Material for R for Water Professionals

Peter Prevos

Material for R for Water Professionals

Peter Prevos

This book is for sale at http://leanpub.com/courses/leanpub/R4H2O

This version was published on 2019-06-23

Contents

R for Water Professionals **1**
Introduction to Water Utility Data Science 2
Case Study 1: Introduction to the R Language — Water Quality Regulations 2
Case Study 2: Processing Data — Understanding Customer Perception 3
Case Study 3: Creating Data Products — Analysing Water Consumption 4
Capstone project 5
Prerequisites 5
Downloading the workshop material 6

Data Science for Water Utilities **8**
What is data science? 8
The Elements of Data Science 10
The Water-Data Value Chain: The Digital Water Utility 13
Data Science Tools 14
Good Data Science 16
Best-Practice Data Science with R 31

Introduction to the R Language **33**
Using R and RStudio 33
Basics of R 34
RStudio scripts and projects 36

Case Study: Water Quality Regulations **38**
Turbidity 38
Problem Statement 39
Methodology 39
Analysing the case study 40

Quiz: Water Quality Regulations **49**

Welcome to the Tidyverse **50**
Packages for water management 50
The Tidyverse 50

Case Study 2: Understanding Customer Perception **52**

Consumer Involvement 52
Problem Statement 53
Methodology 53
Analysing the Case Study 55
Load the data 55
Visualise the data 58
Assignment 60

Answers to the questions **61**
Introduction to the R Language 61
Case Study: Water Quality Regulations 63
Quiz 1: Water Quality Regulations 65

R for Water Professionals

Managing reliable water services requires not only a sufficient volume of water but also significant amounts of data. Water professionals continuously measure the flow and quality of water and asses how customers perceive their service. Water utilities are awash, or even flooded with data. Data professionals use data pipelines and data lakes and make data flow from one place to another.

Data and water are, as such, natural partners. Professionals in the water industry rarely directly interact with water or customers, but they are constantly analysing data that describes these realities. The purpose of collecting and analysing this data is to maintain or improve the level of service to customers and to minimise the impact on the natural environment.

Most professionals use spreadsheets to work with data. While these tools are handy, they are not ideal when working with large and complex sets of data. Specialists in data analysis prefer to write code in one of the many available computing languages.

This course introduces water utility professionals to the R language[1] for statistical computing. This language is one of the most popular tools among data scientists to create value from data.

This workshop is not an exhaustive introduction into data science programming but a teaser to inspire water professionals to ditch their spreadsheets[2] and start writing code to analyse data. The

[1] https://en.wikipedia.org/wiki/R_(programming_language)
[2] https://lucidmanager.org/spreadsheets-for-data-science/

best way to learn to solve problems with code is to solve these problems and learn as you need new skills.

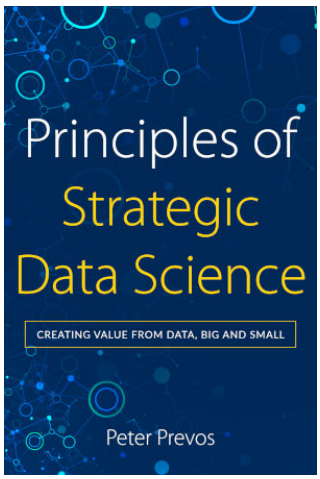
This course only discusses the basics of using the R language with a limited scope. This course does not include any discussion of more advanced techniques such as machine learning. All data used in this course is tabular, text analysis and other unstructured data are excluded.

The course consists of four sessions and a capstone project. The first session introduces the principles of data science within the context of managing a water utility. Following a case-study approach, this course includes three realistic case studies about water management. These sessions start with a problem statement and introduce participants to the relevant aspects of the R language. Participants have to load, transform, explore and analyse the data to solve the problem. The course closes with a capstone project where participants develop a water quality dashboard.

Introduction to Water Utility Data Science

The first session defines data science as an evolution of traditional analysis. The greater availability of data, enhanced computer capacity and tools to analyse this information have revolutionised the industry. The second part of this session introduces a framework for best practice in data science.

The content of this session is based on the ebook *Principles of Strategic Data Science*. Participants of this course can download a discounted copy[3] of this book.



Prevos, P. (2019) Principles of Strategic Data Science.

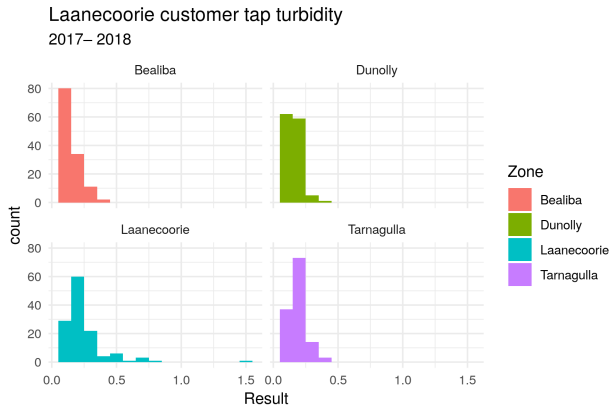
Case Study 1: Introduction to the R Language — Water Quality Regulations

This first session introduces to the principles of data science and presents a framework for and best practice. This session also introduces the basics of the R language to undertake simple statistical

[3] http://leanpub.com/strategic_data_science/c/r4h2o

analysis.

This session uses laboratory testing data from a drinking water network. Participants use this data to assess descriptive statistics and compliance with regulations.

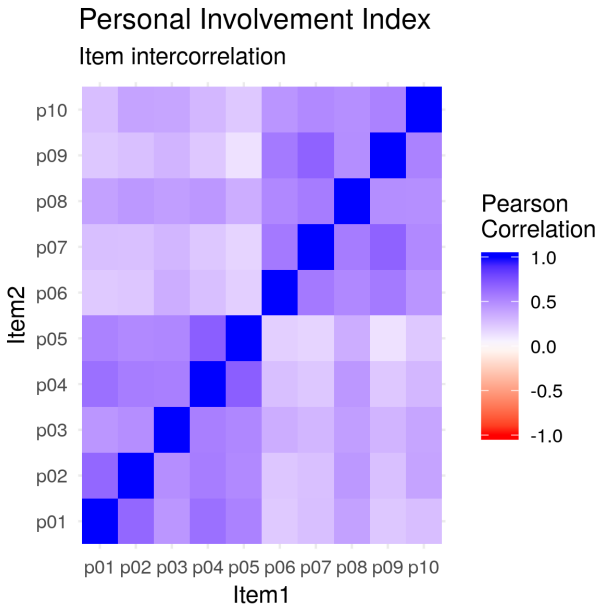


Distribution of turbidity results (Case Study 1).

Case Study 2: Processing Data — Understanding Customer Perception

The Tidyverse is an extension of the R language that provides additional functionality to simplify analysing manipulating and data. In the second session, participants learn how to clean and explore data.

The case study for this data are the results of a survey among American consumers about their perception of water services. Participants use this data set to clean, transform and visualise the data.

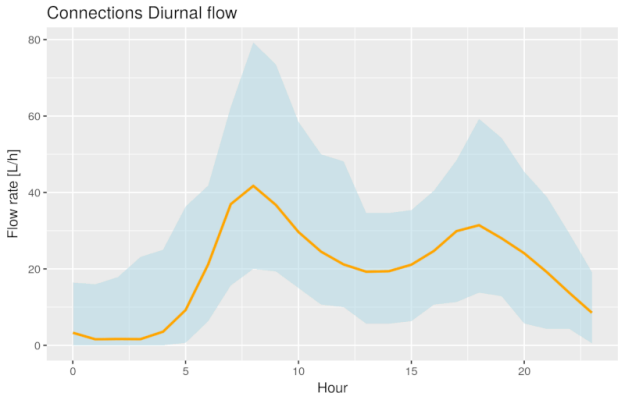


Consumer involvement with tap water (Case Study 2).

Case Study 3: Creating Data Products — Analysing Water Consumption

In the last session, participants learn about the data science workflow and creating data products for end-users. Participants analyse an extensive data set to find anomalies in water consumption. This session closes with an introduction to literate programming to present results.

The case study for this session is smart meter data for a simulated water system, including leaks and other anomalies.



Digital metering diurnal curve (Case Study 3).

Capstone project

The capstone project involves developing a water quality dashboard for the Board of Directors of a water utility. The capstone project includes a problem statement, data sets and guidance on how to complete this challenge.

The capstone can be completed in many different ways, so participants need to use other resources to learn new techniques in the R language. The capstone chapter explains how to use the extensive help functionality within R and the best way to find assistance online.

Participant Activities

The content of this course contains several activities for participants. These icons are used throughout the text to indicate these activities:

Questions (The answers are available at the end of the course).

Tasks.

Points of discussion for face-to-face workshops. If you follow the online version, then you can add your answer to the course community.

Optional additional information to consider.

Prerequisites

To follow participate in this workshop, you need to have some understanding of the issues surrounding water management. Experience with analysing data is also preferred. This course is designed with spreadsheet users in mind. Experience with writing computer code is helpful, but not required.

You also need access to a recent version of the R language and RStudio. The best way to access the R language is by downloading the latest version from the website of the R Project for Statistical

Computing[4]. RStudio is an IDE (Integrated Development Environment) that simplifies working with R and data. You can download a free version of this software from the RStudio[5] website. Follow the instructions on these websites to complete the installation. You need to install the R language before you install the IDE.

- Go to the R Project download[6] site
- Download the *base* version for your operating system (approximately 80 megabytes)
- Install the software
- Go to the RStudio download page[7]
- Download the installer for the free version for your operating system
- Install the software

Alternatively, you can sign-up for a free account to access the cloud version[8] of R Studio. This account gives you full access to R Studio and R in your browser without the need to install any software. The cloud version is fully functional but not very fast. Installing R and RStudio on your laptop is the preferred method.

Downloading the workshop material

All resources for this workshop are available on the GitHub[9] website. GitHub is a repository for computer code and associated information that allows developers to share their work and collaborate.

You can download the documents by clicking on the 'clone or download' button and extract the files to your computer. You can open the RStudio project file to begin the workshop and start playing with the data and code.

If you use Git, then fork or clone the repository. Feel free to create an issue or pull request if you find errors or like to provide additional content.

For those using the cloud version of RStudio, click on the arrow next to the 'New Project' button and select 'New Project from GitHub Repo'. Copy the URL (`https://github.com/pprevos/r4h2o/`[10]) in the text field and hit enter. After a little while, RStudio opens the project.

The repository contains several folders:

- The manuscript folder source files of the course text, images and videos.
- The session folders contain the data and code for each of the sessions and case studies.

[4] https://www.r-project.org/
[5] https://www.rstudio.com/
[6] https://cran.r-project.org/
[7] https://www.rstudio.com/products/rstudio/download/
[8] https://rstudio.cloud/
[9] https://github.com/pprevos/r4h2o/
[10] https://github.com/pprevos/r4h2o/

The next chapter[11] introduces the principles of data science and presents a framework for good data science.

[11] https://leanpub.com/courses/leanpub/R4H2O/read/2

Data Science for Water Utilities

This first session provides a framework for good data science. The first section defines data science within the context of managing water and sewerage services. The second section presents a framework for good data science.

What is data science?

The earliest known form of writing is not an epic poem or religious text, but data. The Ishango bone[12] is an engraved fibula of a baboon which was carved in central Africa 20,000 years ago. Some scholars hypothesised that the carvings represent an early number system as it lists several prime numbers, while others believe it to be a calendar. Some researchers dismiss these ideas and believe that the markings merely improve grip when using the bone as a club. Whatever their purpose, the groupings of the markings are distinctly mathematical (Figure 1.1).

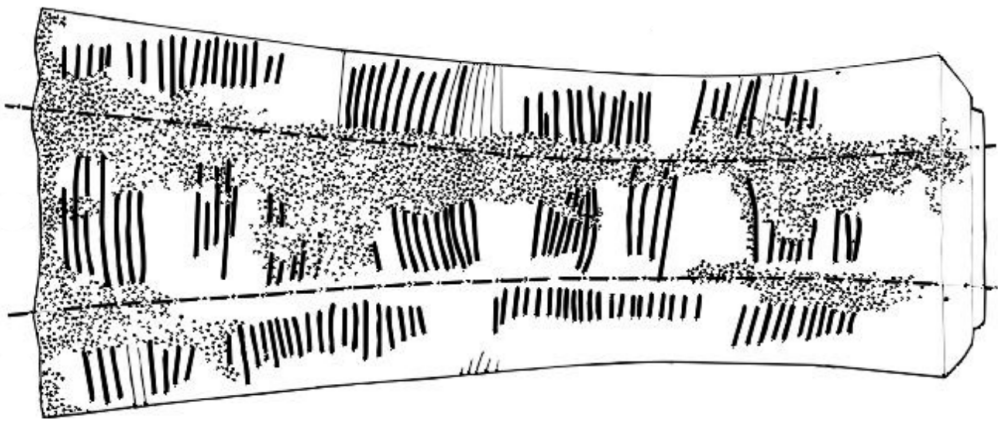


Figure 1.1: Markings on the Ishango Bone

The idea that data can be used to understand the world is thus almost as old as humanity itself and has gradually evolved into what we now call data science. Using data in organisations is also called business analytics or evidence-based management. There are also specific approaches, such as Six-Sigma, that use statistical analysis to improve business processes.

Although data science is merely a new term for something that has existed for decades, some recent developments have created a watershed between the old and new ways of analysing a business. The difference between traditional business analysis and the new world of data science is threefold.

Firstly, businesses have much more data available than ever before. The move to electronic transactions means that almost every process leaves a digital footprint. Collecting and storing this

[12] https://arxiv.org/abs/1204.1019

data has become exponentially cheaper than in the days of pencil and paper. Many organisations collect this data without maximising the value they extract from it. After the data served its intended purpose, it becomes 'dark data', stored on servers but languishing in obscurity. This data provides opportunities to optimise how an organisation operates by recycling and analysing it to learn about the past to create a better future.

Secondly, the computing power that is now available in a tablet was not long ago the domain of supercomputers. Piotr Luszczek[13] showed that an iPad-2 produced in 2012 matched the performance of the world's fastest computer in 1985. The affordability of enormous computing power enables even small organisations to reap the benefits of advanced analytics.

Lastly, sophisticated machine learning algorithms are freely available as Open Source software, and a laptop is all that is needed to implement sophisticated mathematical analyses. The R language for statistical computing and Python are potent tools that can undertake a vast array of data science tasks such as visualisations and machine learning. These languages are 'Swiss army chainsaws' that can tackle any business analysis problem. Part of their power lies in the healthy communities that support each other in their journey to mastering these languages.

These three changes have caused a revolution in how we create value from data. The barriers to entry for even small organisations to leverage information technology are low. The only hurdle to take is to make sense out of the fast-moving developments and follow a strategic approach instead of chasing the hype.

To what extent has your organisation digitised the collection of data? Are all sources of data available for analysis?

This revolution is not necessarily only about powerful machine learning algorithms, but about a more scientific way of solving problems. The vast majority of analytical problems in supplying water or sewerage services do not require machine learning to solve. The definition of data science in this book is not restricted to machine learning, big data and artificial intelligence. These developments are essential aspects of data science, but they do not define the field.

The expectations of data science are very high. Business authors position data science, and its natural partner 'big data', as a panacea for all societal problems and a means to increase business profits. In a 2012 article in *Harvard Business Review*, Davenport and Patil[14] even proclaimed data scientist the "sexiest job of the 21st century" (Figure 1.2). Who would not want to be part of a new profession with such enticing career prospects? The Google Trends[15] website shows a significant increase in the number of people searching for 'Data Science' since the publication of this article.

