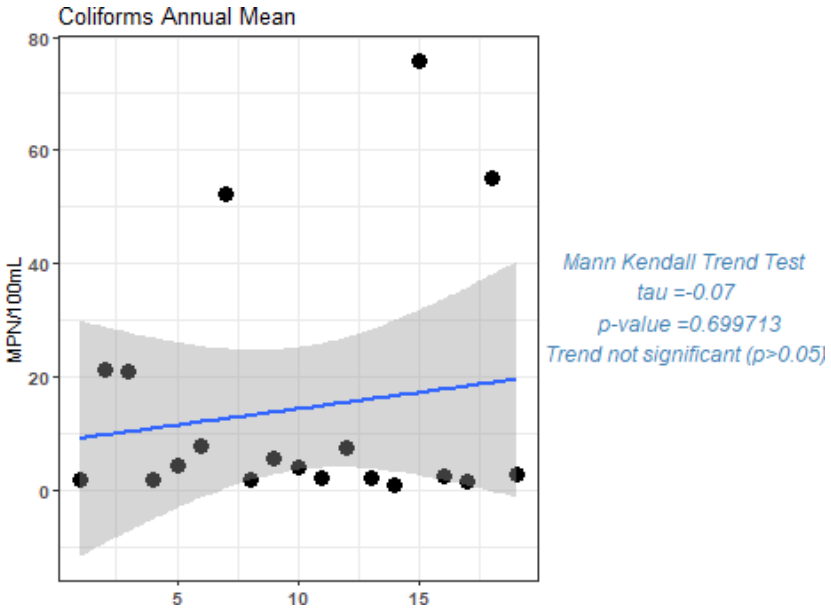

Figure 8-5: Coliform annual mean and Mann Kendall Trend test

8.6 DIFFERENCE IN COLIFORM RESULTS EACH YEAR

With reference to Question 1 (see Section 8.3), we can also use the Kruskal-Wallis rank sum test to see if there is a statistically significant difference in coliform results for each year.

- **Null hypothesis:** the coliforms each year are from distributions with the same median.
- **Alternative hypothesis:** coliforms each year are from distributions with a different median.

The data.frame `filtered_coliforms0` has the non-censored results in the `Result` column and the years in the `Years` column (already a `factor`); the R code below runs the Kruskal-Wallis test.
Always Visualise the Data! See Figure 8-6.

```
ggplot(filtered_coliforms0,aes(x=factor(Year),y=Result,fill=Year))+
geom_boxplot() +
scale_y_log10() +
labs(title ="Coliforms", y = "MPN/100mL")+
theme_bw() +
scale_fill_tableau("Tableau20")+
theme (legend.position="none") +
theme(axis.title.x=element_blank())
```

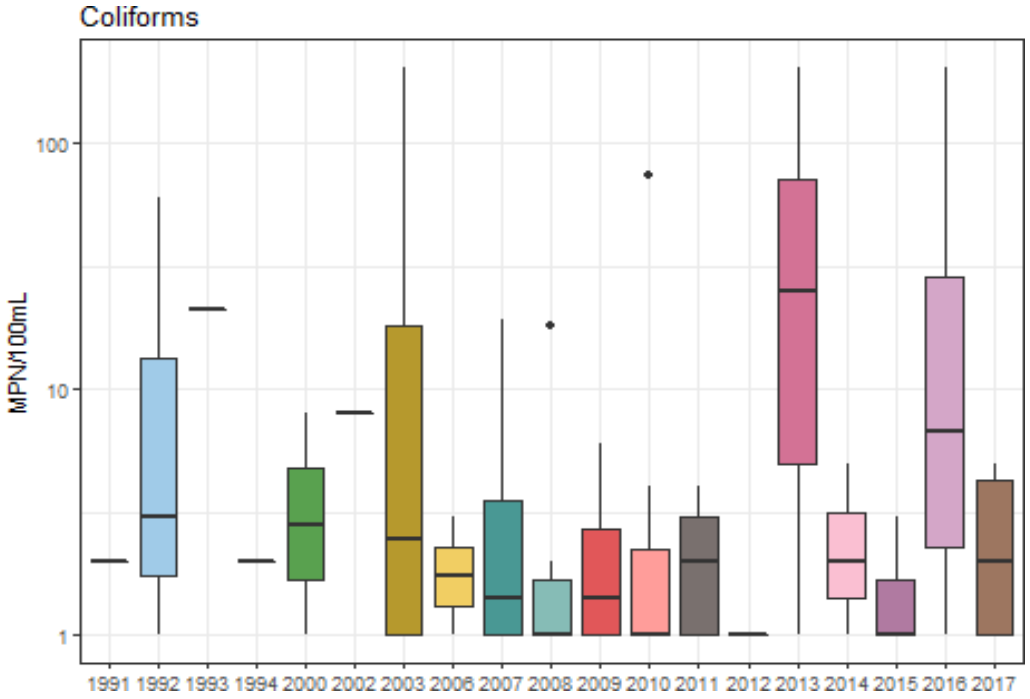

Figure 8-6: Boxplot of coliforms for each year

```
# The p value from the kruskal.test is saved to a variable P.stat to be used # in the text below
P.stat <- kruskal.test(Result~Year, data=filtered_coliforms0)$p.value
#Used in the text below
K_W.hypo <- "coliforms eachyear"
```

The resulting p-value of 0.666 is (P >0.05) tells you to not reject the null hypothesis.
There is no difference between the coliforms each year; they come from distributions with the same median.

