Vrije Universiteit Brussel

FACULTEIT WETENSCHAPPEN EN BIO-INGENIEURSWETENSCHAPPEN
Departement Computerwetenschappen
Web & Information Systems Engineering Laboratory

Guided Data Visualisation for the MindXpres Presentation Tool

Proefschrift ingediend met het oog op het behalen van de titel Master of Science in Applied Sciences and Engineering: Computer Science, door:

Jasper Debie

Promotor: Prof. Dr. Beat Signer
Begeleider: Reinout Roels

AUGUSTUS 2014

Vrije Universiteit Brussel

FACULTY OF SCIENCE AND BIO-ENGINEERING SCIENCES
Department of Computer Science
Web & Information Systems Engineering Laboratory

Guided Data Visualisation for the MindXpres Presentation Tool

Graduation thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Applied Sciences and Engineering: Computer Science, by:

Jasper Debie

Promoter: Prof. Dr. Beat Signer
Advisor: Reinout Roels

AUGUST 2014

Samenvatting

We leven in een wereld overspoeld met data. Van kleine tot grote bedrijven die data verzamelen over hun klanten voor winstgevende doeleinden, families die huisprijzen en specificaties opzoeken voor het kopen van een nieuw huis, of studenten die feiten en cijfers over universiteiten afschuimen om er mogelijks te studeren, mensen gebruiken data voor bijna al de aspecten van hun leven. Niettemin moeten we onszelf afvragen, worden we ook overspoeld met betekenisvol inzicht in de data? Data wordt regelmatig bewaard als ongeorganiseerde verzamelingen van cijfers en letters die voor de mens onoverzichtelijk zijn en de inzicht in de data belemmert. Door de data beter te organiseren en te voorzien van een begripvolle representatie is het mogelijk om deze data te transformeren naar iets verstaanbaar en makkelijker te onderzoeken door de mens. Het proces van data transformeren naar een meer bruikbare representaties, zoals grafische visualisatie, is één van de grootste topics die behandeld wordt in het onderzoeksveld informatie visualisatie.

Onderzoek in informatie visualisatie is belangrijk door het feit dat gezichtsvermogen één van de meest dominantste en sterkste zintuig is waarmee we de wereld waarnemen. Door het creëren van data visualisaties kunnen we patronen, uitschieters en trends eenvoudiger waarnemen. Samen met de dichte relatie tussen ons gezichtsvermogen en cognitief vermogen kunnen we sneller kennis opdoen uit de data. Het is dan ook interessant om visualisaties van de data te gebruiken tijdens een presentatie. Het gebruik van visualisaties helpt het publiek om de gepresenteerde data sneller en gemakkelijker te begrijpen, wat ook kan bijdragen tot de kwaliteit van de presentatie. We moeten echter rekening houden met het feit dat ons gezichtsvermogen zijn eigen regels heeft. We kunnen in bepaalde situaties gemakkelijk patronen, uitschieters en trends zien, maar enkel als ze op de correcte manier worden gepresenteerd. Er is reeds veel onderzoek beschikbaar over de werking van ons visueel systeem in de vorm van richtlijnen en principes voor het visualiseren van data.

Na uitgebreid onderzoek kunnen we veilig concluderen dat huidige presentatie software de informatie visualisatie richtlijnen en principes voor het visualis-

eren van data niet volledig ondersteunen. Een groot aantal van de presentatie software voorziet de gebruiker met functionaliteit voor het creëren van eenvoudige en goed verstaanbare visualisaties, maar ook irriterende, verwarrende en misleidende visualisaties. De meeste gebruikers hebben echter niet altijd de intentie om ineffectieve visualisaties te creëren. Wat ze enkel missen is een correcte begeleiding door de presentatie software om visualisaties samen te stellen die de informatie visualisatie principes en richtlijnen volgen.

In deze thesis voeren we een uitgebreide literatuur studie uit om onszelf bekend te maken met de informatie visualisatie principes en richtlijnen. Door een kritische blik te werpen op hoe huidige presentatie tools de gebruiker begeleidt doorheen het creatie proces van visualisaties merkten we enkele gebreken. Door alternatieve aanpakken beter te bestuderen en oude vaste waarden los te laten konden we succesvol het data visualisatie proces herbekijken en opnieuw uitvinden. Als resultaat definieerden we een verzameling van vereisten dat visualisatie software moet ondersteunen tijdens het visualiseren van data.