[13] https://www.phoronix.com/scan.php?page=news_item&px=MTE4NjU
[14] https://hbr.org/2012/10/data-scientist-the-sexiest-job-of-the-21st-century
[15] https://trends.google.com/trends/explore?date=all&q=data%20science

Data Scientist:

The Sexiest Job of the 21st Century

Meet the people who can coax treasure out of messy, unstructured data.

by Thomas H. Davenport and D.J. Patil

When Jonathan Goldman arrived for work in June 2006 at LinkedIn, the business networking site, the place still felt like a start-up. The company had just under 8 million accounts, and the number was growing quickly as existing members invited their friends and colleagues to join. But users weren't

Figure 1.2: Sexiest job of the 21st century?

For organisations that deliver physical products, data science is about improving how they collect, store and analyse data to extract more value from this resource. The objective of data science is not the data or the analysis itself but is closely intertwined with the strategic goals of the organisation. For a water utility, these objectives are generically maintaining or improving the experience that customers have with their service and minimising impact to the natural environment. Whatever kind of organisation you are in, the purpose of data science is to assist managers with changing reality to a more desirable state. A data scientist achieves this objective by measuring the current and past states of reality and using mathematical tools to predict a future state.

Data science is a systematic and strategic approach to using data, mathematics and computers to solve practical problems. The problems of data scientists are practical because the objectives of science are different objectives to business. A data scientist in an organisation is less interested in a generalised solution to a problem but focuses on improving how the organisation achieves its goals. In this sense, a data scientist is not strictly speaking a scientist.

The Elements of Data Science

Now that we have defined data science within the context of managing a water utility, we can start describing the elements of data science. The best way to unpack the art and craft of data science is Drew Conway's often-cited Venn diagram cite (Figure 1.3). Conway[16] defines three competencies that a data scientist, or a data science team as a collective, need to possess. The diagram positions data science as an interdisciplinary activity with three dimensions: domain knowledge, mathematics and computer science. A data scientist is somebody who understands the subject matter under consideration in mathematical terms and writes computer code to solve problems.

[16] http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram

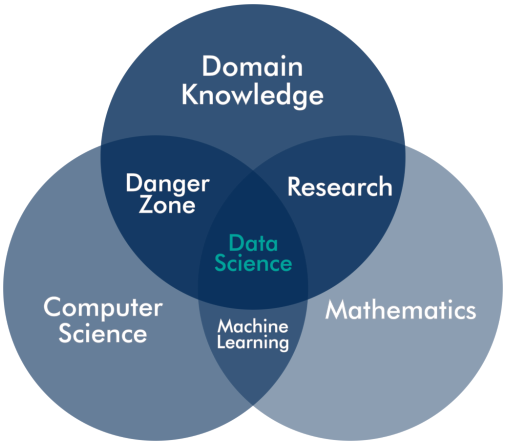


Figure 1.3: Conway's data science Venn Diagram.

Domain Knowledge

The most vital skill within a data science function is *domain knowledge*. While the results of advanced applied mathematics such as machine learning are impressive, without understanding the reality that these models describe, they are devoid of meaning and can cause more harm than good. Anyone analysing a problem needs to understand the context of the issues and the potential solutions. The subject of data science is not the data itself but the reality this data describes. Data science is about things and people in the real world, not about numbers and algorithms.

A domain expert understands the impact of any confounding variables on the outcomes. An experienced subject-matter expert can quickly perform a sanity check of the process and results of the analysis. Domain knowledge is essential because each area of expertise uses a different paradigm to understand the world.

Each domain of human enquiry or activity has different methodologies to collect and analyse data. Analysing objective engineering data follows a different approach to subjective data about people or unstructured data in a corpus of text. The analyst needs to be familiar with the tools of the trade within the problem domain. The earlier-mentioned example of a graduate professional beating a team of machine learning experts with a linear regression shows the importance of domain knowledge.

Domain expertise can also become a source of bias and prevent innovative ways of looking at information. Solutions that are developed through systematic research can contradict long-held beliefs that are sometimes hard to shift. Implementing data science is thus as much a cultural process as it is a scientific one.

Mathematical Knowledge

The analyst uses mathematical skills to convert data into actionable insights. Mathematics consists of pure mathematics as a science in itself and applied mathematics that helps us to solve problems.

The scope of applied mathematics is broad, and data science is opportunistic in choosing the most suitable method. Various types of regression models, graph theory, k-means clustering, decision trees, and so on, are some of the favourite tools of a data scientist.

Combining subject-matter expertise with mathematical skills is the domain of traditional *research* and analysis. Also, academics are moving towards integrating abilities in computer science with their work towards a data science approach of research.

Numbers are the foundations of mathematics, and the craft of quantitative science is to describe our analogue reality into a model that we can manipulate to predict the future. Not all mathematical skills are necessarily about numbers but can also revolve around logical relationships between words and concepts. Contemporary numerical methods help us to understand relationships between people, the logical structure of a text and many other aspects beyond the realm of traditional quantitative analysis. These types of analysis are outside the scope of this course.

Computer Science

Not that long ago, most of the information collected by an organisation was stored on paper and archived in copious volumes of arch lever files. Analysing this information was an arduous task that involved many hours of transcribing information to a format that is useful for analysis.

In the twenty-first century, almost all data is an electronic resource. To create value from this resource, data engineers extract it from a database, combine it with other sources and clean the data before analysts can make sense of it. This requirement implies that a data scientist needs to have computing skills. Conway uses the term hacking skills, which many people interpret as unfavourable. Conway is, however, not referring to a hacker in the sense of somebody who nefariously uses computers, but in the original meaning of the word of a developer with creative computing skills. The core competency of a hacker, developer, coder, or whatever other terms might be preferable, is algorithmic thinking and understanding the logic of data structures. These competencies are vital in extracting and cleaning data to prepare it for the next step of the data science process.

The importance of hacking skills for a data scientist implies that we should move away from point-and-click systems and spreadsheets and instead write code in a suitable programming language. The flexibility and power of a programming language far exceed the capabilities of graphical user interfaces and leads to reproducible analysis.

The mathematical interpretation of reality needs to be translated to computer code. One of the factors that spearheaded data science into popularity is that the available toolkit has grown substantially in the past ten years. Open source computing languages such as R and Python are capable of implementing complex algorithms that were previously the domain of specialised software and supercomputers. Open Source software has accelerated innovation in how we analyse data and has placed complex machine learning within reach of anyone willing to make an effort to learn the skills.

Conway defines the *danger zone* as the area where domain knowledge and computing skills combine, without a good grounding in mathematics. Somebody might have sufficient computing

skills to be pushing buttons on a business intelligence platform or spreadsheet. The user-friendliness of some analysis platforms can be detrimental to the outcomes of the analysis because they create the illusion of accuracy. Point-and-click analysis hides the inner workings from the user, creating a black-box result. Although the data might be perfectly structured, valid and reliable, a wrongly-applied analytical method leads to useless outcomes.

The Unicorn Data Scientist?

Conway's diagram is often cited in the literature on data science. His simple model helped to define the craft of data science. Other data scientists have proposed more sophisticated models, but they all originate with Conway's basic idea. A quick internet search reveals several variants.

The diagram illustrates that the difference between traditional research skills or business analytics exists in the ability to understand and write code. A data scientist understands the problem they seek to resolve, they understand the mathematics to analyse the problem, and they possess the computing skills to convert this knowledge into outcomes.

The so-called soft skills seem to be missing from this picture. However, communication, managing people, facilitating change, and so on, are competencies that belong to every professional who works in a complex environment, not just the data scientist.

Some critics of this idea point out that these people are unicorns. Data scientists that possess all these skills are mythical employees that don't exist in the real world. Most data scientists start from either mathematics or computer science, after which it is hard to become a domain expert.

This course is written from the point of view that we can breed unicorns by teaching domain experts how to write code and, where required, enhance their mathematical skills. Teach water professionals to understand the principles of data science and write code helps an organisation's ability to embrace the benefits of the data revolution.

Many data scientists have published modifications of this model. Can you think of some other competencies?

The Water-Data Value Chain: The Digital Water Utility

The flow of data in a utility follows the flow of the water through the value chain. The water value chain (Figure 1.4) starts and ends in the natural environment. Water utilities extract water from nature, process it in their value chain, and eventually return it to the environment.

Water utilities collect data along the flow path of the water. This data describes the quantity and quality of the water, including wastewater, as it makes its way from the environment to the consumer and back. The data derived from instrumentation provides an objective view of the status of the water supply chain. Customer-centric water utilities also collect data from the perspective of

the consumers of the services they supply. This data is, by definition, subjective. Data science for water utilities merges the objective measurements from the field with the subjective perspectives of customers to maximise value to the community overall.

The term ‘digital water utility’ is often used to describe the situation where the flow of water and customer experience is fully captured with data. Some experts even suggest that digitisation represents a disruption of water utilities. The term digital water utility is a distraction because data is not a replacement for effective water management. No matter how much water utilities digitise, electronics will not meaningfully change the service utilities provide: a reliable supply of drinking water and sewerage services.

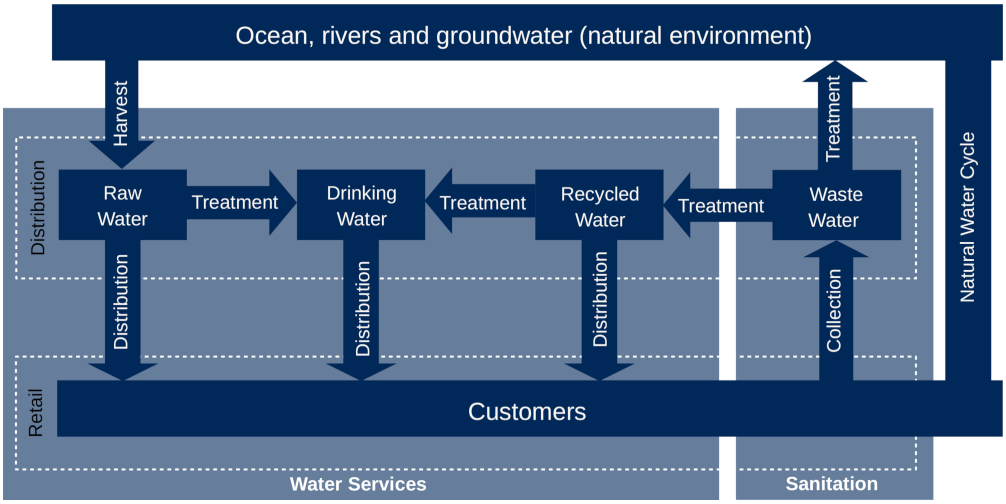


Figure 1.4: Tap water supply chain.

Digitisation also has limitations. Firstly, data cannot describe everything. Measuring physical processes is only ever a sample of the reality we seek to control. Secondly, the experience of customers is subjective, which requires human insight to understand. These limitations highlight the need for domain expertise to complement skills in mathematics and computing. Relying on data alone, without recognising the physical and social reality of water management does not add value to a community.

Data Science Tools

The last decade has seen an explosion of available data science tools. There is no one single tool that can do everything. Just like a trades-person uses each tool for a specific activity, so does a data scientist use tools for particular tasks within the workflow.

Spreadsheets

Spreadsheets are the most common tool to solve data problems. They are a great product that combines storing data, writing and executing code and displaying output. Their versatility is also their Achilles heel. Spreadsheets have limited capabilities and some intrinsic constraints.

Spreadsheets are straightforward to build, but they are almost impossible to reverse-engineer (Figure 1.5). We all would have had the unpleasant experience of trying to understand how a spreadsheet made by somebody else, or one that you did ages ago, actually functions. The biggest issue with spreadsheets is the reproducibility of the analysis process.

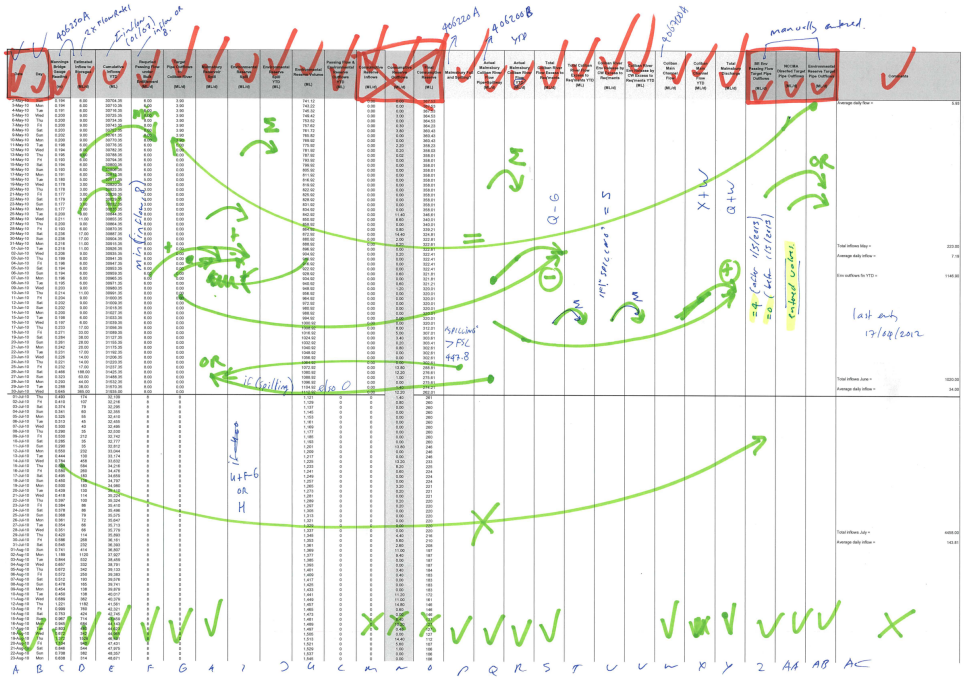


Figure 1.5: Reverse-engineering a spreadsheet.

Business Intelligence Systems

The software market is saturated with point-and-click business intelligence systems, such as Qlik[17], Tableau[18] or Microsoft Power BI[19]. These tools are user-friendly portals for end-users to consume data. Business intelligence tools are, however, not an ideal tool to analyse data. Their main strength is to present the results of analysis without providing access to the underlying steps and statistics.

Another limitation of these systems is that they are limited to visualisations, without any meaningful capacity to include a narrative. Business intelligence tools are almost like a 'choose your own adventure'. The user can choose how the system presents the data and thus create their owns stories. While a well-designed visualisation is, as they say, worth a thousand words, the complexity of the analytical process often needs a narrative to help the reader understand the purpose, method and conclusions. Limiting a data product to visualisations creates appealing visuals, but the narrative is lacking.

[17] https://www.qlik.com/
[18] https://www.tableau.com/
[19] https://powerbi.microsoft.com/

Data science code

Writing computer code has long been the domain of information technology professionals. Stereotypes of coders do not help this view as slightly eccentric geeks that prefer to communicate with their terminal instead of with people. The main objective of this course is to dispel this false idea and promote that water professionals should ditch their spreadsheets and learn how to write code.

For those who develop spreadsheets, the jump to writing code is not as massive as it might seem. Every formula in a spreadsheet is -pin essence a part of a computer program.

There are almost as many computer languages as there are human ones. Many of these languages are suitable to analyse data, and the list in this section only includes the most common ones. Some languages, such as Python, C or Java are general programming languages that can create any software. Other languages are explicitly developed to manipulate and analyse data.

The Structured Query Language (SQL, pronounced sequel) is a language to access and manipulate databases. Many varieties of SQL exist, but they all have many similarities. The main strength of SQL is its ability extract, transform and load data. The first version of this language was released in 1986, and it is a robust data interface. This language is not very good at actually analysing data because it does not include any higher-order mathematics.

Python is a general-purpose programming language that developers use to develop many types of applications. Python has many extensions with specific data science functions. Some people are passionate about why they use either Python or R. Both languages have their strengths and weaknesses, and complex data science projects combine these languages.

There are many other less well-known programming languages specialised such as Julia, Haskell, Fortran, Mathematica, and so on.