8.7 DIFFERENCE IN COLIFORM RESULTS FROM 2010

With reference to Question 1 we can also use the Kruskal-Wallis rank sum test to see if there is a statistically significant difference in coliform results from 2010:

- **Null hypothesis:** the coliforms each year are from distributions with the same median.
- **Alternative hypothesis:** coliforms each year are from distributions with a different median.

First filter the results from 2010, see R-code below.

```
filtered_post2010 <- filtered_coliforms0 %>%
filter(Sample_Date > lubridate::ymd(20100101))
```

The data.frame filtered_coliforms0 has the non-censored results in the Result column and the years in the Years column (already a factor); the R code below runs the Kruskal-Wallis test.

Always Visualise the Data!

See Figure 8-7: Boxplot of coliforms for each year

```
# The p value from the kruskal.test is saved to a variable P.stat to be used
# in the text below
P.stat <- kruskal.test(Result~Year, data=filtered_post2010)$p.value
#Used in the text below
K_W.hypo <- "coliforms each year"
```

```
ggplot(filtered_post2010,aes(x=factor(Year),y=Result,fill=Year))+
        geom_boxplot() +
        scale_y_log10() +
        labs(title ="Coliforms", y = "MPN/100mL")+
        theme_bw() +
        scale_fill_tableau("Color Blind") +
        theme (legend.position="none") +
        theme (axis.title.x=element_blank())
```

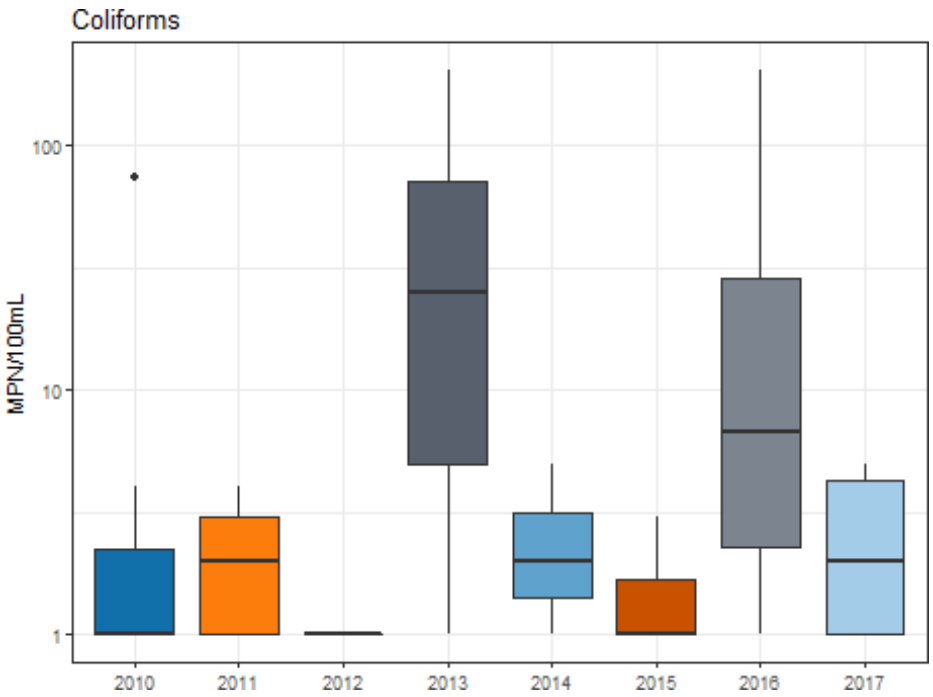

Figure 8-7: Boxplot of coliforms for each year

```
# The p value from the kruskal.test is saved to a variable P.stat to be used
# in the text below
P.stat <- kruskal.test(Result~Year, data=filtered_post2010)$p.value
#Used in the text below
K_W.hypo <- "coliforms each year"
```

The resulting p-value of 0.332 is (P >0.05) tells you to not reject the null hypothesis. There is no difference between the coliforms each year; they come from distributions with the same median.