Onze oplossing, genaamd Visualisation Picker, maakt gebruik van dit nieuw bedacht data visualisatie proces. Door de gebruiker te begeleiden naar de beste visualisatie type voor zijn of haar data, zorgen we ervoor dat al de patronen, uitschieters en trends kunnen worden waargenomen. Visualisation Picker creëert enkel data visualisaties die voldoen aan de informatie visualisatie principes en richtlijnen. Achteraf kunnen deze visualisaties gebruikt worden in MindXpres, een uitbreidbaar platform voor presentaties gebaseerd op webtechnologieën.

Met de Visualisation Picker voorzien we een solide uitbreidbaar prototype dat de gebruiker begeleid in het kiezen van een gepaste visualisatie type voor zijn of haar data en deze ook presenteert in een visualisatie dat de informatie visualisatie richtlijnen en principes volgt. Dit nieuw data visualisatie proces verhoogt de kwaliteit van de visualisaties in presentatie software. De Visualisation Picker bestaat ook uit een plug-in architectuur dat de gebruiker de mogelijkheid biedt om zelf nieuwe of bestaande visualisaties te implementeren. Hoewel we onze voorgestelde oplossing en vereisten als prototype hebben geïmplementeerd in MindXpres, is het ook mogelijk deze in de toekomst toe te passen op andere presentatie software.

Abstract

In the world we live in, we are flooded with data. Whether it is corporations of all sizes collecting data about their customers for profit purposes, families examining house prices and specification to buy a new home, or students skimming through facts and figures about universities as they plan to apply, people use data in almost all aspects of their lives. However, we also have to ask ourself, are we also flooded in meaningful insight? All this data is often stored as an unorganised sea of numbers and letters that are difficult to understand and gain insight from. Fortunately, by organising the data better and providing a meaningful representation it is possible to transform this data in a form that allows people to easily and quickly explore. The process of transforming data to a more useful form, such as graphical visualisations, is what the field information visualisation is all about.

Why should we even be interested in information visualisation? Vision is by far the dominant and most powerful of the five channels through which we sense the world. By creating visualisation from the data we can make patterns, outliers and trends visible and understandable. This makes it easier to process the data and helps us think about it while using the close relationship between vision and cognition. This is also the reason why it is interesting to use visualisation of the data during a presentation. It can help the audience to understand the data that is presented faster and more easily, which will also contribute to the quality of the presentation. However, the visual system through which we sense the world has its own rules. We can easily see patterns, trends and outliers presented in certain ways, but if they are presented in other ways they become invisible. There is a lot of research available about how our visual system works, and this knowledge can be translated into guidelines and principles for visualising data.

After some extensive research, we can safely state that current presentation tools' features for creating visualisation do not fully support these guidelines and principles defined in the field of information visualisation. Most of these presentation tools have a lot of functionality that allow the user to create

good and easily understandable visualisations, but also irritating, confusing and misleading visualisations. The creators of those bad visualisations do not often intend to come off in any of these ways, but they simply did not get the proper guidance in the presentation tool to create a visualisation that follows these principles and guidelines.

In this thesis we will do a literature study in information visualisation to become familiar with the principles and guidelines that are covered in this field. A closer look at how some of the current presentation tools guide the user in the process of creating data visualisations reveals a number of flaws. By broadening our view and letting go of the fixed values in creating visualisations in current presentation tools, we were successful in rethinking the process of visualising data and defined a set of minimal requirements a visualisation tool should support when providing functionality for creating data visualisation.

Our Visualisation Picker solution provides this new process of data visualisation creation for presentations. By guiding the user to the visualisation type that suits the data and message in the best possible way we make sure that no pattern, trend or outlier is missed. The visualisations created by the Visualisation Picker also meet the design guidelines and principles of the design aesthetics in information visualisation. These visualisation can be used in MindXpres, a web technology-based extensible platform for content-driven cross-media presentations. On top of that, the Visualisation Picker consist of a plug-in architecture that provides the possibility for the user to implement existing or new visualisation types.