This workshop uses the R language because it is designed to analyse data. The basic functionality of R includes many higher-order functions to undertake statistical analysis. This course is about the R language mainly because the way it is structured is close to the way subject-matter-experts think about analysis, instead of the way computer scientists structure software. The RStudio development environment, combined with R provides a potent tool for analysing data and presenting the results.

Good Data Science

The question that arises from this introduction is how to manage and analyse data so it can become a valuable resource. These final sections present a normative model to create value from data using three basic principles derived from architecture. This model is useful for data scientists as an internal check to ensure that their activities maximise value. Managers can use this model to assess the outcomes of a data science project without having to understand the mathematical intricacies of the craft and science of analysis.

The three case studies of this course implement these principles so that participants not only learn R syntax but also best practice in analysing data.

Data Science Trivium

Although data science is a quintessential twenty-first-century activity, to define good data science, we can find inspiration in a Roman architect and engineer who lived two thousand years ago. Vitruvius wrote his books *About Architecture*, which inspired Leonardo da Vinci to draw his famous Vitruvian man. Vitruvius wrote that an ideal building must exhibit three qualities: *utilitas*, *firmitas* and *venustas*, or usefulness, soundness and aesthetics.

Buildings must have utility so people can use them for their intended purpose. A house needs to be functional and comfortable, and everybody in a theatre needs to see the stage. Each type of building has specific functional requirements. Secondly, buildings must be sound in that they are firm enough to withstand the forces that act upon them. Last but not least, buildings need to be aesthetic. In the words of Vitruvius, buildings need to look like Venus, the Roman goddess of beauty and seduction.

The Vitruvian rules for architecture can also define good data science. Excellent data science needs to have utility; it needs to be useful to create value. The analysis should be sound so it can be trusted. The products of data science also need to be aesthetic to maximise the value they provide to an organisation (Figure 1.6).

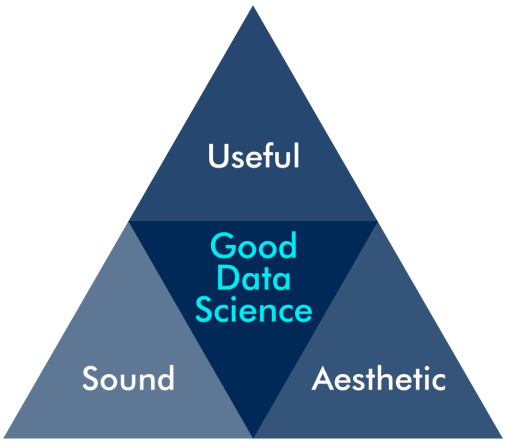


Figure 1.6: The principles of good data science.

Useful Data Science

How do we know that something is useful? The simple, but not very illuminating answer is that when something is useful, it has utility. Some philosophers interpret utility as the ability to provide the greatest good for the highest number of people. This definition is quite compelling, but it requires contextualisation. What is right in one situation might not be so beneficial in another.

The highest number of people is open to interpretation. Is something only useful when it benefits all of humanity, or can it also be useful when it helps just one person? The requirement to include the highest number of people in our definition of usefulness might work well for government organisations. This rule is not so evident in corporations that seek to maximise the benefits to their shareholders.

Whether something is useful or not depends on context, but also on the values used to judge them. Defining usefulness in generic terms is an impossible quest because of the dependence on the context and the relevant value system. For example, to Greenpeace, analysing data from fracking activities has a different level of usefulness as it does to a gas exploration company. The same data can satisfy different types of merit depending on the context.

These philosophical deliberations aside, defining usefulness for organisations is more straightforward because we apply a pragmatic approach. Usefulness for organisations is the extent to which something contributes to their strategic or operational objectives. If the result of a data science project is unable to meet this criterion, then it is strictly speaking useless. As a civil engineer and social scientist, I could spend many hours analysing the vast amounts of data collected by my organisation. Dredging the data to find something of value might be an exciting way to waste time, there is also a significant risk of finding fool's gold instead of valuable nuggets of wisdom. The first step that anyone working with data should undertake before starting a project is to define the business problem that needs solving.

This book follows a pragmatic and perspectivist view of usefulness. For a data science strategy to be successful, it has to facilitate the objectives of the organisation. Data scientists are opportunistic in the approach they use to resolve problems. Perspectivism implies that the same data can be used for different issues, depending on the perspective you take on the available information and the problem at hand.

After digesting a research report or viewing a visualisation, managers ask themselves: "What do I do differently today?" Usefulness in data science depends on the ability of the results to empower professionals to influence reality positively. In other words, the conclusions of data science either comfort management that objectives have been met or they provide actionable insights to resolve existing problems or prevent future ones.

Providing actionable intelligence is only a narrow scope of works for data scientists. The concept of usefulness in business needs to be extended beyond this short term and one-dimensional view. Business scholar Bernard Jaworski[20] classified the results of research into two types to help us make sense of how theory relates to practice. Some knowledge is suitable for action, which is the much sought-after actionable intelligence. Other research doesn't lead to action but inspires deeper thinking about managerial practice. Data science can direct action, but it can also motivate innovation by providing a more profound understanding of the current reality. The results of data science should either stimulate action or inspire contemplation. While the relevance of taking action is self-evident, managers often fail to recognise contemplation as a beneficial managerial impact.

[20] https://doi.org/10.1509/jmkg.75.4.211

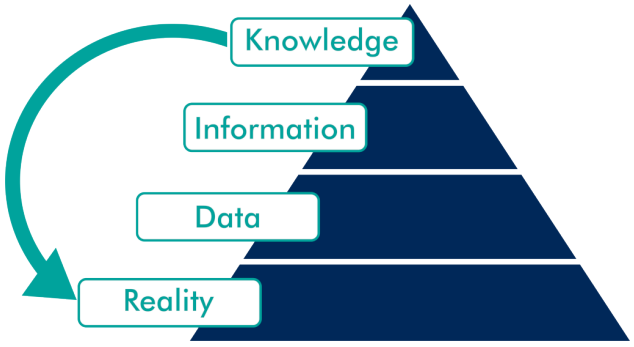


Figure 1.7: Reality, Data, Information, Knowledge Pyramid.

For data science to provide actionable intelligence, the raw data needs to be converted to knowledge following a standardised workflow. The well-known DIKW Pyramid (Data, Information, Knowledge and Wisdom) explains how data produces a useful analysis. The source of the original version of this model is lost in time as a multitude of authors has used it without citation. The basic principle of the hierarchy is that to obtain wisdom; you need to have the relevant knowledge, which derives from information, which in turn consists of the conclusions drawn from the data. Various versions of the model have been proposed, with slightly different terminology and interpretations.

The version in this book is modified to understand better how to create useful data science (Figure 1.6). Firstly, wisdom no longer forms part of the model because this concept is too nebulous to be helpful. Anyone seeking wisdom should study philosophy or practice religion as data science is unable to provide this need. Secondly, the bottom of the pyramid needs to be grounded in reality. The standard DIKW model ignores the reality from which the data is collected that creates the information and knowledge used to make business decisions. The second addition to the traditional model is a feedback loop from knowledge to the real world. The purpose of data science is to enhance the knowledge that professionals use to influence reality by converting data into information.

Reality

Useful data science positively influences reality by collecting data, creating information and increasing our knowledge about and understanding of reality. This knowledge is useful when it changes the way we perceive reality to innovate the way we do things, and when it enables better operational or strategic decisions. When data science is abstracted from the world it seeks to understand or influence, it loses its power to be valuable.

This reality of data science can be either physical or social, each of which requires a different paradigm to describe the world. Our physical reality can be measured with almost arbitrary precision. We can measure size, weight, chemical composition, time, and so on, with high validity and reliability.

The social world can also be summarised in numbers, but these measurements are almost always indirect. We cannot read people's minds. When we want to know how somebody feels about a level of service or another psychological parameter, we can only indirectly measure this variable. Data

from the social world is often qualitative and requires different considerations than in the physical world.

The complicated relationship between the data and the reality it seeks to improve emphasises the need for subject-matter expertise about the problem under consideration. Data should never be seen as merely an abstract series of numbers or a corpus of text and images, but should always be interpreted in its context to do justice to the reality it describes.

Data

Data is the main ingredient of data science, but not all data sources provide the same opportunities to create useful data products. The quality and quantity of the data determine its value to the organisation. This mechanism is just another way of stating the classic *Garbage-In-Garbage-Out* (GIGO) principle. This principle derives from the fact that computers blindly follow instructions, irrespective of truthfulness, usefulness or ethical consequences of the outcomes. An amazing algorithm with low quality or insufficient data is unable to deliver value to an organisation. On the other hand, analysing high-quality data with an invalid algorithm results in 'garbage', instead of valuable information.

The quality of data relates to the measurement processes used to collect the information and the relationship of this process to the reality it describes. The quality of the data and the outcome of the analysis is expressed in their validity and reliability. The next discusses the soundness of data and information in more detail.

The next step is to decide how much data to collect. Determining the appropriate amount of data is a balancing act between the cost of collecting, storing and analysing the data versus the potential usefulness of the outcome. In some instances collecting the required data can be more costly than the benefits it provides.

The recent steep reduction in the cost of collecting and storing data seems to render the need to be selective in data gathering a moot point. Data scientists might claim that we should collect everything because a time machine is much more expensive than collecting and storing more data than strictly necessary. Not measuring parts of a process has an opportunity cost because future benefits might not be realised. This opportunity cost needs to be balanced with the estimated cost of collecting and storing data.

This revolution in data gathering is mainly related to physical measurements and the so-called Internet of Things, mobile phones and other wearable devices. Measurement devices and the transmission and storage of data are affordable, so carpet-bombing a region with sensors to collect data becomes more feasible. Some aspects of reality remain complicated and expensive to measure as the Internet of Things cannot be applied everywhere. Assessing how people feel about something, their intentions, and so on will, until we have access to cost-effective mass mind reading, remain a complicated and expensive undertaking.

One guideline to determine what and how often to collect is to work backwards from the sought benefits. Following the knowledge pyramid, we should collect data that enables us to influence

reality positively. The frequency of collection is an outcome of the statistical power that is required to achieve the desired objectives. In most cases, the more data is available, the higher the statistical power of the analysis.

The amount of data points required to achieve a specific outcome also depends on the type of data. The more reliable and valid the measurements, the fewer data points are needed to obtain a reliable sample. Lastly, the need to ensure that a sample represents the population it describes defines the minimum size of the sample in a social context. Determining sample sizes is a complex topic, and the statistics literature provides detailed information about how much data to collect to achieve the required statistical power.

Gathering data about people because it might be useful in the future also has ethical consequences. Storing large amounts of personal data without a defined need can be considered unethical because the data might be used for a purpose for which the subjects did not consent. Medical records are a case in point. They are collected to manage our health and not for insurance companies to maximise their profits.

A case study illustrates how to decide the ideal amount of data to collect. A water utility discussed how much data they wanted to gather to measure how much water customers use. The existing method only provided one data point for each water meter every three months. The water engineers would ideally like a reading every five minutes, while the billing department was more than happy with one daily reading.

New technology became available that collects data at a higher frequency. However, the higher the rate, the higher the cost of collection due to transmission bandwidth and battery life. Collecting data every five minutes was considered to be unfeasible and potentially unethical because it reveals too much about the lifestyles of customers. Daily data was insufficient to provide benefits in network design and operation. The utility decided to collect hourly data because it allows for most of the sought benefits, doesn't significantly impact the privacy of customers and is within reasonable reach for the current level of technology.

Information

Within the context of this book, information is defined as processed data. Information is data placed with the context of the reality from which it was extracted. To ensure information is sound and useful, professionals need to use an appropriate methodology, logically present the information and preserve the results for future reuse or review.

Data scientists use an extensive range of methods to convert data into information. At the lowest level, summarising the averages of the various data points converts provides some value. More sophisticated analysis transforms data about the past into a prediction of the future. These techniques require a solid understanding of mathematics and analytical methods to ensure they don't result in data pseudo data science.

Communicating information is where art meets data science. Writing useful reports and designing meaningful visualisations ensures that a data product is useful.

Lastly, the information needs to be preserved so that it is accessible to those who need it in the future or for those who seek to review the methods.

Knowledge

Professionals with subject-matter expertise gain knowledge from the results of data science, which they use to decide on future courses of action. Knowledge can be either tacit or explicit. The result of a data science project, also known as a data product, is explicit knowledge, which can be transferred through writing or instruction.

Numbers and visualisations help professionals to understand the reality they need to manage. This process of understanding and using these results in practice leads to tacit knowledge, which is the essence of domain expertise. Tacit knowledge is difficult to transfer because it consists of a combination of learnt explicit knowledge mediated through practical experience.

Data science thus not only requires domain expertise to be useful, but it can also increase this expertise. This topic is outside the scope of data science as it ventures into knowledge management.

Feedback Loop

The last and most important part of this data science model is the feedback loop from knowledge back to reality. The arrow signifies actionable intelligence, which is how reality is improved through knowledge. The key message of this section is that the results of data science need to either lead to a different way of thinking about a problem or provide actionable intelligence to propose.

Either option eventually leads to improved decisions using the best available data and analysis. Care needs to be taken, however, that the correct conclusions are drawn. The GIGO principle only covers the input of the process, but also the process itself needs to be sound. Although the data might be of good quality, a lousy analysis still result in ‘garbage’. The next two sections discuss how we can ascertain whether the outcomes of data science are sound and ensure the user draws the correct conclusion from the information.

Sound Data Science

Just like a building should be sound and not collapse, a data product needs to be sound to be able to create business value. Soundness is where the science and the data meet. The soundness of a data product is defined by the validity and reliability of the analysis, which are well-established scientific principles (Figure 1.8). The soundness of data science also requires that the results are reproducible. Lastly, the data and the process of creating data products need to be governed to assure beneficial outcomes.

The distinguishing difference between traditional forms of business analysis and data science is the systematic approach to solving problems. The key word in the term data science is thus not data but

science. Data science is only useful when the data answers a helpful question, which is the science part of the process.

This systematic approach ensures that the outcomes of data science can be relied upon to decide on alternative courses of action. Systematic data science uses the principles of scientific enquiry, but it is more pragmatic in its approach. While scientists search for general truths to explain the world, data scientists pragmatically seek to solve problems. The basic principles that underpin this systematic approach are the validity, reliability and reproducibility of the data, the methods and the results.

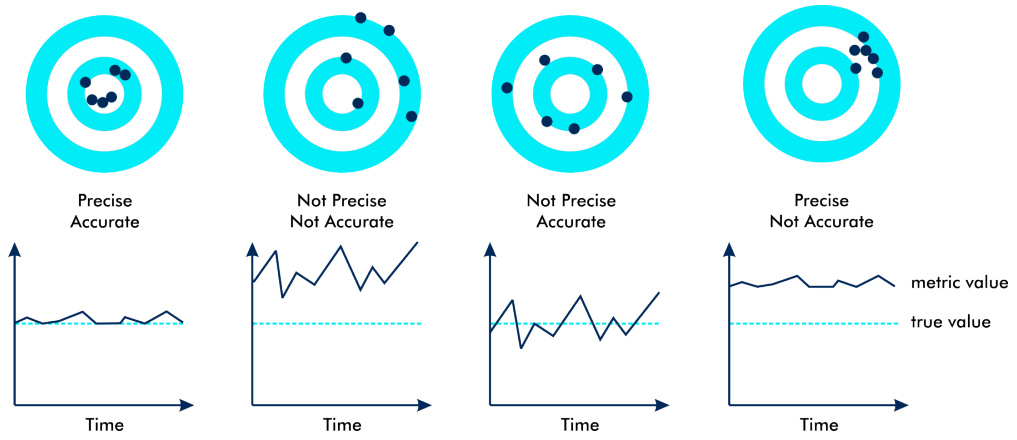


Figure 1.8: Visualising validity and reliability.