8.8 ALTERNATIVE APPROACH – COLIFORMS KERNEL DENSITY

Analysis of Coliforms using percentage of positive counts (the number of cfu's has been ignored) Analysis focuses on data post 2002. The Kernel density plots show that most Coliform samples are returned negative.

```
coliforms_Pre2002 <- magic_WW %>%
        filter(Determinand == "Coliforms",
Sample_Date < lubridate::ymd(20030101))
ggplot(coliforms_Pre2002,aes(Result))+
        geom_density(fill = "black") +
        xlim(0, 300)
```

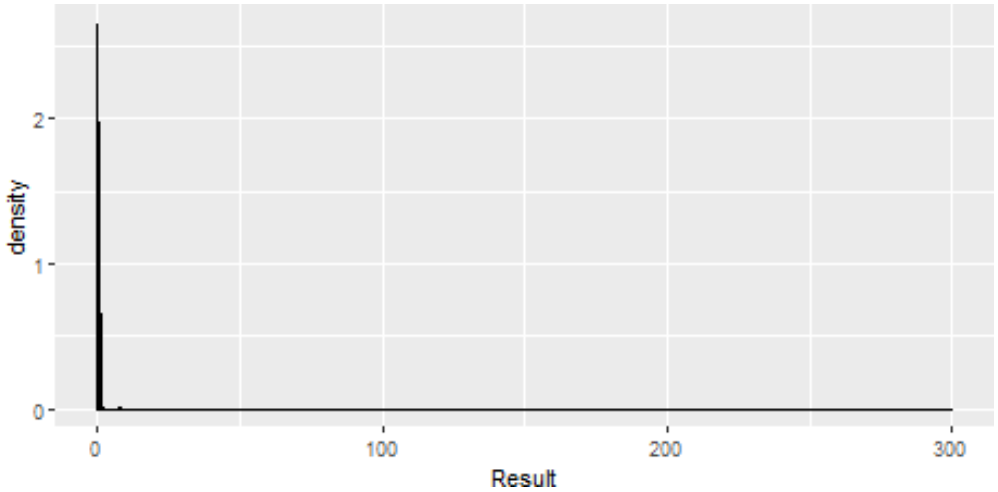

Figure 8-8: Coliform Kernel Density Plot – pre 2002

```
coliforms_post2002<-magic_WW%>%
        filter(Determinand == "Coliforms",
Sample_Date >= lubridate::ymd(20030101))
ggplot(coliforms_post2002,aes(Result))+
        geom_density(fill = "black") +
        xlim(0, 300)
```

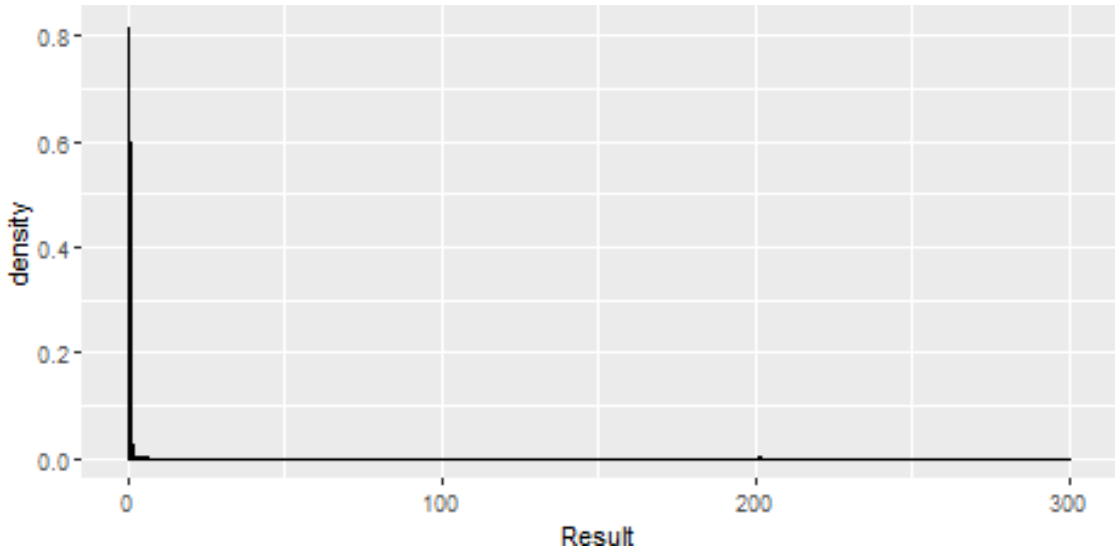

Figure 8-9: Coliform Kernel Density Plot – Post 2002

8.9 POSITIVE COLIFORMS ANNUALLY POST 2002 (PERCENTAGE POSITIVES OF COLIFORM SAMPLES)

Count the positive Coliforms recorded annually post 2002 (calculated as a percentage (%) of Coliforms samples taken).