With the Visualisation Picker we provide a solid extendible prototype that can guide users through the process of picking a visualisation type that suits the data and create visualisations that follow the guidelines and principles of information visualisation. The resulting improvements enhance the process of creating visualisations for presentations tools. While we implemented a first prototype for the MindXpres presentation tool the proposed solution and principles can also be applied to other presentation tools in the future.

Acknowledgements

I owe a debt of gratitude to many people for the help and guidance in this Master's thesis. First, I would like to thank my promoter Prof. Dr. Beat Signer, for giving me the opportunity to realise my thesis and for his guidance in the process. I also want to express the same gratitude towards my advisor Reinout Roels, for his continues guidance and support during my work. Many thanks for his efforts in monitoring and reviewing this thesis.

I am obliged to the members of the research group WISE, for the interesting and valuable information provided by them in their respective fields. I am very grateful for their guidance during the period of my work and studies.

Lastly, I would like to thank my parents, sisters and friends for their constant encouragement. Without your love, patience and support this thesis would not have been possible.

Contents

1 Introduction **2**

2 Problem Statement **8**
2.1 Information Visualisation 8
2.2 Terminology 10
2.2.1 Graphs, Charts and Visualisations 10
2.2.2 Pattern Perception 10
2.2.3 Graphical Integrity 10
2.2.4 Distortion 10
2.2.5 Cognitive Load 11
2.2.6 Categorical, Ordinal and Quantitative Data 11
2.3 The History of Information Visualisation 11
2.4 The Challenge of Information Visualisation 17
2.4.1 Choosing Visualisation Types 17
2.4.2 Graphical Integrity 22
2.4.3 Design Aesthestics 25
2.5 Overview 27

3 Principles of Graphing Data **28**
3.1 Types of Visualisations 29
3.1.1 Bar Chart 29
3.1.2 Dot Chart 30
3.1.3 Line Chart 30
3.1.4 Scatter Plot 31
3.1.5 Pie Chart 32
3.1.6 Histogram 32
3.1.7 Box Plot 33
3.2 Choosing Visualisation Types 33
3.2.1 Determine Your Message and Identify Your Data . . . 34
3.2.2 Table or Graph to Communicate the Message 39
3.2.3 Relationship that Best Supports the Message 39

3.2.4 Conclusion . . . 45
3.3 Graphical Integrity . . . 45
3.3.1 Reasons for Poor Graphical Integrity . . . 46
3.3.2 Tell the Difference . . . 48
3.3.3 Conclusion . . . 54
3.4 Design Aesthestics . . . 54
3.4.1 Visual Perception . . . 55
3.4.2 Principles of Grouping . . . 61
3.4.3 Data-Ink and Graphical Redesign . . . 63
3.4.4 Colours . . . 67
3.4.5 Legend . . . 70
3.4.6 Tick Marks and gridlines . . . 71
3.4.7 Tables . . . 72
3.4.8 Conclusion . . . 73

4 Graphing Data in Existing Presentation Tools 74
4.1 Common Presentation Tools . . . 75
4.1.1 Microsoft PowerPoint . . . 75
4.1.2 Apple Keynote . . . 76
4.1.3 LibreOffice Impress . . . 77
4.1.4 Beamer . . . 77
4.2 Visualisations in Presentation Tools . . . 78
4.2.1 Picking the Visualisation Type . . . 79
4.2.2 Testing Visualisation Type . . . 80
4.2.3 Misleading Visualisations . . . 87
4.3 Conclusion . . . 88

5 Towards Better Visualisations 90
5.1 Requirements for Visualising Data . . . 91
5.1.1 Understand the Data . . . 91
5.1.2 Guidance in Picking a Visualisation Type . . . 92
5.1.3 Suitable Design . . . 92
5.1.4 Extensibility . . . 93
5.2 The Ideal Visualisation Tool . . . 93
5.3 The MindXpres Presentation Tool . . . 96
5.3.1 MindXpres Architecture . . . 97
5.3.2 Plug-in Mechanism . . . 99
5.3.3 How to create plugins . . . 100
5.4 Conclusion . . . 101