Validity

The validity of a data set and the information derived from it relates to the extent to which the data matches the reality it describes. The validity of data and information depends on how this information was collected and how it was analysed.

For physical measurements, validity relates to the technology used to measure the world and is determined by physics or chemistry. If, for example, our variable of interest is temperature, we use the fact that materials expand, or their electrical resistance increases, when the temperature rises. Measuring length relies on comparing it with a calibrated unit or the time it takes light to travel in a vacuum. Each type of physical measurement uses a physical or chemical process, and the laws of nature define validity. When measuring pH, for example, we want to be sure that we measure the power of hydrogen ions and not some other chemical property.

Mental processes, such as customer satisfaction or personality, are much harder to measure than physical properties. Although a state of mind is merely a pattern of electrical and chemical activity in the brain, no technology can directly measure it. Not much is known about the relationship between the physical events in our mind and our feelings, motivations and other psychological states.

Not all data about people has a validity problem. Observations that relate directly to our behaviours, such as technology that tracks our movements, or eye-tracking equipment to record our gaze, are physical measurements. Demographic data is a direct measurement of a social state. However, even

seemingly simple aspects such as gender can lead to significant complexity when trying to measure it. What often seems a simple demographic variable can be quite complicated to define.

Scientists use sophisticated machinery to scan brains to discover how our mind functions. Marketers regularly use this technology to understand customers. Neuromarketing produces insights from brain scans to fine-tune the design of products and marketing communication to maximise the likelihood that a customer purchases their offering. Brain scanning gives insight into the processes inside our brain, but it is quite expensive and intrusive. Scanning technologies are insightful, but not an efficient way to get to know your customers.

In practice, social scientists and psychologists use psychometrics to measure states of mind by using survey techniques indirectly. We might ask a customer whether they agree or disagree with a series of statements such as "The hotel room was comfortable". Most commonly, these questions are measured using a Likert scale with five or seven descriptors.

The basic principle of surveys to measure a state of mind is that the mental processes we seek to understand causes the subject to answer a survey question in a certain way. People who think that the hotel room was very comfortable fully agree with the statement in the survey. The statistical analysis of such questions seeks to reverse this causality to learn about the subject's thoughts and feelings.

The variables that we are interested in are latent variables because they are hidden within the mind of the subject and only reveal themselves indirectly through the survey answers. The validity of psychometric measurements is a complex topic with many types of validity, each with their specific purpose. A vast field of literature describes how to measure and analyse latent variables.

Another method to understand people is to use a Big Data approach. This technology does not rely on surveys or brain scans from a sample of the population but uses our behaviour on social media or measured through a wearable device as a proxy for our psychology. The big data approach is entirely different from any of the other techniques.

Brain scanning and psychometrics aim to understand future behaviour by indirectly measuring what we think, feel and believe. One of the main problems with surveys is that our answers are biased and perhaps not an accurate reflection of what we believe. Brain scans are impressive but are still only a proxy for our internal states of mind.

Big data methods measure our actual behaviour by recording what we purchase, where we travel, what we search for, and so on. The other significant difference between big data and traditional approaches is that the first two methods rely on a small sample of the population, while big data approaches, by their very nature, include millions of subjects. The validity of this information is very high because it measures actual behaviour instead of indirect parameters.

Reliability

The reliability or accuracy of physical measurements depends on the quality of the instrumentation used to obtain the data. Engineers spend much effort to assure the reliability of instrumentation

through maintenance and calibration programs. The instruments need to be installed and maintained to the manufacturer's specifications to ensure their ongoing accuracy. Quality assurance of instrumentation is possibly the most costly aspect of collecting and storing data about physical processes.

Several methods exist to test the reliability of psychological survey data. One simple test is to check for respondents that provide the same answer to all questions. The chances that somebody would genuinely answer "Neither agree or disagree" to all items is negligible, and it is good practice to remove these people from the sample. Researchers also use questions to trap fake responses. The researcher can, for example, ask people whether they agree or disagree with certain factual statements, such as: "You live in Australia." Any subject not wholly agreeing with this question (assuming this is an Australian survey) should be removed from the sample.

Brain scanning technology relies on small samples of the population because of the cost of the technology. These methods are sensitive to error, illustrated by an infamous example. Craig Bennett and Abigail Baird placed a dead Atlantic salmon[21] in an fMRI scanner and were surprised to find brain activity. They published their results to warn scientists of the risk of unreliable statistical methods. The spoof scientific journal Annals of Improbable Research[22] awarded Bennet and Baird with the igNobel prize. This annual prize awards unusual and trivial results in science.

The reliability of big data approaches is arguably very high because people provide this information not to satisfy the need of a researcher but to guide their actions. Rather than asking somebody whether they will purchase something in the future, our actual purchase patterns are naturally a strong predictor for future purchases.

Reproducibility

The third aspect of the soundness of a data product is its reproducibility, which is the ability for other people to reconstruct the workflow of the analyst from raw data collection to reporting. This requirement is a distinguishing factor between traditional business analysis and data science. The condition of reproducibility ensures that managers who base business decisions on a data product can review how the results were obtained or at least have trust in the results because they are potentially auditable. Reproducibility ensures that peers can evaluate all analysis and negates the problems of black boxes.

A specific aspect of machine learning that relates to its reproducibility is whether the user can understand how the model came to a conclusion. Can the data scientist explain to the subject-matter expert how the model works? The great benefit of machine learning is that these algorithms can detect patterns in large volumes of data that are impossible for humans to comprehend within a lifetime. More often than not, however, machine learning results in a black box that converts data into output. Although the algorithms are reproducible in that they produce the same result with the same data, they are not necessarily *explainable*. The outcomes of, for example, a random forest model, can take tens of pages to print and the logic is hard to verify for humans.

[21] http://prefrontal.org/files/posters/Bennett-Salmon-2009.pdf
[22] https://www.improbable.com/ig/winners/#ig2012

Machine learning algorithms are sometimes simplified so that the responsible managers can understand the logic of the decisions they make. In these cases, reliability is sacrificed for greater explainability. Whether a data science product is explainable depends not only on the code itself but also on the level of mathematical insight of the consumer of the outcomes. Explainability is thus a direct result of the level of data literacy of the organisation.

Governance

The fourth aspect of sound data science is governance. The process of creating data products needs to be documented in line with quality assurance principles. Practical considerations such as naming conventions for scripts, coding standards to ensure readability and so on are a necessary evil when managing complex data science projects.

The same principle also applies to managing data. Each data source in an organisation needs to have an owner and a custodian who understands the relationship between this data and the reality from which it is extracted. Large organisations have formal processes that ensure each data set is governed and that all employees use a single source of truth to safeguard the soundness of data products.

Governance is a double-edged sword as it can become the 'wet blanket' of an organisation. When governance becomes too strict, it smothers the very innovation that data science is expected to deliver. The art of managing a data science team is to find a middle way between strictly following the process and allowing for deviations of the norm to foster innovation. Good governance minimises risk, while at the same time enabling positive deviance that leads to better outcomes.

Aesthetic Data Science

Vitruvius insisted that buildings, or any other structure, should be beautiful. The aesthetics of a building causes more than just a pleasant feeling. Architecturally designed places stimulate our thinking, increase our well-being, improve our productivity and stimulate creativity.

While it is evident that buildings need to be pleasing to the eye, the aesthetics of data products might not be so obvious. The requirement for aesthetic data science is not a call for beautification and obfuscation of the ugly details of the results. The process of cleaning and analysing data is inherently complex. Presenting the results of this process is a form of storytelling that reduces this complexity to ensure that a data product is understandable.

The data science value chain starts with reality, described by data. This data is converted to knowledge, which managers use to influence reality to meet their objectives. This chain from reality to human knowledge contains four transformations, each with opportunities for a loss of validity and reliability. The last step in the value chain requires the user of the results of data science to interpret the information to draw the correct conclusion about their future course of action. Reproducibility is one of the tools to minimise the chance of misinterpretation of analyses. Another mechanism to ensure proper interpretation is to produce aesthetic data science.

Aesthetic data science is about creating a data product, which can be a visualisation or a report, that is designed so that the user draws correct conclusions. A messy graph or an incomprehensible report

limits the value that can be extracted from the information. The remainder of this section provides some guidelines on designing useful visualisations and writing reports.

Visualisation

Data visualisations are everywhere. They are no longer the domain of scientific publications and business reports. Publications in every medium use graphs to tell stories. The internet is awash with infographics on a wide range of topics. These popular images are often data science porn because they are designed to entertain and titillate, with limited usability from a business perspective. They are a fantastic tool to supply information to customers but should not be used to report data science.

Aesthetics and usefulness go hand in hand. Some data visualisations in engineering remind me of a Jackson Pollock[23] painting, with multitudes of lines and colours splashed over the screen. Adding too much information to a graph and using too many colours reduces its usability. When visualisation is not aesthetic, it becomes harder to interpret, which leads to the wrong conclusions and can even deceive the user.

Jackson Pollock (1952) Blue Poles number 11. Drip Painting in enamel and aluminium paint with glass on canvas (National Gallery, Canberra. Source: Wikimedia).

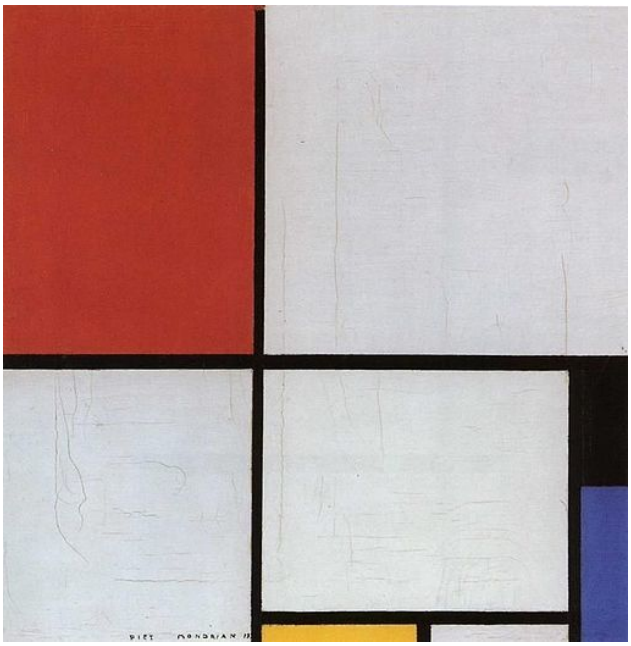
A data scientist needs to be aware of cognitive biases to prevent them and create data products that don't deceive. Many of these biases relate to how information is presented.

Our perception is not always an accurate representation of reality, and we often misinterpret the images that our retina collects. Optical illusions are funny internet memes, but they also occur in real life. Besides optical illusions, messy visualisations are hard to interpret because our mind does not know what element to focus on. A messy graphic confuses the brain so that it starts to form its interpretations.

Perhaps a good data visualisation should look more like a painting by Piet Mondrian[24] who is famous for his austere compositions with straight lines and primary colours. Using art to explain data

[23] https://en.wikipedia.org/wiki/Jackson_Pollock
[24] https://en.wikipedia.org/wiki/Piet_Mondrian

visualisation is not an accidental metaphor because visual art represents how the artist perceives reality. This comparison between Pollock and Mondrian is not a judgement of their artistic abilities. For Pollock, reality was chaotic and messy, while Mondrian saw a geometric order behind the perceived world.



Piet Mondrian (1928) Composition with red, yellow and blue. Oil on canvas (Municipal Museum, the Hague).

Although visualising data has some parallels with art, it is very different. All works of art are a form of deception. The artist paints a three-dimensional image on a flat canvas, and although we see people, we are just looking at blobs of paint. Data visualisation as an art form needs to be truthful and not deceive, either intentionally or accidentally. The purpose of any visualisation is to validly and reliably reflect reality.

Aesthetic data science is not so much an art as it is a craft. Following some basic rules prevents confusing the consumers of data products. Firstly, visualisation needs to have a straightforward narrative. Secondly, visualising data should be as simple as possible, minimising elements that don't add to the story.

Storytelling

First and foremost, visualisation needs to tell a story. The story in data visualisation should not be a mystery novel. A visualisation should not have suspense but get straight to the point. Trying to squeeze too much information into one graph confuses the reader. Ideally, each visualisation should contain only one or two narratives. If there is more to tell, then use more charts and create a dashboard.

Numerical data can contain several types of narratives. A graph can compare data points to show a trend among items or communicate differences between them. Bar charts are the best option to

compare data points with each other. A line graph is possibly your best option to compare data points over time. The distribution of data points is best visualised using a histogram. Scatter plots or bubble charts show relationships between two or three variables (Figure 1.9).

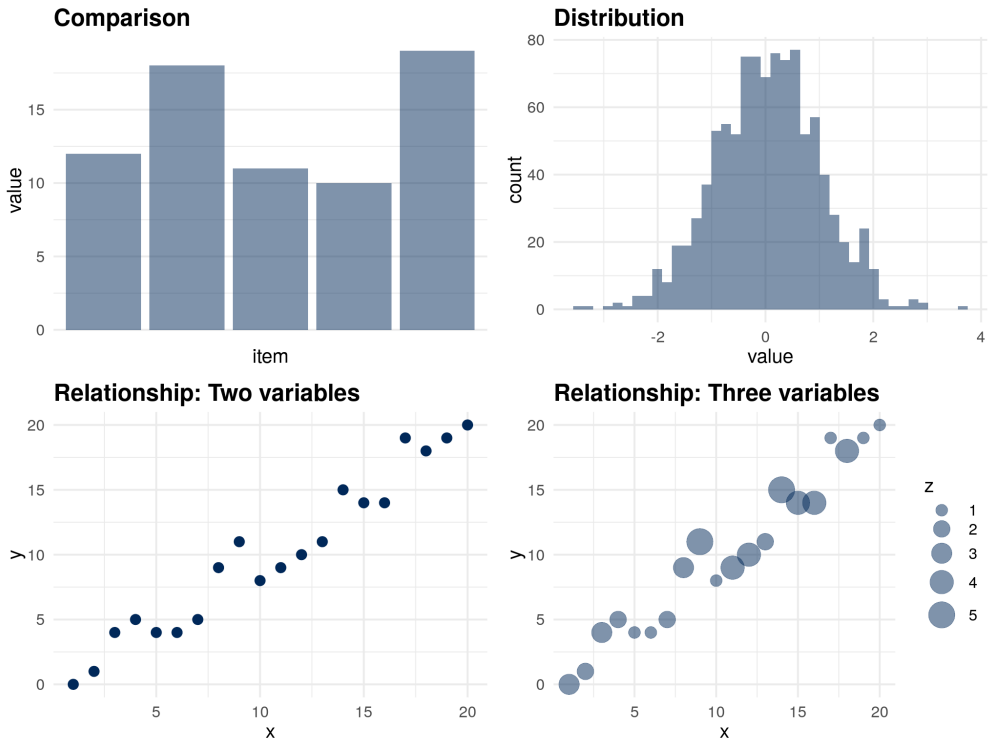


Figure 1.9: Examples of stories with quantitative data.

The detailed considerations of choosing the most suitable visualisation are outside the scope of this book. The Chart Chooser[25] website provides a dynamic interface to select the best graph to tell a story. This website uses a method developed by Andrew Abela[26]. You can download a high-resolution poster of the Chart Chooser from his website. The main point is that every visualisation needs to tell a story and not just summarise a bunch of data.

You are writing a new water resource plan and like to visualise the relationship between daily consumption per property, the size of the property and the number of inhabitants. Use the Chart Chooser website or poster and choose the most suitable visualisation.

Visualising qualitative information is a language with many options to tell a story. Displaying qualitative information is more an art than a craft because there is less reliance on mathematics. Network diagrams are another common visualisation tool. Networks are a convenient method to analyse relationships between people or other qualitative entities such as journal articles.