Table 8-5: Positive Coliforms recorded annually post 2002

Year	Total Samples (No)	Positive Samples No.	Positive (%)
2003	1090	8	0.7339450
2004	1084	0	0.0000000
2005	1085	0	0.0000000
2006	1083	2	0.1846722
2007	1086	4	0.3683241
2008	1084	6	0.5535055
2009	1088	10	0.9191176
2010	1088	12	1.1029412
2011	1090	7	0.6422018
2012	1094	2	0.1828154
2013	1095	3	0.2739726
2014	1093	3	0.2744739
2015	1092	6	0.5494505
2016	1096	4	0.3649635
2017	360	4	1.1111111

Plot a graph of the percentage failures and see if there is a trend using Mann-Kendal. To do this we use the following code:

```
grid.arrange(g1,t_mk1,ncol=2,widths=c(2,1))
```

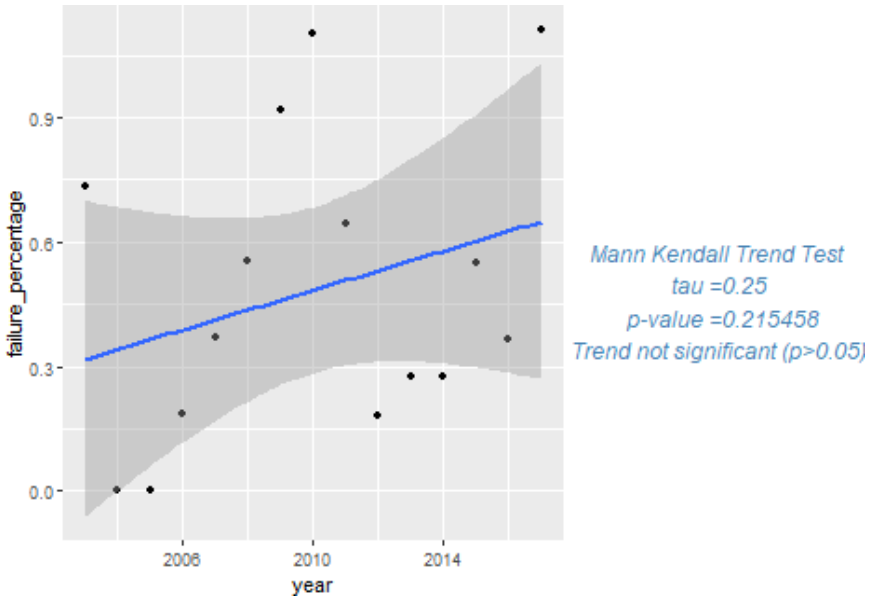

Figure 8-10: Coliform Positive results and trends

If the three final waters are reviewed separately and the percentage positive coliforms are determined for each one separately the following observations and code is provided.

Table 8-6: Percentage of positive coliforms for each final water

Year	WTW1 final 1	WTW1 final 2	WTW1 final 3
2003	0.2762431	1.3661202	0.5524862
2004	0.0000000	0.0000000	0.0000000
2005	0.0000000	0.0000000	0.0000000
2006	0.5586592	0.0000000	0.0000000
2007	1.0989011	0.0000000	0.0000000
2008	0.2762431	0.8287293	0.5555556
2009	0.8241758	0.8241758	1.1111111
2010	1.6483516	0.8287293	0.8287293
2011	0.5524862	0.5509642	0.8219178
2012	0.0000000	0.5479452	0.0000000
2013	0.2739726	0.2739726	0.2739726
2014	0.8264463	0.0000000	0.0000000
2015	0.8219178	0.2739726	0.5524862
2016	0.2739726	0.5479452	0.2732240
2017	1.6666667	0.8333333	0.8333333

```
ggplot(Coliform_data, aes(x=factor(description), y=failure_percentage, fill=description))  +
        geom_boxplot() +
        labs(title = "Postive Coliforms (%)",
        y = "(%)")+
        theme_bw()+
        scale_fill_tableau("Color Blind") +
        theme (legend.position="none") +
        theme(axis.title.x=element_blank())
```