6 The Visualisation Picker Tool 102

6.1 Goals . . . 103
6.1.1 Structure JSON Data Files . . . 103
6.1.2 GUI . . . 103
6.1.3 Architecture . . . 103
6.1.4 Plug-ins . . . 104
6.2 Structure JSON Data Files . . . 104
6.3 GUI . . . 107
6.3.1 Analysis and Manipulation of JSON file . . . 107
6.3.2 Picking a Visualisation Type . . . 109
6.3.3 Visualisation of the Data . . . 110
6.4 Architecture . . . 112
6.4.1 RequirejS . . . 112
6.4.2 Core . . . 114
6.4.3 Plug-in Architecture . . . 117
6.5 Plug-ins . . . 119
6.5.1 New Visualisation Type in Visualisation Picker . . . 119
6.5.2 New Visualisation Type in MindXpres . . . 121

7 Use Case 124
7.1 The Scenarios . . . 124
7.2 First Scenario . . . 125
7.2.1 Sneak Peak in Data File . . . 125
7.2.2 Analysing Data File . . . 126
7.2.3 Generated Recommended Visualisation Types . . . 127
7.2.4 Create Box Plot . . . 127
7.2.5 Create Historgram . . . 128
7.2.6 MindXpres Code . . . 129
7.3 Second Scenario . . . 130
7.3.1 Sneak Peak in Data File . . . 131
7.3.2 Analysing Data File . . . 131
7.3.3 Generated Recommended Visualisation Types . . . 132
7.3.4 Create Scatter Plot . . . 133
7.4 Conclusion . . . 134

8 Conclusions and Future Work 136
8.1 Contribution . . . 136
8.2 Future Work . . . 138
8.2.1 Analysing Data Files . . . 138
8.2.2 Recommended Visualisation Type . . . 138
8.2.3 Plug-ins . . . 139
8.2.4 Evaluation . . . 139

1 Introduction

We have all noticed that we live in a data-rich world. When you hear the term "information overload", you probably know exactly what it means, as it is something most people deal with daily. You can see it everywhere: large databases, organisations taking decisions based on their collected data, millions of transactions executed every second. The amount of information that is available to individuals and businesses has increased dramatically in the last few years, which means that we have far more data than we ever had [15].

Although the problem of information overload has existed for many years, in recent years the problem has become more widely recognised and experienced [8]. The entrance of information technology may be a primary reason for information overload due to its ability to produce information more quickly and to distribute this information to a wider audience than ever before. A question that we can ask ourselves is: *"Do we have this information problem because we have too much information?"*

The quantity and the growth of the data is not the problem, it does in fact present a wealth of potential [13]. The problem is that the amount of information that is available increased dramatically, but the ability to make use of it has hardly increased. A lot of the people who are responsible for analysing and communicating data are not schooled to do this. The available software tools on the market vary in how effectively they can assist

in analysing the data, but no matter how well these tools are designed, the quality of the result will depend on how skilled we are in utilising them. Since computers can not make sense of data in the same way that people can, we still need people with analysing skills to process data.

Data analysis can take multiple forms: visualisation (or data visualisation) is an approach used in visual data analysis which provides a way where information can be seen, explored and understood. In *visual data analysis* the data is visualised in a graphical representation that can serve to display clearly and effectively the messages carried by the data. *Statistical analysis* uses statistics to reduce large and complex data sets to a few numbers. However, statistical analysis that goes beyond the basics can only be used by trained specialist, while visual analysis is accessible to a broader audience. Also, visualisation taps on an ability that all but a few of us naturally possesses: vision. Vision is by far the dominant and most powerful of the five channels through which we sense the world. William Playfair realised that if data could be represented visually, they could be perceived, explored and understood more quickly and to a higher degree than not visually represented data [13]. Visualisation not only make the patterns, outliers and trends visible and understandable, but also helps us think by using the close relationship between vision and cognition [50].

In visual data analysis, graphs are used to explore data to see overall patterns and to see detailed behaviour. Graphs, also called visualisations, use spatial arrangement on a page or screen to visualise numerical data. Data represented on a graph are often easier to interpret than repetitive numbers or complex tables. The assumption seems to be made that creating good graphs is easy and natural. Yet, in literature many graph that give disinformation or misinformation can be encountered which disprove this. Disinformation is intentionally false or inaccurate information that is spread deliberately, while misinformation is information that is unintentionally false.