[25] http://labs.juiceanalytics.com/chartchooser/index.html
[26] https://extremepresentation.typepad.com/blog/2008/06/visualization-taxonomies.html

Visualisation Design

Beauty is in the eye of the beholder, and there are no formulas or algorithms to ensure perfect visualisations. The social network Reddit has two groups dedicated to visualisations. Users members of the Data is Ugly[27] and Data is Beautiful[28] groups share images of visualisations they consider ugly or beautiful. These two groups sometimes share the same visualisations because of different interpretations of aesthetics in data. What is a beautiful visualisation to one person, is an abomination to somebody else. The aesthetics of data visualisation is for a significant part in the eye of the beholder. However, when viewing aesthetics from a practical perspective, we can define what this means with a simple heuristic.

Edward Tufte is an American statistician who is famous for his work on visualisation. Tufte introduced the concept of the data-ink ratio. In simple terms, this ratio expresses the relationship between the ink on the paper that tells a story and the total amount of ink on the paper. Tufte argues that this ratio should be as close to one as possible. In other words, we should not use any graphical elements that don't communicate any information, such as background images, superfluous lines and text.

Now that we are in the paperless era, we can use the data-pixel ratio as a generic measure for the aesthetics of visualisations. The principle is the same as in the analogue days. Remove any redundant information in your visualisation. Unnecessary lines, multiple colours or multiple narratives risk confusing the user of the report.

The data-ink ratio is not a mathematical concept that needs to be expressed in exact numbers. This ratio is a guideline for designers of visualisations to help them decide what to include and, more importantly, what to exclude from an image.

Figure 1.10 shows an example of maximising the data-ink ratio. The bar chart on the left has a meagre data-pixel ratio. The background image of a cat might be cute and possibly even related to the topic of the visualisation, but it only distracts from the message. Using colours to identify the variables is unnecessary because the labels are at the bottom of the graph. The legend is not very functional because it also duplicates the labels. Lastly, the lines around the bars have no function.

To improve this version, all unnecessary graphical elements have been removed. Assuming that the story of this graph is to compare variables, the columns have been ranked from large to small. If the narrative of this graph was to compare one or more of the variables with other variables, then groups of bars can be coloured to indicate the categories.

The basic rule of visually communicating data is to not 'pimp' your visualisations with unnecessary graphical elements or text that does not add to the story. When visualising data, austerity is best-practice.

[27] https://reddit.com/r/dataisugly/
[28] https://reddit.com/r/dataisbeautiful/

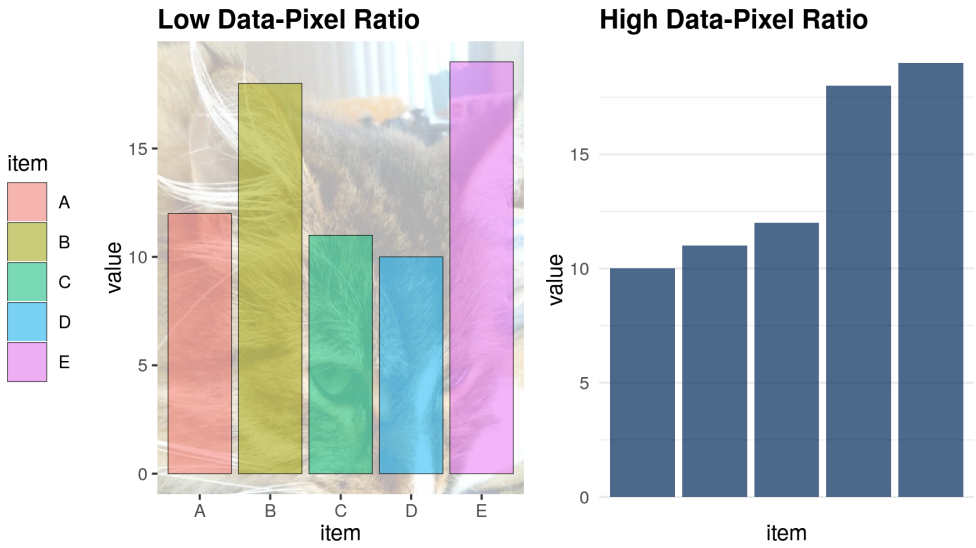


Figure 1.10: Examples of the data-pixel ratio.

Reports

Advertising executive Fred Barnard coined the now famous idiom that "a picture is worth (ten) thousand words" in 1927. While this might be the case, the complexity of data science in most cases requires text to explain the analysis.

To claim that a report needs to be written with clarity and precision in proper spelling and grammar almost seems redundant. The importance of readable reports implies that the essential language a data scientist needs to master is not Python or R but English, or another human language.

Writing a good data report enhances the reproducibility of the process by describing all the steps in the process. A report should also help to explain any complex analysis to the user to engender trust in the results.

The topic of writing useful business reports is too broad to do justice within the narrow scope of this book. For those people that need help with their writing, data science can also assist. There are many great online writing tools to support authors not only with spelling but also grammar and idiom. These advanced spelling and grammar checkers use advanced text analysis tools to detect more than spelling mistakes and can help fine-tune a text utilizing data science. As English is my second language, I rely heavily on the Grammarly software to ensure it is free of apparent issues. However, even grammar checking with machine learning is not a perfect replacement for a human being who understands the meaning of the text.

Best-Practice Data Science with R

The principles of data science discussed above are applied in the next three case studies. Each case study starts with a problem definition that explains the problem that needs to be solved. Starting

analysis with a problem statement is critical to ensure that our work is useful and able to deliver value. Each case study also includes a description of the reality from which the data was extracted. Understanding the context of abstract data minimises mistakes in interpretation.

To prepare for the first case study, please install the R software[29] and the RStudio[30] environment. You also need to download the GitHub[31] to obtain the data files an code examples.

The next chapter[32] introduces the R language and using RStudio.

[29] https://www.r-project.org/
[30] https://www.rstudio.com/
[31] https://github.com/pprevos/r4h2o
[32] https://leanpub.com/courses/leanpub/R4H2O/read/3

Introduction to the R Language

R is a programming language for statistical computing and visualisation. This language is developed and maintained through the R Foundation for Statistical Computing[33]. The R software is open source, which means that anyone can freely download, use, modify and share the software. The open source model relies on communities of developers that continuously improve the software.

Open source software is free. Not free as in free beer, but free as in freedom[34]. The people developing open source software also need to be paid, and most projects are not-for-profit organisations funded by organisations that use the software commercially. If your organisation uses R commercially, then I highly recommend considering supporting the R Foundation.

The R language is one of the most popular tools for analysing data. This language includes advanced mathematical capabilities, missing from general-purpose languages. This language also has extensive built-in visualisation capabilities. Furthermore, R can be integrated with many other data science software systems, such as *Power BI*, *Tableau*, *Mathematica*, *MATLAB* and do so on.

This session only gives a cursory overview of the R language with enough theory to solve the case study. This course is only a teaser to motivate water professionals to ditch their spreadsheets and start to write code. For a more systematic introduction to this language, I recommend following one of the many courses available on the internet or read a book. Two recommended sources to systematically learn the basics of the R language in detail are:

- DataCamp[35]: Free introduction courses and paid advanced courses.
- R for Dummies[36]: Introduction from the well-known *For Dummies* series.

Using R and RStudio

The best way is to use R in combination with an *Integrated Development Environment* (IDE). The most popular IDE for the R language is RStudio[37]. This software is also an open source project, with free and paid versions.

An integrated development environment is a software application with comprehensive functionality for developing software. An IDE typically consists of at least a source code editor, automation tools, and functionality to make writing and running code easier.

[33] https://www.r-project.org/foundation/
[34] https://www.gnu.org/philosophy/free-sw.html
[35] https://www.datacamp.com/
[36] http://rfordummies.com/
[37] https://rstudio.com/

Before you continue, make sure you have access to R and RStudio and have downloaded the course files from GitHub.

When you open RStudio for the first time, the window is divided into three panes, each with various tabs. The left pane is the console. The top right pane shows the system environment and the one below that shows a list of files and folders (Figure 2.1).

You can change the default fonts and colours in the *Tools > Global Options > Appearance* menu. Most developers prefer a dark theme with light text because it is more gentle on the eyes than a stark white background. You can also set default font size and magnification to your liking.

Open the appearance menu and change the settings to your personal preferences.

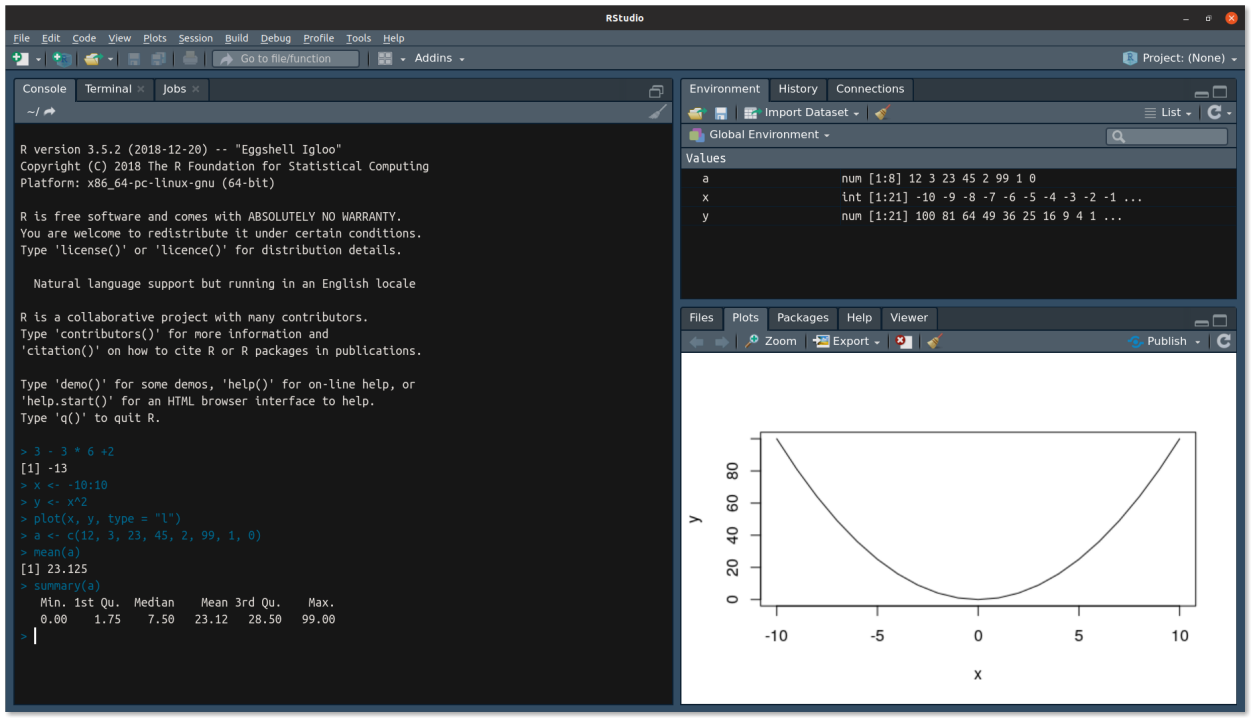


Figure 2.1: RStudio default screen layout

Basics of R

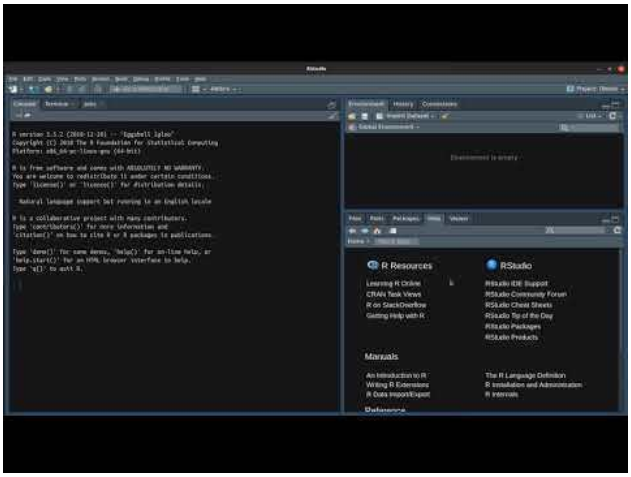
Now we are ready to write some code. Move your cursor to the console and type the code examples listed below. Don't copy and paste them because typing the code develops your muscle memory for the R syntax and you some of the experience the features of the text editor.

```
3 - 3 * 6 + 2

x <- -10:10
y <- x^2

plot(x, y, type = "l")

a <- c(12, 3, 23, 45, 2, 99, 1, 0)
mean(a)
```



View this Video at https://youtu.be/roTCgjxpMEg[38].

Introduction to RStudio

This code demonstrates some basic features of the language. The first line is a simple, arithmetic problem. After you hit enter, R displays the answer below the line.

The next two lines define the variables `x` and `y`. The values -10 to + 10 are assigned (`<-`) to variable `x`. The `y` variable is given the value of x^2.

The third part plots the variables `x` and `y` as a line, showing the parabola in the plot window. Without the `type ="l"` parameter, the plot consists of points.

The variable `a` is assigned a vector of eight numbers using the `c()` function. The `mean()` function shows the arithmetic mean of the vector `a`.

You should notice a few things when you start typing:

- When you hit enter, the result of the expressions without the `<-` symbol is shown in the console
- When you type plot and mean, R gives you suggestions on how to continue
- When typing brackets or quotation marks, RStudio includes the closing bracket or quotation mark
- The variables you declared (`x`, `y` and `a`) are shown in the Environment window

- The plot appears in a tab of the bottom-right window.

Now retype the plot command, but only type the first two letters and then hit the TAB key. R now gives you suggested functions that start with `pl`. You can use the cursor keys to select the plot function. You can continue this way, and R guides you through the function. This functionality is great for when you forget the specific syntax when writing code.

Another useful function of the console is to use the arrow keys to repeat or modify previous commands.

Now it is your turn to play with the basic syntax of R and functionality of RStudio.

Produce a plot of the function $y = -x^2 - 2x + 3$.

The formula for determining where the parabola intersects the x-axis is:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Use the quadratic formula in the R console. Where does this parabola intersect with the x-axis?

Now it is time to use these basic skills to the first case study[39].

RStudio scripts and projects

The console provides a running record of the actions taken by R. While this is great, using the console makes it hard to reconstruct what steps you have taken to get to your result. To create reproducible code, you need to write your code.

Create a new R script by going to *File > New File > R Script* or by hitting Control-Shift N.

Add the same code as above in the script.

A project is a set of files that relate to each other. RStudio projects divide your work into multiple contexts, each with their own working directory, workspace, history, and source documents. Every time you open a project file, it will be in the same state where you left it when you last closed the program. There are several ways to open a project:

[39] https://leanpub.com/courses/leanpub/R4H2O/read/3

- Using the Open Project command (available from both the Projects menu and the Projects toolbar) to browse for and select an existing project file (e.g. `r4h2o.Rproj`).
- Selecting a project from the list of most recently opened projects (also available from both the Projects menu and toolbar).
- Double-clicking on the project file within Windows Explorer, OSX Finder, or another file manager.

Open the project file for this course.

After you open this file, you see the relevant files in the bottom-left window. When you close the project after this session, all variables, the history of your commands and open files are stored for use in a later session.

This section only provides a concise introduction to the basics of the R language and RStudio. The remainder of the workshop introduces further concepts as needed by the case studies.

Case Study: Water Quality Regulations

The case study for this first session is about assessing compliance with water quality regulations. The data for this case study is a set of turbidity measurements for the Laanecoorie water network[40], situated just over 100 km North of Melbourne in Victoria, Australia. The plant extracts water from the Laanecoorie reservoir, situated on the Loddon River.

The water network is divided into four zones, each of which has a set of sample points installed at customer taps in the front of the house. Each of these sample points has a unique identifier that consists of three digits (090 for the Laanecoorie system), a letter to indicate the zone, and two digits to indicate the number of the sample point.