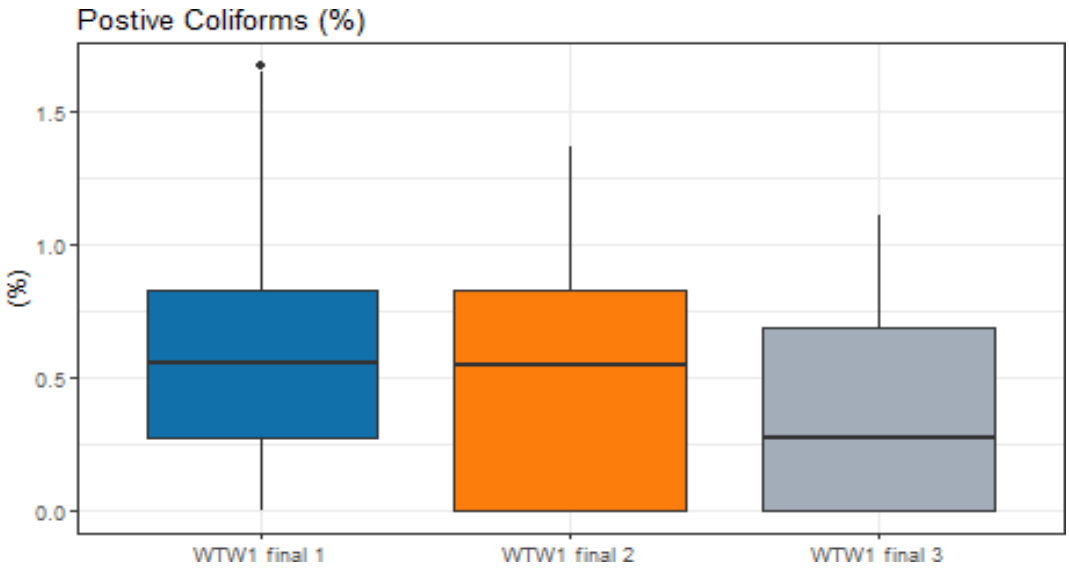

Figure 8-11: Box plot of the three final water positive percentages

```
ggplot(Coliform_data,aes(x=year,y=failure_percentage,colour=description))+
        geom_point()+
        geom_smooth(method = lm) +
        scale_colour_tableau("ColorBlind")+
        facet_wrap(~description)
```

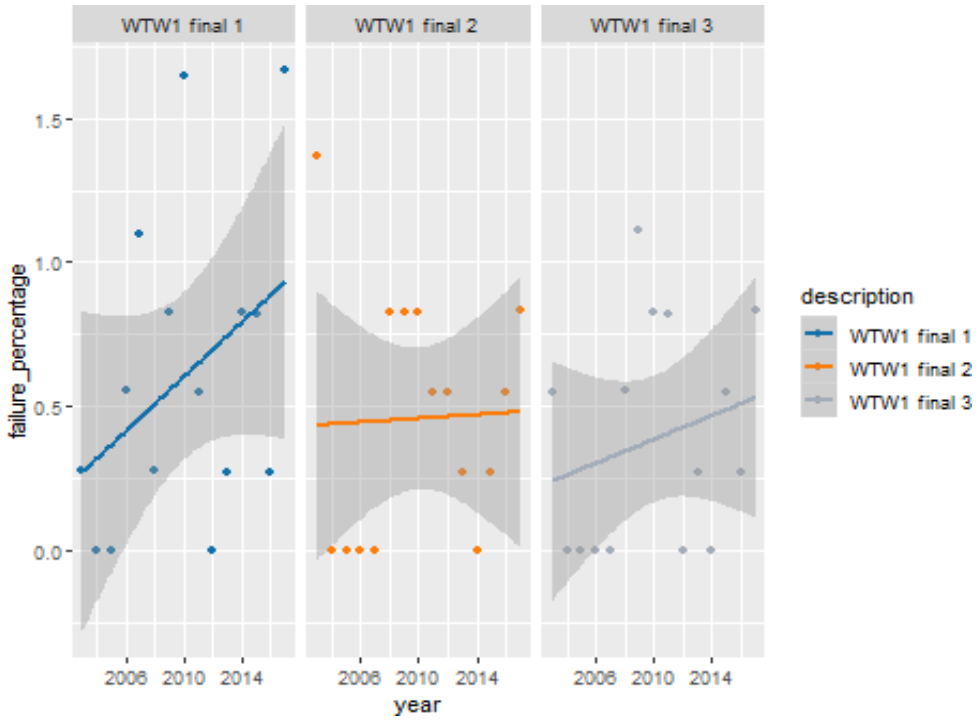

Figure 8-12: Trends in the three final water samples for coliforms

If we examine one location at a time, we can perform the Mann Kendall trend test per location and the code for this is provided below.

```
one_location <- Coliform_data %>% filter(description=="WTW1final1")
g1<-ggplot(one_location,aes(x=year,y=failure_percentage,colour=description))+
        geom_point(colour = "black")+
        geom_smooth(method = lm)
#Pull a vector for the Mann-Kendall function
MK_vector<-one_location%>% pull(failure_percentage)
#Call Mann-Kendall function
t_mk1 <- M_K(MK_vector)
grid.arrange(g1, t_mk1, ncol = 2,widths=c(2,1))
```