A lot of the visualisations created today are used in presentation tools. While presenting an idea that include data, it can be helpful to use a visualisation to present the idea or the message behind it. These visualisations can also make it easier to describe and defend the idea. While visualisation in presentation tools can be powerful, they can convey the wrong message or confuse the audience which will ruin the presentation. It is important to have the tools and skills to create visualisations that are not misleading and bring the idea or message in the best possible way. In this thesis we aim to help the user to create effective visualisations in presentation tools without falling into traps

that common presentation software packages seem to encourage. An effective visualisation presents data in a form that allows people to almost instantly see the message or idea behind the displayed data and also understand the information residing in the data.

When we examine the field of data visualisation more closely, we can identify two important components [44]. The first component is the supporting technical foundations for creating representations of data. This includes mathematics, computer graphics, user interface techniques, ... that allow users to create visualisations that are responsive, informative and in some case even beautiful. Think of spreadsheet application like Microsoft Excel[1], LibreOffice Calc[2], and also presentation tools like Microsoft PowerPoint[3], Apple Keynote[4] and LibreOffice Impress[5], which all belong to these technical foundations. The second important component is the human aspect. It is more challenging to develop systems that are useful and informative when you do not have a deep understanding of the tasks and goals of the user. A careful understanding of the capabilities of the users is necessary to develop informative information visualisations.

We believe that an important responsibility of a presentation tool is to make intelligent use of the human aspects when visualising data in graphs. What you can notice in current common presentation tool is that they have simple to use and well-designed visualisation functionality, which can makes it easy to create and adapt graphs. However, the convenience they provide is sometimes superseding the clarity and effectively of demonstrating the messages of the data through graphs. Some of these graphs are often filled with redundant symbols, distorting elements or have an incorrect visualisation type for displaying that type of data or message. At the end a user could create a presentation with graphs that do not tell the viewer much about the data or even causing a distraction. When the user gives a presentation to the audience, it is important that the graphs presents the data in the best possible way to support the user in his or her talk.

[1] `http://office.microsoft.com/en-us/excel/`, last accessed on 07/07/2014

[2] `http://www.libreoffice.org/discover/calc/`, last accessed on 07/07/2014

[3] `http://office.microsoft.com/en-us/powerpoint/`, last accessed on 07/07/2014

[4] `https://www.apple.com/mac/keynote/`, last accessed on 07/07/2014

[5] `http://www.libreoffice.org/discover/impress/`, last accessed on 07/07/2014

In Chapter 2 we will demonstrate some of these graphs that do not present the data effectively. We will also introduce the research area information visualisation and discuss some of the historical highlights in this still evolving research area. By studying the challenges in information visualisation we will establish three problematic topics that will be used as a basis for this document. In Chapter 3 we will investigate and discuss these topics, which will give us a better insight into the human aspects of information visualisation. It will help improve our knowledge about information visualisation and provide us with principles and guidelines to create decent visualisations that presents the data and its message efficiently and effectively.

When we have more knowledge about the human aspects of information visualisation we can start investigating common presentation tools. In Chapter 4 we will investigate if these technical foundations for visualising data trough graphs provide enough support to help the user create decent visualisation of the data. By using the principle and guidelines defined in Chapter 3 we will be able to validate the quality of the visualisations created in these common presentation tools.

We will discuss our thoughts about the ideal visualisation process and the minimum requirements a presentation tools has to satisfy to be able to help the user create quality visualisation. Additionally, we will introduce the MindXpres presentation tool in Chapter 5. MindXpres has a very interesting plug-in architecture which enables the reuse and integration of media types. This plug-in system gives us the opportunity to develop a new process in creating visualisations for data from scratch.

In Chapter 6 we provide a solution by implementing a new tool for creating visualisation for data. This tool will follow the guidelines and principles defined in Chapter 3 to create the visualisations. The architecture will be based on the ideal visualisation process discussed in Chapter 5. The purpose of this tool, called the Visualisation Picker, is to help the user pick the best visualisation type for the data and also present the data in an automatically generated graph. The Visualisation Picker also makes it possible to transfer these generated graph to MindXpres, so that they can be presented to an audience.

In order to demonstrate the benefits that the Visualisation Picker claims to have, we will provide a use-case in Chapter 7 in which we walk through the process of picking and creating a visualisation for a data set. We believe that our solution provides the basis for future work, which we discuss in Chapter 8, together with a final conclusion of this thesis.