The laboratory service provider regularly samples these taps and tests the water for a range of parameters, including turbidity. All turbidity measurements are recorded for a specific sample point at a certain date. The data set is already cleaned and is ready for analysis.

The states of Australia each have their own water quality regulations. The state regulations all refer to the federal Australian Drinking Water Quality Guidelines[41].

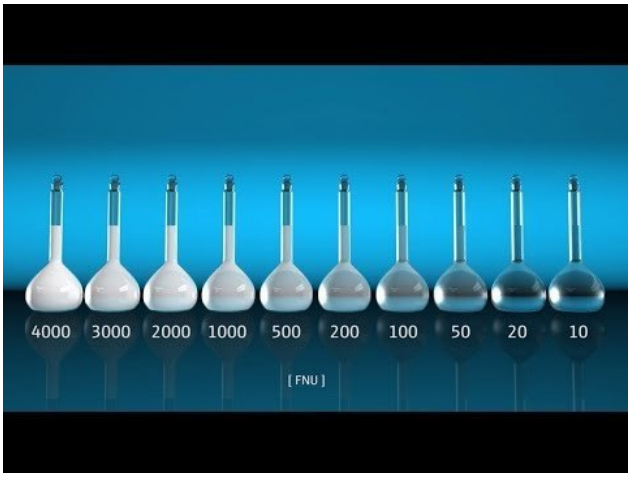
The Victorian regulations for water quality, the Safe Drinking Water Regulations[42] 2015, specify that "the 95th percentile of results for samples in any 12 months must be less than or equal to 5.0 Nephelometric Turbidity Units". The regulations also specify that each water quality zone needs to be sampled at least once per week.

In a separate guidance document[43], the Victorian regulator also specifies that the percentile for turbidity should be calculated with the 'Weibull Method'.

Turbidity

Turbidity is a measurement of the cloudiness of the water. In drinking water, the higher the turbidity level, the higher the risk that consumers develop gastrointestinal diseases. Particles in the water scatter light, which is used to measure turbidity with a nephelometer (from the Greek nephÃ©la, "cloud"). Turbidity is expressed in dimensionless Nephelometric Turbidity Units (NTU). The video below gives a detailed overview of how to measure turbidity in liquids.

[40] https://www.coliban.com.au/site/root/your_town/loddon/laanecoorie.html
[41] https://www.nhmrc.gov.au/about-us/publications/australian-drinking-water-guidelines
[42] https://www2.health.vic.gov.au/public-health/water/drinking-water-in-victoria/drinking-water-legislation
[43] https://www2.health.vic.gov.au/Api/downloadmedia/%7BA1F6D255-D5C7-4B7E-AAE5-8B7451EDE81A%7D



View this Video at https://www.youtube.com/watch?v=qz8xHQJw6qY[44].

Determination of Turbidity of water: Calibration and Measurement

Problem Statement

You are writing the annual report to the regulator about the Laanecoorie system. What was the 95th percentile of turbidity for each of the water zones in the system?

Methodology

Good data science needs to be valid and reliable. The validity and reliability of the measurements in this case study relates to the design, installation and maintenance of the instrument to measure turbidity.

The soundness of good data science also requires an appropriate methodology to be used to analyse the data. This case study has some specific requirements concerning how to analyse the data. The guidance document from the regulator raises two questions: What is the Weibull method? How do you implement this method in R?

The basic process to determine a percentile is a three-step process (McBride, 2005[45]):

1. Rank into ascending order ($X_1, X_2, \ldots, X_n$).
2. Determine the rank (r) of the required percentile.
3. The percentile is the value in position r. When the rank is not an integer, interpolate between two values X_{r-1} and X_{r+1}.

With 52 ranked weekly turbidity samples, the 95th percentile is between sample 49 and 50 (0.95×52). However, this method is only valid for normally-distributed samples. Statisticians have defined

[45] http://amzn.to/2k8shr8

several methods to determine percentiles. The difference between these methods is determining the rank r. Hyndman & Fan (1996[46]) give a detailed overview of nine methods of calculating percentiles or quantiles. These nine methods are incorporated into R, as discussed below.

This paper gives the Weibull method the less poetic name $\hat{Q}_6(p)$ because it is the sixth option in their list. Waloddi Weibull, a Swedish engineer famous for his statistical distribution, was one of the first to describe this method. The rank of a percentile p is given by:

$$r_{weibull} = p(n+1)$$

For a sample of 52 turbidity tests, the percentile thus lies between ranked result number 50 and 51. This method is suitable for highly skewed samples, as is often the case with water quality data.

Please note that there is no correct way to calculate percentiles. The most suitable method depends on the distribution of the population. In this case study, the method is prescribed by the regulator.

You have received 99 turbidity results from the laboratory. The first 94 are 0.1 NTU, and the last five are 5 NTU. What is the 95th percentile using the Weibull method?

Analysing the case study

The sections below explain how to analyse an example data set with turbidity data for compliance with the Victorian Safe Drinking Water Regulations. The data and the code is available in the GitHub[47] repository. Before we can start determining the relevant statistics, we need to load and explore the data.

The code is available in the `session2` folder in the `casestudy1.R` file. You can find the answers to the questions in the last section of the course. The best way to learn the material is to type all the examples and assignments in your file.

Create a new R file for this case study in RStudio.

Load the data

The data is stored in a CSV file, which the `read.csv` file can read. The text between quotation marks is the path to the file. The path is relative to the working folder, so in this case, we need to add the folder and the file name. Note that R uses the forward slash, common in Unix systems, and not the Windows backslash (\) to form a path.

[46] https://www.researchgate.net/publication/222105754_Sample_Quantiles_in_Statistical_Packages
[47] https://github.com/pprevos/r4h2o/

```
turbidity <- read.csv("session2/turbidity_laanecoorie.csv")
```

The turbidity data is now visible in the *Environment* tab. The turbidity data is a 'data frame', which is a tabular set of data with rows and columns, very much like a spreadsheet.

R can read many types of data. Some specialised extensions can be used to connect R to Excel spreadsheets, SQL databases, scrape websites, and many other sources. The `extract_data.R` file in the case study folder shows how the turbidity data was extracted from an SQL server.

Many organisations maintain spreadsheets with data as their single source of truth. If a spreadsheet is indeed your only solution to store data, you should stick to some simple rules to be able to easily use it in R or any other data science package:

- Use the top row as a header
- Don't use colours to indicate values
- Prevent using spaces in column names
- Don't add any calculations in the data tab

Following these guidelines, you can store your data in a clean way that makes analysing the results with R much more straightforward. The data in this case study complies with these guidelines and has the following fields:

- `Date_Sampled`: The sampling date.
- `Sample_No`: Reference number of the sample.
- `System`: name of the water system.
- `Zone`: The zone within the water system.
- `Sample_Point`: The reference number of eh sample point.
- `Result`: The result of the laboratory test.
- `Units`: The units of the result (NTU).

Inspect the data

The next step is to see what is in the data. When you type the name of the variable in the console, R displays the data up to the first 1000 rows. This method is not an effective method to view large sets because the data scrolls quickly across the screen. R has a series of functions to inspect data frames in more detail.

The `head` function only shows the first half dozen rows of the data, which prevents the screen from scrolling away. R also includes the `tail` function, that shows the last rows of a data frame.

The `names` function displays the names of the columns as a vector of character strings. You can also use this function to rename the variables in a data frame.

The `dim` function shows the number of rows and columns.

The `View` function (note the capital V) opens the data in a separate read-only window. This function is the most convenient way to inspect the data. You can also view the data this way by clicking on the variable name in the Environment tab. You cannot edit the data, but you can sort the information by column by clicking o the variable name.

```
head(turbidity)

names(turbidity)

dim(turbidity)

View(turbidity)
```

Use the `nrow` and `ncol` functions to determine the size of the data frame.

Lastly, the `str` function provides a succinct overview of the fields in the data set, including the data types. When executing this function on the turbidity data we see:

```
> str(turbidity)
'data.frame':    508 obs. of  7 variables:
 $ Date_Sampled: Factor w/ 127 levels "2017-01-05","2017-01-11",..: 124 124 124 107 \
92 92 92 89 78 78 ...
 $ Sample_No   : int  6075624 6075623 6075626 5923634 5797980 5605791 5605792 ...
 $ System      : Factor w/ 1 level "Laanecoorie": 1 1 1 1 1 1 1 1 1 1 ...
 $ Zone        : Factor w/ 4 levels "Bealiba","Dunolly",..: 2 3 1 1 3 1 4 1 3 2 ...
 $ Sample_Point: Factor w/ 24 levels "090A01","090A02",..: 13 2 21 21 3 21 15 24 1 7\
 ...
 $ Result      : num  0.1 0.2 0.1 0.1 0.2 0.3 0.2 0.2 0.2 0.2 ...
 $ Units       : Factor w/ 1 level "NTU": 1 1 1 1 1 1 1 1 1 1 ...
```

This table means that `turbidity` is a data frame with 508 observations (rows) and 7 variables (columns). The `Date_Sampled`, `System`, `Zone`, `Sample_Point` and `Units` are factors.

You can also obtain this information by clicking on the triangle next to the variable name in the Environment tab.

R converts most character strings to factors to save memory and to assist with analysis. Factors are beneficial in repetitive data. The levels of the factors are the unique values within the data. In this data, there is only one system, and there are four zones and 24 sample points.

The levels are numbered by default in alphabetical order. The `levels()` function displays the levels in a factor. This function can also change the labels of the levels.

The `Sample_No` is an integer variable, which makes sense because we expect most of these values to be unique for each observation.

The sample date is expressed in a factor because R sees it as a character string in the first instance. The dates are formatted following the ISO 8601[48] standard (YYYY-MM-DD). For example, 27 September 2012 is represented as 2012-09-27.

For R to understand which date comes before the next, we need to convert this variable. R has extensive functionality for working with dates, which is covered in Case Study 3.

Explore the data

To view any of the variables within a data frame, you need to add the column name after a `$`, e.g. `turbidity$Result`. When you execute this command, R shows a vector of the selected variable. You can use this vector in calculations, as explained below.

If you want to use only a subset of a vector, you can indicate the index number between square brackets. For example: `turbidity$Results[1:10]` shows the first ten results.

R has various ways to view or analyse a subset of the data. The most basic way is to add the number of the row and column between square brackets. For example, `turbidity[1:10, 4:5]` shows the first ten rows and the fourth and fifth variable. When there is no value in either the place for the rows or the columns, R shows all values.

```
turbidity[, 4:5]  ## Show all rows with column four and five
turbidity[1:10, ] ## Show all variables for the first ten rows
```

This syntax can also include the names of variables, e.g. `turbidity[1:10, c("Zone", "Result")]` shows the first ten rows of the one and the result.

In summary, you can subset a vector data frame by adding an index number between square brackets. For vectors, you add one number to indicate the element number. For a data frame, you use two numbers: `[rows, columns]`. When you omit either the row or column number, R shows all available values.

Please note that R is a mathematical language and the index numbers thus start at one. In generic programming languages, the index starts at zero. Besides numerical values, you can also add formulas as indices.

What is the result of the last sample taken in the turbidity? Hint, use the `nrow()` function/

You can also filter the data using conditions. If, for example, you like to see only the turbidity data for the Bealiba water quality zone, then you can use the following two methods:

[48] https://www.iso.org/iso-8601-date-and-time-format.html

```
turbidity[turbidity$Zone == "Bealiba", ]
subset(turbidity, Zone == "Bealiba")
```

The first method looks similar to what we discussed above. The row indicator now shows an equation. When you execute the line between brackets separately, you see a list of values that are either `TRUE` or `FALSE`. These values indicate whether the variable at that location meets the condition. For example, the following code results in a vector with the values TRUE and FALSE.

```
a <- 1:2
a == 1
```

The second method uses the `subset()` function, which is a bit more convenient than using square brackets. The first parameter in this function is the data frame, and the second parameter is the condition. Note that this method is tidier than the brackets method because we don't have to add the data frame name and `$` to the variables.

You can build elaborate conditionals by combining more than one condition. Some of the most common options are:

- `==`, `!=`: Equal to or not equal to.
- `<`, `>`, `>=`, `<=`: Inequality.
- `&`, `|`, AND, OR.

For example, `turbidity[turbidity$Zone == "Laanecoorie" & turbidity$Result > 1, ]` shows the samples in Laanecoorie larger than 1 NTU. Note that testing for equality requires two equal signs.

In the next case study, we dig deeper into manipulating and filtering data using the Tidyverse libraries.

How many turbidity results in Bealiba are lower than to 0.1 NTU?

Visualise the data

The fastest way to explore data is to visualise it. R has extensive built-in visualisation function, some of which we explore below. The R Graph Gallery[49] provides some guidance on the available methods.

Use the Chart Chooser or the R Graph Gallery to determine the best way to visualise the data.

[49] https://www.r-graph-gallery.com/

Given the requirements n the regulations, we need to visualise the distribution of the results for each zone. We only have a single variable, which leads us to a histogram.

The `hist()` function plots a histogram of a vector of integers or numerical values. The breaks option in this function defines the number of bars in the graph. The results of the turbidity tests have a maximum value of 1.5 NTU, so to get bars at 0.1 NTU, the number of breaks needs to be 15. The variable `b` in the code below calculates the size of the bars by dividing the maximum value by the desired resolution (Figure 2.2).

```
b <- max(turbidity$Result) / 0.1
hist(turbidity$Result, breaks = b, main = "Turbidity Results")
```

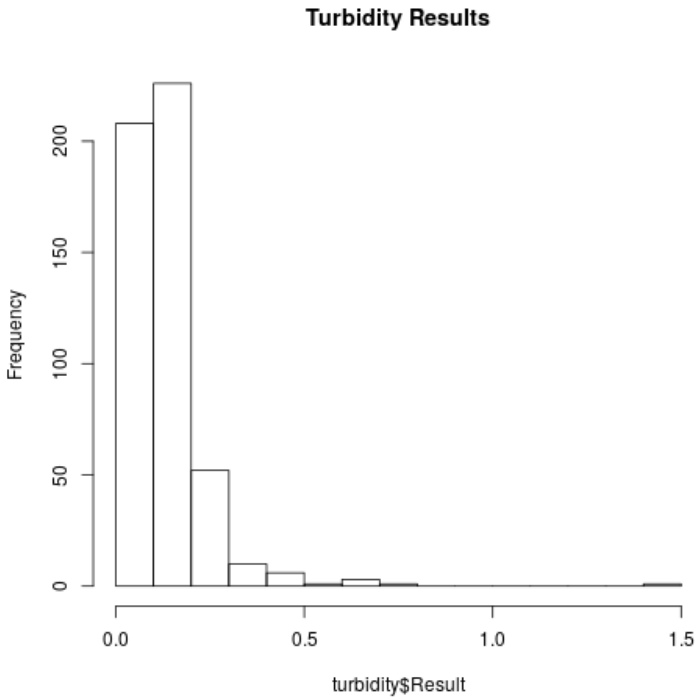


Figure 2.2: Histogram of turbidity results.

The regulations apply separately to each water quality zone, so we need to subset the data before plotting.

Plot the histogram of each of the Laanecoorie water quality zone.

Subsetting the data for each zone is tedious. One of the visualisations not listed on the *Chart Chooser* is the boxplot. This versatile visualisation summarises the distribution of numerical data (Figure 2.3).

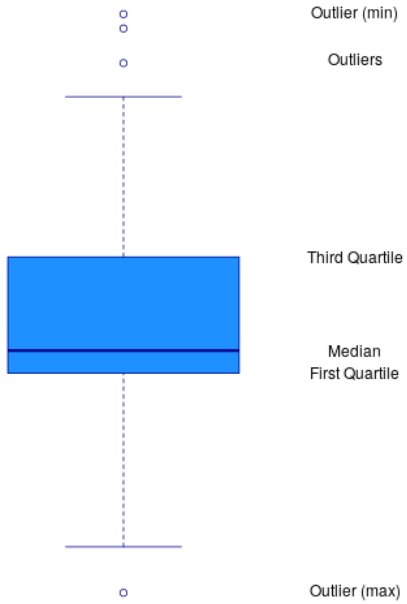


Figure 2.3: Boxplot anatomy.