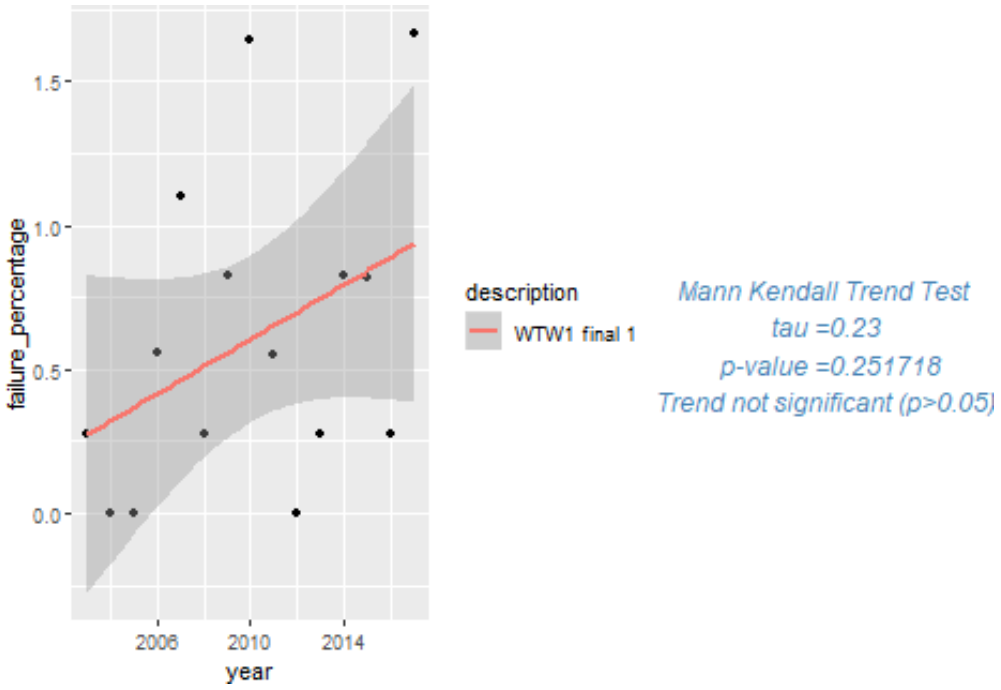

Figure 8-13: Final 1 Mann Kendall trend test for WTW Final 1

```
one_location <- Coliform_data %>% filter(description=="WTW1final2")
g1<-ggplot(one_location,aes(x=year,y=failure_percentage,colour=description))+
        geom_point(colour = "black")+
        geom_smooth(method = lm)
#Pull a vector for the Mann-Kendall function
MK_vector <- one_location %>%
pull(failure_percentage)
#Call Mann-Kendall function
t_mk1 <- M_K(MK_vector)
grid.arrange(g1, t_mk1, ncol = 2,widths=c(2,1))
```

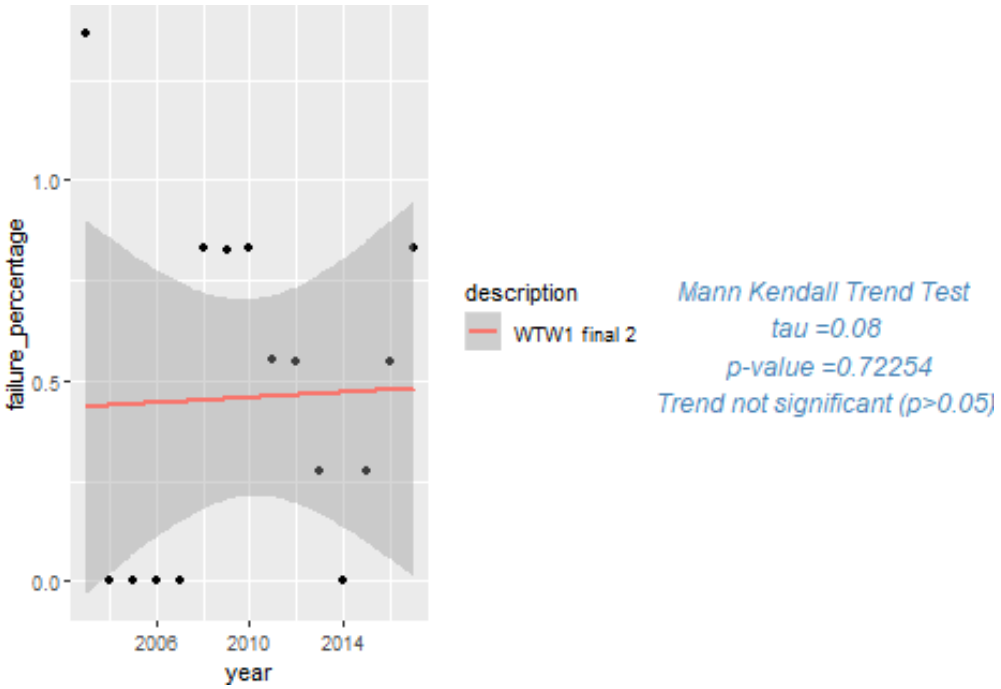

Figure 8-14: Final 2 Mann Kendall trend test for Final 2

```
one_location <- Coliform_data %>% filter(description=="WTW1 final 3")
g1<-ggplot(one_location, aes(x=year, y=failure_percentage, colour=description))+
        geom_point(colour = "black")+
        geom_smooth(method = lm)
#Pull a vector for the Mann-Kendall function
MK_vector <- one_location %>%
        pull(failure_percentage)
#Call Mann-Kendall function
t_mk1 <- M_K(MK_vector)
grid.arrange(g1, t_mk1, ncol = 2, widths=c(2,1))
```