By investigating the human aspect of data visualisation, we aim to provide better support for creating visualisations of data. The support provided for the presenter by current common presentation tools is in most cases not enough to create visualisation that can help the audience understand and explore the data efficiently. With our implementation called the Visualisation Picker we try to solve as many of the issues we have found in current common presentation tools and try to improve the quality of the visualisations created for presentations.

2
Problem Statement

In this chapter, we see that data visualisation is an umbrella term for scientific visualisation and information visualisation. We will elaborate on some new terminology that will be used throughout this document. We will also see some historical highlights that have paved the way of today's information visualisation. Even with al that history behind us, we can observe that today there are still a lot of visualisations that do not present the data effectively. Once the history and the challenges of information visualisation have been examined we will have a better idea of what is still going wrong in the human aspect of information visualisation.

2.1 Information Visualisation

As Stephen Few explained in his book *Now You See It: Simple Visualisation Techniques for Quantitative Analysis* [13], the term visualisation can be preceded by three words when used for visual representations of information:

- Data visualisation
- Scientific visualisation
- Information visualisation

Data visualisation can be seen as an umbrella term to cover all types of visual representations with which you can perceive, explore and even communicate

data. When the representation is visual and represents information, then it is part of data visualisation. *Scientific visualisation* is a subset of data visualisation and comprises visual representations of scientific data that are usually physical in nature, rather than abstract. Example of scientific data is data from MRI or X-ray scans that result in displaying objects that possess actual physical form. The scientific data attempts to present this form in a way that is easy to recognise it as the object and explore. *Information visualisation* is a subset of data visualisation and is "the use of computer-supported, interactive, visual representations of abstract data to amplify cognition"(Card, Mackinlay and Shneiderman) [3].

From this point onwards we will mostly focus on information visualisation. When we take a closer look to the definition of Card et al. we can notice that the purpose of information visualisation is not to make pictures, but to help us think, amplify cognition. Viewing and interacting with these visualisations helps us think about information by supporting our memory and representing the data in a comprehensible way for our brains. Computer-supported means that the visualisation is displayed by using a computer, usually on a screen. This is a constraint in the definition that is not really necessary, drawing your visualisation with pencil and paper is also part of information visualisation. It is also interesting to note that the visualisations do not need to involve a visual experience, sound and other sensory modalities can be used to present data.

The next part of the definition, the interactive visual representation, tells us that the data is in a visual form using attributes such as locations, lengths, colors, shapes, sizes. These attributes can be manipulated with interaction and allow us to see patterns, trends and exceptions in the data. Interactive also means that it is possible to filter data and to focus on details. The last term in the definition is abstract data, which covers quantitative data, processes or relationships. A visual representation of physical objects (e.g. geography or human body) is not abstract data.

2.2 Terminology

2.2.1 Graphs, Charts and Visualisations

Graph and chart are commonly used to display data in a graphical form. There are definition that state a difference between these two terms, but we will use them mostly equivalently. The term visualisations will be used as collective term for graphs and charts.

2.2.2 Pattern Perception

When looking around we do not really see edges, colors, sizes etc. What we see are objects like books, tables, trees and so on. We do not need to focus on every single part of an object to see the pattern. This is the same with pattern perception. When used properly we can view the whole data pattern in the visualisations almost instantly. We will discuss visual and pattern perception more detailed in section 3.4.1.

2.2.3 Graphical Integrity

A visualisation has graphical integrity if it accurately represents the data, or in other words, when the visualisations is consistent with the numerical representation. The visualisation has to be clear, detailed enough to easily see the pattern of the data and should also be labelled to avoid distortion and ambiguity.

2.2.4 Distortion

Distortion in information visualisation is when the message presented in the visualisation differs from the message contained in the data. Put differently, misrepresenting the data in the visualisations. You can create distorted visualisations, also known as misleading visualisations, by picking the incorrect visualisation type and ignoring the Design Aesthetics principles and guidelines. More about distortion can be found in section 2.4.

2.2.5 Cognitive Load

The term cognitive load is commonly used to describe the amount of work imposed on working memory. Our working memory is limited with respect to the amount of information it can hold and the number of operation it can perform on that information. When we want to create our visualisation to be effective we have to keep these limitation in mind and adapt our visualisation to it.