- The line that divides the box indicates the median.
- The ends of the box shows the upper ($Q3$) and lower ($Q1$) quartiles. The difference between quartiles 1 and 3 is the interquartile range (IQR)
- The lines show $Q3 - 1.5 \times IQR$ to $Q1 + 1.5 \times IQR$ (the highest and lowest value, excluding outliers).
- Dots beyond the lines shows outliers.

The boxplot function includes a convenient way to group the results by a factor variable. To achieve this, use the tilde $\sim$ symbol to indicate the variable that is analysed and the variable by which it is grouped, as shown below. Because the data option indicates the data frame, we don't have to use the `$` indicator. The `main` and `ylab` options add text to the plot, as shown below and in figure 2.4.

```
boxplot(Result ~ Zone, data = turbidity, col = "lightblue",
        main = "Turbidity Results Laanecoorie water system",
        ylab = "Turbidity (NTU)")
```

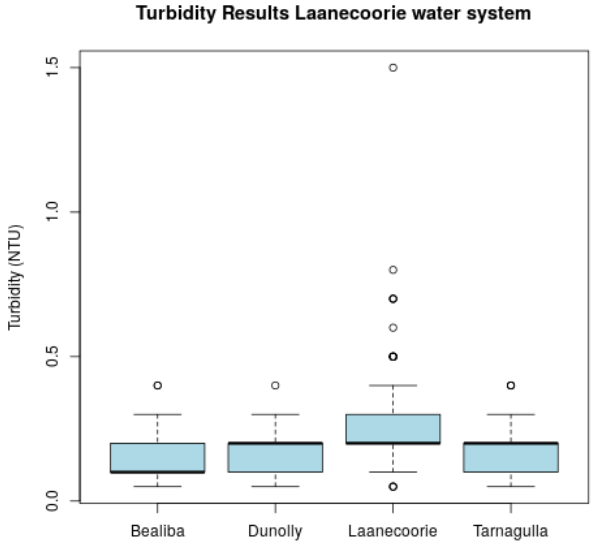


Figure 2.4: Distribution of turbidity results.

Each of these visualisation functions has extensive options to change the plot, which are outside the scope of this course. In the next two case studies, we explore the powerful visualisation functionality of the Tidyverse extension to the R language.

Analyse the data

While a plot provides a quick overview of the data, we need to numerically analyse the results to find the values to report to the regulator. R has extensive functionality to analyse data. We already saw the `mean()` function that calculates the arithmetic mean of a vector.

What is the mean turbidity value for the samples in Bealiba?

Another helpful function is `summary()` which shows six basic statistics: the minimum value, the first quartile, median, mean, third quartile and the maximum.

What is the third quartile for the turbidity of sample point 090A01?

The `quantile()` function calculates the percentiles of a vector of numbers. The default setting gives five values, similar to the `summary()` function. The quantile function can also take a vector of one or more probabilities to calculate different outcomes, for example `quantile(turbidity$Result, c(0.50, 0.95))` results in:

```
50% 95%
0.2 0.3
```

The regulator has specified that we need to calculate the 95th percentile with the Weibull method. The quantile function has access to all nine formulas described by Hyndman & Fan (1996). As we saw above, the Weibull method is the sixth option, which we can pass as a parameter.

```
quantile(turbidity$Result, 0.95, method = 6)
```

In this particular case, the results are not very skewed, so all methods give the same result.

One last function to review is a more convenient way to analyse subsets of the data. The `aggregate()` function splits the data into subsets, computes summary statistics for each, and returns the result in a data frame. For example, to determine the maximum turbidity value for each water quality zone, we use:

```
aggregate(turbidity$Result, list(turbidity$Zone), max)
```

The first argument in this function is the data vector, and the second argument is a list of the grouping variables. In this case, we only have one, but it can be more. The function that is applied to the groups is the third parameter, followed by this function's parameters.

We now have all the knowledge to answer the original question.

Determine the 95thpercentile using the Weibull method for all water quality zones in Laanecoorie.

Quiz: Water Quality Regulations

Now it is time to complete the first quiz. After you complete the quiz, we continue with cleaning and visualising data using the Tidyverse and a case study[50] about customer perception.

Take this quiz online[51]

[50] https://leanpub.com/courses/leanpub/R4H2O/read/4
[51] http://leanpub.com/courses/leanpub/R4H2O/quizzes/casestudy1?preview=true

Welcome to the Tidyverse

One of the most exciting aspects of the R language is that developers can write extensions, the so-called packages or libraries. R has a large community of users who develop code and make it freely available to other users in the form of packages.

Thousands of specialised packages are available that undertake a vast range of tasks. You can, for example, use R as a GIS and analyse spatial data. Other packages help you to access data from various sources, such as SQL databases. Many R extensions undertake tasks such as machine learning.

The majority of R packages are available on CRAN[52], the *Comprehensive R Archive Network*. You can install packages in R with the `install.packages` function. Within RStudio you can install packages in the *Tools* menu.

Before you can start using a library, you need to initiate it with the `library()` command.

Packages for water management

The CRAN library contains many packages with functions to analyse water. This workshop does not cover any of these packages. The list below gives some examples:

- baytrends[53]: Long Term Water Quality Trend Analysis.
- biotic[54]: Calculation of Freshwater Biotic Indices.
- CityWaterBalance[55]: Track Flows of Water Through an Urban System.
- driftR[56]: Drift Correcting Water Quality Data.
- EmiStatR[57]: Emissions and Statistics in R for Wastewater and Pollutants in Combined Sewer Systems.

The Tidyverse

One of the most popular series of packages is the Tidyverse[58], developed by Kiwi R guru Hadley Wickham and many others.

[52] https://cran.r-project.org/
[53] https://cran.r-project.org/web/packages/baytrends/index.html
[54] https://cran.r-project.org/web/packages/biotic/index.html
[55] https://cran.r-project.org/web/packages/CityWaterBalance/index.html
[56] https://cran.r-project.org/web/packages/driftR/index.html
[57] https://cran.r-project.org/web/packages/EmiStatR/index.html
[58] https://www.tidyverse.org/

The tidyverse packages provide additional functionality to extract, transform, visualise and analyse data. The additional functionality provided by these packages is easier to use than the base R code.

This case study focuses on cleaning and visualising customer data. The next case study uses Tidyverse to analyse digital metering data.

Install the Tidyverse collection of packages using `install.packages(tidyverse)`. When completed, initiate it with `library(tidyverse)`.

Installing the complete Tidyverse can take a little while, depending on your computer and the operating system. If you have problems installing, make sure that you are connected to the internet, and that your firewall or proxy don't block cloud.r-project.org[59].

When you load the Tidyverse, the following packages are loaded by default when you start the Tidyverse:

- ggplot2[60]: Visualise data.
- tibble[61]: Replacement for data frames.
- dplyr[62]: Data manipulation.
- tidyr[63]: Data transformation.
- purrr[64]: Functional programming.
- stringr[65]: Manipulate text.
- readr[66]: Read and write CSV files.
- forcats[67]: Working with factor variables.

Some data scientists prefer not to load the complete set and choose to load each package separately to spare computer memory. This course does not discuss the purrr, stringr or forcats libraries.

The startup message also shows some warnings about conflicts with some of the base functionality, which we can ignore. Many other packages are available that follow the principles of the Tidyverse.

Packages in the tidyverse change frequently. You can see if updates are available, and optionally install them, by running `tidyverse_update()`. You can also update all avaiilable packages in the *Tools > Check for Package Updates*.

The next case study[68] looks at data collected from tap water consumers in the United States and introduces the Tidyverse principles using this data.

[59] https://cloud.r-project.org/
[60] https://ggplot2.tidyverse.org/
[61] https://tibble.tidyverse.org/
[62] https://dplyr.tidyverse.org/
[63] https://tidyr.tidyverse.org/
[64] https://purrr.tidyverse.org/
[65] https://stringr.tidyverse.org/
[66] https://readr.tidyverse.org/
[67] https://forcats.tidyverse.org/
[68] https://leanpub.com/courses/leanpub/R4H2O/read/5

Case Study 2: Understanding Customer Perception

This second case study investigates the perception of consumers of tap water in the United States. This research formed part of a dissertation on water utility marketing[69] by Peter Prevos.

This research included a survey of American tap water consumers to measure their perception of tap water services. The survey included ten questions to measure consumers' involvement with tap water services.

Consumer Involvement

Consumer involvement is an important metric in marketing to describe the relevance a product or service has in somebody's life. People who own a car will most likely be highly involved with purchasing and owing the car due to the large amount of money involved and the social role it plays in developing their public self. Consumers will most likely have a much lower level of involvement with the instant coffee they drink than with the clothes they wear. More formally, consumer involvement can be defined as a person's perceived relevance of the object based on inherent needs, values, and interests.

From a managerial point of view, involvement is important because it is causally related to willingness to pay and perceptions of quality. Consumers with a higher level of involvement are willing to pay more for a service and have a more favourable perception of quality.

Understanding involvement in the context of urban water supply is also important because sustainably managing water as a common pool resource requires the active involvement of all users. The level of consumer involvement depends on a complex array of factors, which are related to psychology, situational factors and the marketing mix of the service provider. The lowest level of involvement is considered a state of inertia which occurs when people habitually purchase a product without comparing alternatives.

The highest possible level of involvement are the cult products where customers are fully devoted to a particular product or brand. Commercial organisations use this knowledge to their advantage by maximising the level of consumer involvement through branding and advertising. This strategy is used effectively by the bottled water industry. Manufacturers focus on enhancing the emotional aspects of their product rather than on enhancing the cognitive aspects. Water utilities tend to use a reversed strategy and emphasise the cognitive aspects of tap water, the pipes, plants and pumps, rather than trying to create an emotional relationship with their consumers.

[69] http://hdl.handle.net/1959.9/561679

Problem Statement

The fact that water is essential to life suggests that consumers of tap water have a high level of involvement with the service. Contrary to this common-sense intuition, practitioner experience and literature state that tap water is a low-involvement service. However, the level of consumer involvement with tap water services[70]has until now, not been empirically verified.

Using the data from the American tap water consumers, determine the level of consumer involvement.

Methodology

A commercial survey service provider recruited the respondents, who were paid for their participation. The questionnaire consisted of four pages[71], which respondents accessed through a website. The first page introduced the research and asked respondents to provide their consent to participate. Respondents who did not provide consent were exited from the survey. Due to the broad geographical spread of potential respondents on the American survey panel, respondents were required to also complete two screening questions. The first question related to their place of residence and the second question asked whether they had tap water at home. Only customers located in Los Angeles, Denver or Boston and those with tap water connections continued to the next page of the questionnaire. Other respondents were excluded from the survey.

The second page consisted questions about the level of involvement respondents have with tap water. These questions use the Personal Involvement Inventory developed by Judity Zaichkowsky (1994[72]). The involvement items close with an open text item asking customers: "If you have any additional comments about your views on tap water, please enter them below".

The Personal Involvement Inventory consists of two dimensions: cognitive involvement (importance, relevance, meaning, value and need) and affective involvement
(involvement, fascination, appeal, excitement and interest).

The involvement part of the survey uses a semantic differential scale. This is a method where respondents choose an answer between two antonyms (figure 3). This type of survey measures the meaning that people attach to a concept, such as a product or service. The items were presented in a random order to each respondent. The words on the right indicates a high level of involvement. Fiver questions have a reversed polarity, which means that the left side indicates a high level of involvement (Figure 3.1).

[70]https://www.researchgate.net/publication/326533830_We_Care_About_Water_Even_If_You_Don%27t_Water_As_a_Low_Involvement_Service
[71]resources/session3/customer_survey.pdf
[72]https://www.sfu.ca/~zaichkow/JA%252094.pdf

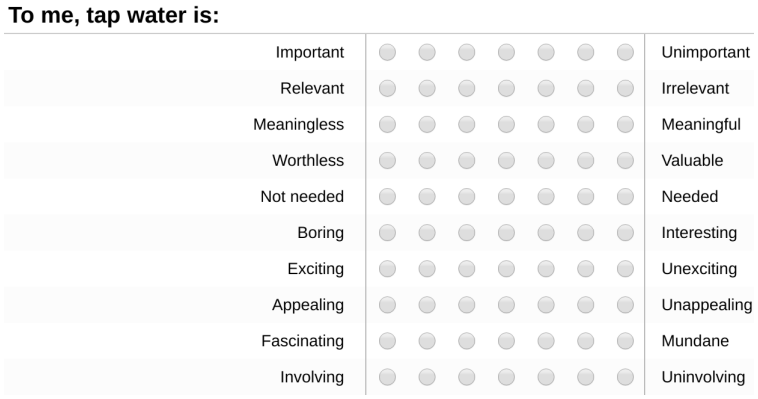


Figure 3.1: Personal Involvement Inventory questionnaire

The last page started with two items related to the customer's relationship with their service provider. Customers were asked to indicate whether they struggle to pay their water bills when they fall due, using a seven-point Likert scale from "Strongly Disagree" to "Strongly Agree".

The second question asked customers to indicate the frequency at which they contact their utility for support also using a seven-point Likert scale:

- Never
- Less than Once a Month
- Once a Month
- 2-3 Times a Month
- Once a Week
- 2-3 Times a Week
- Daily

One of the problems with using paid survey subjects is that they are motivated to submit a lot of responses, without having much regard for the content. American respondents were therefore also subjected to an attention filter: "If you live in the U.S. select Strongly Agree". The survey was only sent to people within the United States, so everybody should respond equally. Any respondent not answering "Strongly Agree" is excluded from the survey. This approach was used to remove inattentive respondents and assure the accuracy of the results.

The survey closed with eighteen service quality questions, which were measured using a seven-point Likert scale from "Strongly Disagree" to "Strongly Agree". The items were presented in random order. The final item of the questionnaire consisted of an open question which invited customers to provide additional comments about their tap water supplier.

If you are interested reading more about customer experience in water utilities, then you can read *Customer Experience Management for Water Utilities* by Peter Prevos, available from IWA Publishing[73].

[73] https://www.iwapublishing.com/books/9781780408668/customer-experience-management-water-utilities-marketing-urban-water-supply

Analysing the Case Study

The `Customer_Perception_USA.csv` file provided in the `session3` folder is the raw data as exported from the Qualtrics[74] survey platform. This session explains how to clean and visualise the data using the Tidyverse functionalities. The code in this section is available in the `casestudy2.R` file. You start the analyses by loading the Tidyverse packages using `library(tidyverse)`.

Load the data

The readr package of the Tidyverse has an alternative function to read and write CSV files. This function is faster than the base R version. This function looks almost the same, except for the underscore.

```
rawdata <- read_csv("session2/Customer_Perception_USA.csv")
```

We use the `rawdata` variable because we want to keep this data intact as we process it in case we need to use it again.

View the data to become familiar with the content and structure.

In Tidyverse, rectangular data is not a data frame, as in base R code, but a tibble. This odd term must be a an expression of the way Hadley Wickham says table in his New Zealand accent.

The first 19 columns contain metadata about the data collections, such as a unique response ID, IP addresses, start and end times and so on. The next 35 columns contain the actual data, which is discussed below. The next two columns contain the latitude and longitude of the respondent, based on their IP address. The last field indicates information of the accuracy of the location.

View the data in the console.

A tibble is exactly like a data frame, but with some extra functionality. When you ask to display it in the console, the text does not scroll away like in the standard version.

How many rows and columns of data does this data have?

[74] https://qualtrics.com/

Clean the data

Looking at the data, we also see that the first two rows contain header information. A tidy data set should only have one header row. Because of the double headers, R thinks that all columns are text.