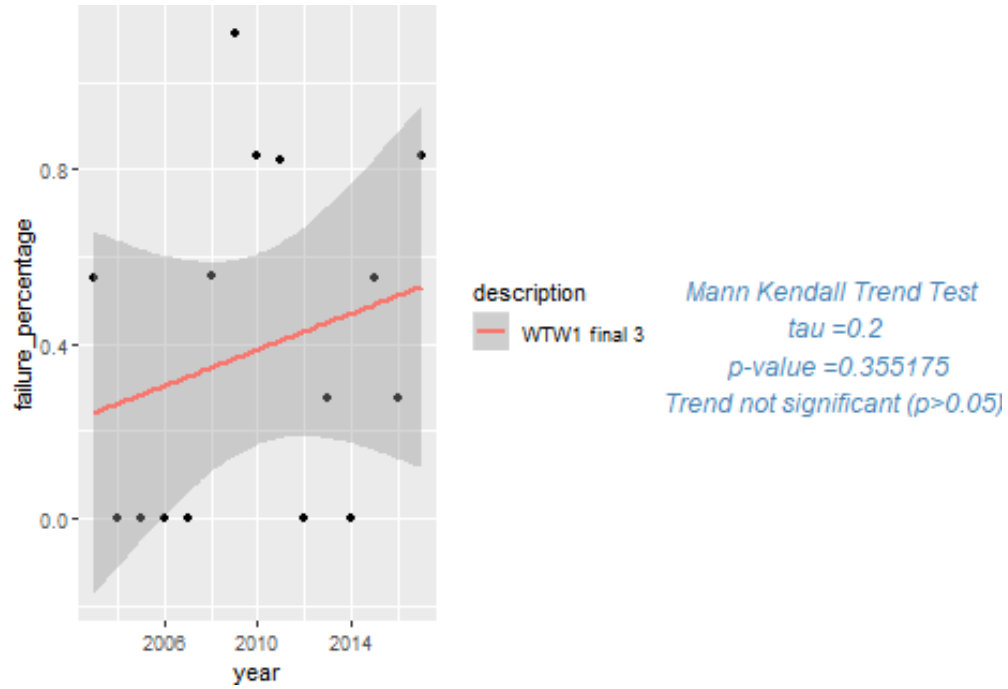

Figure 8-15: Final 3 Mann Kendall trend test for Final 3

Table 8-7: Positive Coliforms recorded

Month	Total Samples (No)	Positive Samples No	Positive (%)
Jan	1386	8	0.5772006
Feb	1269	8	0.6304177
Mar	1387	7	0.5046864
Apr	1344	11	0.8184524
May	1299	5	0.3849115
Jun	1248	2	0.1602564
Jul	1291	1	0.0774593
Aug	1300	1	0.0769231
Sep	1246	3	0.2407705
Oct	1294	3	0.2318393
Nov	1253	14	1.1173184
Dec	1291	8	0.6196747

```
ggplot(Coliform_data,aes(x=factor(month),y=failure_percentage))+
        geom_bar(stat = "identity", fill = "gold")
```

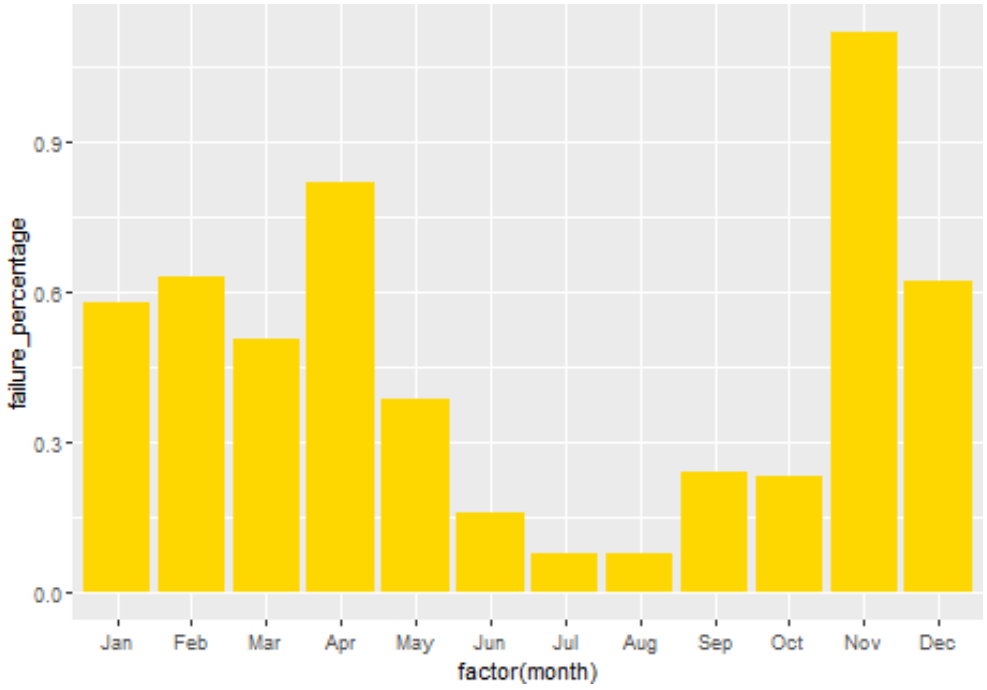

Figure 8-16: Monthly Positives for Coliform

CHAPTER 9: CONCLUSIONS & RECOMMENDATIONS

The ability to look beyond our current data manipulation and visualisation techniques using the faithful Excel spreadsheet, does not come easily and indeed may be quite a paradigm shift for many participants in the water sector. When learning R, it should be borne in mind that this requires discipline and tenacity, perseverance and the ability to search for answers when one hits a brick wall. However, this language provides real progress and a different way of processing any sized dataset until the visualisation of the dataset meets with the data and water scientist's expectations. R is not easy to learn but the rewards are great.

In the area of data management within the laboratory, during the course of the data tidying process the following issues were identified during the consultative process:

- There is a need for standardisation of field identifiers between study datasets.
- Result reporting policy influence on the data requires additional investigation prior to applying any further interpretation to the results relative to making deductions about water quality.
- The presence of "rogue" data points or outliers causes compression of graphical apices and limits the visual assessment of time series data. A protocol was agreed upon for addressing this issue for this project which limits the risk of losing relevant information about isolated genuine events. It is also relevant to anyone who attempts similar projects.
- Should some sample locations be absence from the first assessment, the lack of data should be investigated to ensure that the information is available in the database and is presented in future iterations.
- Consistency in record naming of different analysis to allow for easier data manipulation.
- It is also strongly recommended that a team is constituted to investigate the value of and requirement for a Laboratory Data Management Policy to guide decision-making regarding laboratory data. This may include, amongst others, issues related to reporting limit changes, documentation of analytical method changes, issues related to decimal point reporting, data validation, etc. It is recommended that this team is led by the Quality Team from Laboratory Services, but staff from other water company departments will also need to contribute.
- In future, geo spatial mapping of sample points and water quality can be investigated.
- Interactive consultation with WSPs in South Africa to generate a water quality data science community.

BIBLIOGRAPHY

1. *Prevos, P. (2017).* Lifting the big data veil. *Online Journal of the Australian Water Association, Volume 2. R Core Team (2018).*
2. R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.*
3. *Sefick Jr., S.A. (2016).* Stream Metabolism: Calculate Single Station Metabolism from Diurnal Oxygen Curves*. R package version 1.1.2.*
4. *Wickham, H., Chang, W., Henry, L., Pedersen, T.L., Takahashi, K., Wilke, C., and Woo, K. (2018).* ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. R package version 3.1.0.*
5. *Xie, Y. (2015).* Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.*
6. *Xie, Y. (2018*). Bookdown: Authoring Books and Technical Documents with R Markdown. *R package version 0.7.*
7. *The online book* R for Data Science, *by Garrett Grolemund & Hadley Wickham, https://r4ds.had.co.nz/*
8. R Markdown: The Definitive Guide *by Yihui Xie, J. J. Allaire, Garrett Grolemund accessed athttps://bookdown.org/yihui/rmarkdown/*
9. *The online book* Geocomputation with R *by Robin Lovelace, Jakub Nowosad and Jannes Muenchow CRC Press accessed at https://geocompr.robinlovelace.net/*
10. *The online book* Text Mining with R *by Julia Silge and David Robinson – https://www.tidytextmining.com/*
11. *The online book* Efficient R programming *by Colin Gillespie and Robin Lovelace O'Reilly Publishing 2016 accessed at https://csgillespie.github.io/efficientR/*
12. Handling Strings with R *by Gaston Sanchez https://www.gastonsanchez.com/r4strings/*
13. An Introduction to Statistical and Data Sciences *via R by Chester Ismay and Albert Y. Kim https://moderndive.com/*
14. Exploratory Data Analysis with R *by Roger D. Peng https://bookdown.org/rdpeng/exdata/*
15. R Programming for Data Science *by Roger D. Peng https://www.cs.upc.edu/~robert/teaching/estadistica/rprogramming.pdf*
16. *https://datajobs.com/what-is-data-science Accessed 17 December 2018*
17. *Wickham, Hadley, 2014, Tidy Data, Journal of Statistical Software, Volume 59, Issue 10. https://www.jstatsoft.org*