We need to remove the first row and any irrelevant columns and re-assess the data types to create a clean table. The first line of code creates the new `customers` variable by removing the first line of the raw data. It is good practice to keep raw data and processed data in separate variables.

The `type_convert` function re-assesses the data to ensure it has the correct types. Using the `str` function, we can see that most columns are now numerical values, which is what we want them to be.

```
customers <- rawdata[-1, ]
customers <- type_convert(customers)
```

The next step is to remove any respondents that either:

- Failed the attention filter
- Did not consent
- Does not have tap water
- Does not live in one of the three nominated cities
- Quit the survey before completion

The Qualtrics survey software stores this information in the `term` field.

To summarise the content of this field we can use the `table()` function. This function creates a table of all the elements in a vector and gives the number of occurrences.

```
table(rawdata$term)
```

You will notice that the total number of items in the table does not match the number of rows. When you view the content of this field you will see a lot of entries with `NA` in them. These are empty values (Not Available). R uses this special code to be able to better manage missing values. More about missing values further below.

After reviewing the data we can conclude that we only want those rows of data that have an `NA` value in the `term` field.

In the dplyr package of Tidyverse, the `filter()` conditionally chooses rows of a data frame. using `filter(customers, term == "attention")` results in a data frame with only those entries that failed the attention filter. In our case we want all values with NA. To find these we need to use a special function, as shown below.

One way of cleaning the data would be:

```
customers <- filter(customers, is.na(term))
```

The Qualtrics data contains a lot of meta data that we don't need for further analysis. The first 19 columns contain information about when the survey was taken and so on and the last two columns are irrelevant. The next step is to filter the data so we only use columns 20 to 56.

How would you remove the unnecessary columns using base R code? Hint: Negative values removes columns.

In the dplyr package, the `select()` function that works just like the filter function, but for columns.

```
customers <- select(customers, 20:56)
```

We are close to a clean data set. The first column has the `city` variable, which at the moment is just the integer 1, 2, or 3. These numbers correspond to the order in the drop-down box in the survey. The options were:

1. Los Angeles
2. Denver
3. Boston.

First we create a new tibble to link the numbers with towns, which is then joined to the main data.

The `left_join` function finds the matching fields in the two sets and then merges the sets together. This function keeps all the values in the left data set. The Tidyverse has several other join functions[75] that match values in a different way. You can also define the fields on which you want to match the data frames.

```
cities <- tibble(city = 1:3,
                 city_name = c("Los Angeles", "Denver", "Boston"))
customers <- left_join(customers, cities)
```

When reviewing the code to clean the data we can see some repetition because we change the `customers` variable several times in a row. In a spreadsheet, these steps are often merged in one complex formula, like this:

```
left_join(select(filter(type_convert(rawdata[-1, ]), is.na(term)), 20:56), cities)
```

While the nested approach takes less space, it is not as easy to understand because you have to read from the inside out instead of from left to right.

The Tidyverse uses a pipe (`%>%`) to simplify this process. A pipe transports the output of one process to the input of the next process. The code used to clean the customer data is now written like this:

[75] https://dplyr.tidyverse.org/reference/join.html

```
customers <- rawdata[-1, ] %>%
    type_convert() %>%
    filter(is.na(term)) %>%
    select(20:56) %>%
    left_join(cities)
```

All other examples in this course will use this method of piping results to the next function.

We have cleaned the data for the whole survey, but before we continue we need to take one more step because we are for now only interested in the Personal Involvement Index (PII).

```
pii <- select(customers, City = city_name, starts_with("p", ignore.case = FALSE))
pii[, c(2, 3, 8, 9, 10, 11)] <- 8 - pii[, c(2, 3, 8, 9, 10, 11)]
write_csv(pii, "session2/personal_involvement_index.csv")
```

This code selects the `city_name` column and renames it to City. It also selects all columns that start with a lowercase p.

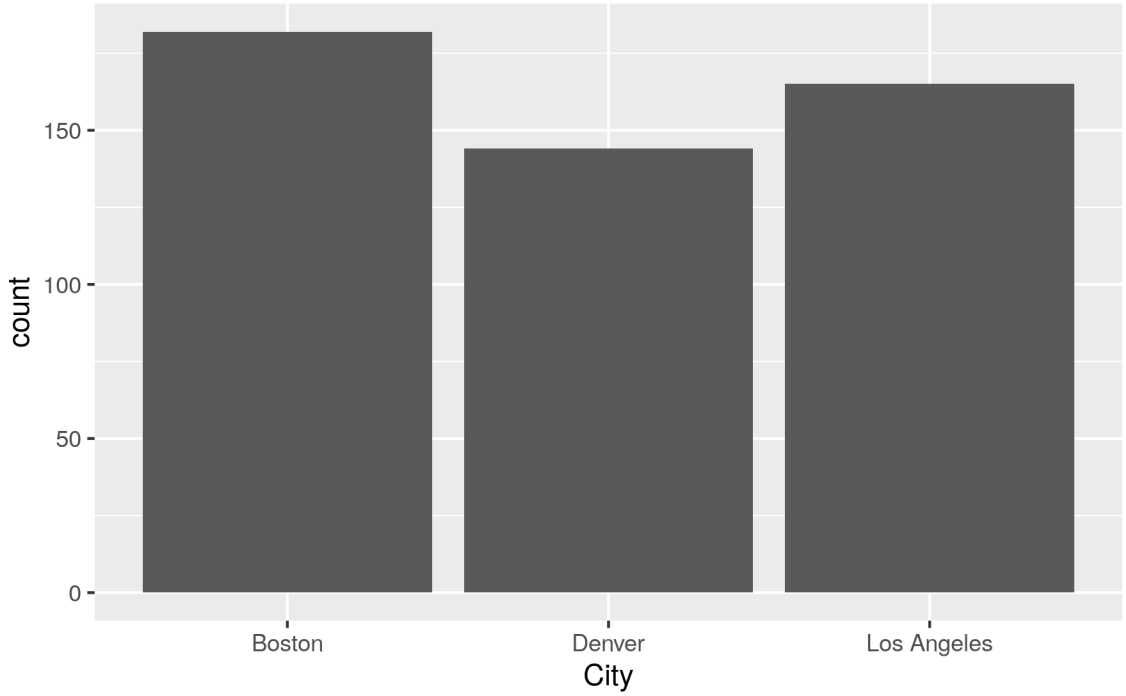
The second line reverses the value of

Visualise the data

The Tidyverse set of packages contains `ggplot`, one of the most powerful visualisation tools. This package implements a layered approach to visualising data, which helps building complex graphics. This session introduces the basic functionality of ggplot.

The basic ggplot function starts with the name of the data set and then a range of aesthetics. The aesthetics consist of the fields that will be used to visualise the data. The second part tells ggplot which geom to use. A geom is a type of visualisation. The ggplot library has many possible geoms. The example below shows a simple bar plot for the number of respondents from each city.

```
ggplot(pii, aes(City)) + geom_bar()
```



Simple bar chart with ggplot

This function creates a simple greyscale plot because we only should add colour to a visualisation when it expresses data. You can force ggplot to use a colour by using `geom_bar( fill = "blue")`, or any other colour which you might fancy.

Add your favourite colour to the bar plot.

The ggplot library implements the principles of the Grammar of Graphics. This grammar provides a structured approach to defining data visualisations using four aspects:

- Data and the aesthetic mapping
- Geometric objects (lines, bars and so on)
- Scales
- Facets of the visualisation

We will discuss each of these step[by step to build a complete visualisation of the PII data.

The ggplot function always takes a data frame or tibble as its first option. In base-R graphics, the data is always a vector, but ggplot uses rectangular data.

The aesthetic mapping defines which variables in the data frame are visualised. The mapping depends on the geometric objects. The aesthetics are at least one variable, for example

Assignment

Answers to the questions

Introduction to the R Language

Produce a plot of the function $y = -x^2 - 2x + 3$.

To plot this function, we can use the same approach as in the example, with a minor enhancement.

```
x <- seq(-5, 3, .1)
y <- -x^2 - 2 * x + 3
plot(x, y, type = "l")
```

This code uses the `seq()` function to create a smoother line than the colon (`-5:3`), which increments by 1. This function creates a vector from -5 to 1 with steps of 0.1.

The formula for determining where the parabola intersects with the x-axis is:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

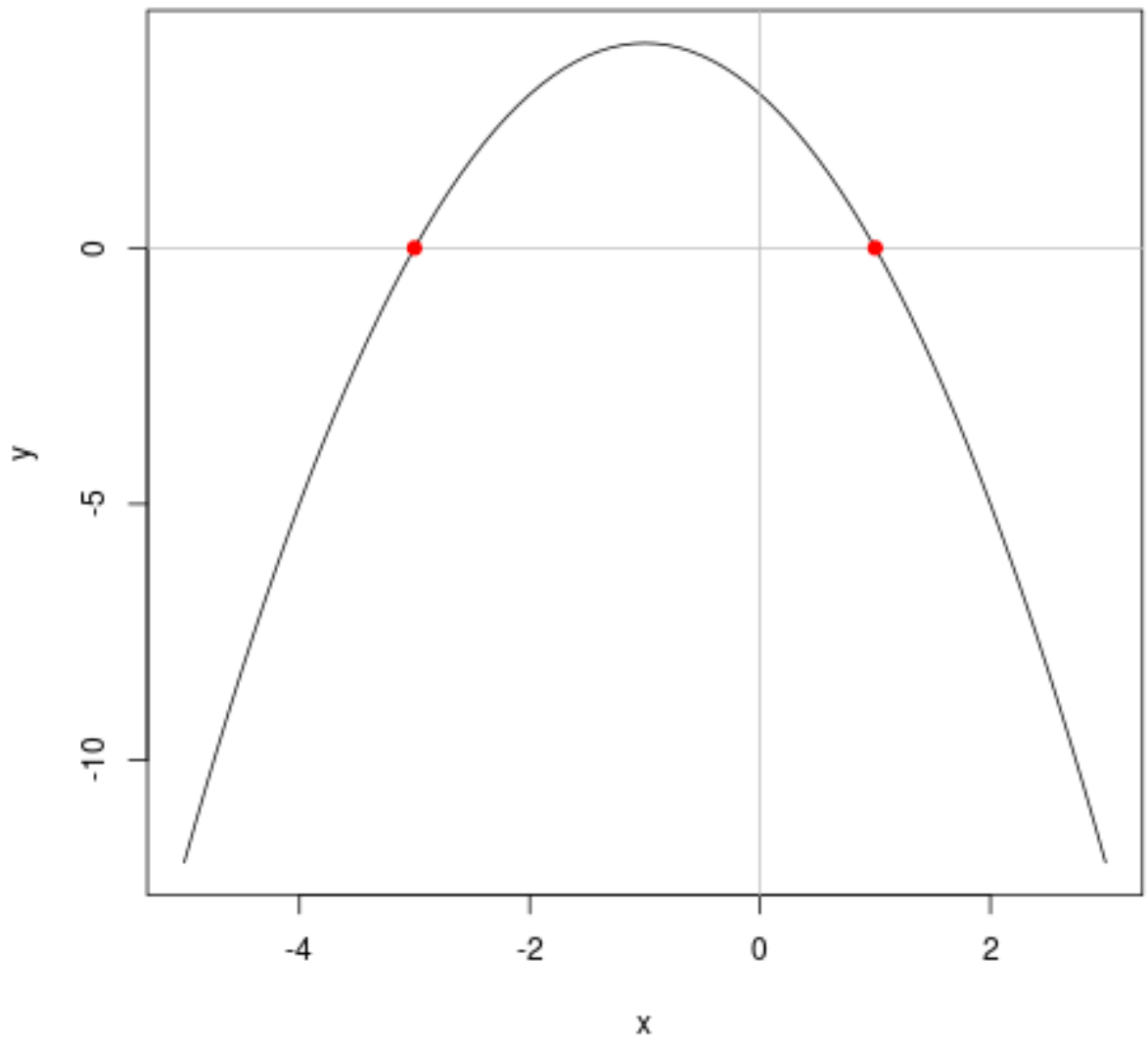
Use the quadratic formula in the R console. Where does this parabola intersect with the x-axis?

```
a <- -1
b <- -2
c <- 3

x1 <- (-b + sqrt(b^2 - 4 * a * c)) / (2 * a)
x2 <- (-b - sqrt(b^2 - 4 * a * c)) / (2 * a)

abline(h = 0, col = "grey")
abline(v = 0, col = "grey")
points(c(x1, x2), c(0, 0), col = "red", pch = 19)
```

We can enhance the basic plot to visualise the solution. The `abline()` function adds a horizontal and vertical grey line to indicate the axes. The `points()` function adds red points at the calculated intersects.



Parabola visualisation

Case Study: Water Quality Regulations

You have 99 turbidity results. The first 94 are 0.1 NTU and the last five are 5 NTU. What is the 95th percentile using the Weibull method?

Answer without using any code:

1. Rank the results in ascending order: $0.1, 0.1, \ldots, 5, 5, 5$.
2. Determine the percentile rank: $0.95 \times (99 + 1) = 95$.
3. The 95th percentile is the 95th result, which is 5 NTU.

We can also answer this question using R code:

```
results <- c(rep(0.1, 94), rep(5, 5))
rank <- 0.95 * (length(results) + 1)
rank_frac <- (r - floor(rank))
(1 - rank_frac) * results[floor(rank)] + rank_frac * results[floor(rank) + 1]
```

The first line create the results. The `rep()` function repeats a variable, in this case 94 times and 5 times. The two vectors are concatenated in one vector, suing the `c()` function.

The second line determines the rank of the 95th percentile in accordance with the Weibull method.

The last line interpolates between the . If the rank is an integer, than that value is used because the fraction is 0. The `floor()` function removes the decimals from a number.

Use the `nrow` and `ncol` functions to determine the size of the data frame.

The nrow and ncol function list the number of rows and columns for a data frame. The result sis a single number. The dim function shows both results in a vector of two numbers.

```
nrow(turbidity)
ncol(turbidity)
dim(turbidity)
```

What is the result of the last sample taken in the turbidity? Hint, use the `nrow()` function.

To find the last element of the data frame, use the `nrow()` function within square brackets.

```
turbidity$Results[nrow(turbidity)]
```

How many turbidity results in Bealiba are lower than to 0.5 NTU?

```
subset(turbidity, Results < 0.5 & Zone == "Bealiba")
```

Plot the histogram of each of the Laanecoorie water quality zone.

To plot a part of the data, we need to first create a subset.

```
l <- subset(turbidity, Zone = "Laanecoorie")
b <- max(l$Result) / 0.1
hist(l$Result, breaks = b)
```

It can be tedious to have to repeat this several times for the same data. A more advanced method is to use a loop. R can also display more than one plot on one screen using the `par()` function. This function modifies various aspects of the plot screen. The `mfrow` option defines how the screen is split. In this case, the screen is divided in two by two plots.

The `for` function lets you loop through a vector, in this case the unique values of the water quality zone. The variable `z` is assigned each of the values of the water quality zones, which are then plotted as above.

```
par(mfrow = c(2, 2))
for (z in unique(turbidity$Zone)) {
    l <- subset(turbidity, Zone = z)
        b <- max(l$Result) / 0.1
        hist(l$Result, breaks = b, main = z)
}
```

What is the mean turbidity value for the samples in Bealiba?

```r
mean(turbidity$Result[turbidity$Zone == "Bealiba"])
```

What is the third quartile for the turbidity of sample point 090A01?

```r
summary(turbidity$Result[turbidity$Sample_Point == "090A01"])
```

Determine the 95thpercentile using the Weibull method for all water quality zones in Laanecoorie.

```r
quantile(turbidity$Result, 0.95, method = 6) ## Weibull method
```

Quiz 1: Water Quality Regulations

Case Study: Customer Perception
Quiz: Customer Perception
Case Study: Smart Meters
Quiz: Smart Meters