2.2.6 Categorical, Ordinal and Quantitative Data

The information inside a visualisation can be partitioned into quantitative values, categorical values and ordinal values. Variable that are quantitative of nature are numerical values. Other variables, like year, names, etc. are categorical. It is not because a variable is a number, that it is quantitative. For example, zip codes are numeric but it does not make sense to calculate the average value of the zip codes. An ordinal variable is similar to a categorical variable, but the difference between the two is that in ordinal data there is a clear ordering of the variables.

2.3 The History of Information Visualisation

Now we will look to some of the historical highlights that have paved the way of today's information visualisation. In the last ten years the area of information visualisation has witnessed a high increase in its popularity and is becoming an increasingly necessary tool to think, explore and comprehend complex data [39]. Many disciplines have started to recognise the efficiency of visualisation as an aid for research and decision making. However, information visualisation has its roots in a long historical tradition of representing information using pictures and shapes in a way that combine art, science and statistics [19].

The ancient Babylonians, Egyptians, Greeks and Chinese used to create visualisations by drawing in sand or scratching on rocks. Most of these visualisations where created to produce maps for navigation, movements of stars, plans for crop planting and city development. Figure 2-1 is an example of a map of the world, visualised from the perspective of Babylonia on a clay tablet ca. 500 BC.

Figure 2-1: The Babylonian World Map (Source:*Map of the World*, by British Museum)

Later visualisations were created on papyrus, which made it easier to share and annotate the visualised information. The Turing Papyrus Map, visible in Figure 2-2, is considered the oldest surviving map of topographical interest [22]. Recent studies have shown that it gives remarkably accurate colour-coded geographical information [19].

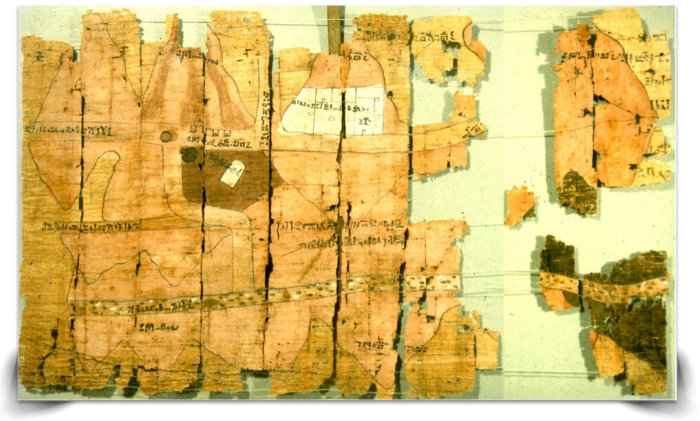


Figure 2-2: Turin Papyrus Map 1150 BC (Source: *Turin Papyrus Map From Ancient Egypt*, by James A. Harrell)

After papyrus they started utilising parchment, material from animal skin, that was often used for writing documents, manuscripts and books. The Tabula Peutinger (Figure 2-3) is a map created on parchment by the Romans as a tool to efficiently plan the movement of their armies and trade throughout the empire.

Figure 2-3: Tabula Peutinger 366-335 BC (Source: Bibliotheca Augustana)

What you can notice in the three map examples above is that during the years, the maps get more detailed and different techniques are used to show the information. The following example shows a part of Ptolemy's World Map created in the 2nd century. Through the use of latitude and longitude markings they developed a global coordinate system that is the forerunners of modern maps [19].

Figure 2-4: Ptolemy's World Map (Source: The British Library)

Ptolemy's World Map was not the only forerunner that emerged in the 2nd century. It was also the century where they started to use tables. A data table is a arrangement of information in tabular form having rows and columns, and is used to view and compare values. The earliest known data table was used to organise astronomical information for navigational aid [13].

Most of the visualisations before the 18th century were created to describe geographic or astronomic information. In the 18th century we begin to see visualisations created to make sense of social and historical data. The main reasons why they started with visualising social and historical data is the invention of new measurement devices and new ways to gather data. They also started visualising data about towns, countries or states to make sense of their economies, populations and medical data. It was the century where abstract graphs and function graphs were introduced, together with the start of statistical theory [31] and the collecting of empirical data.

It wasn't until the 19th century that information visualisation grew into a mature discipline with the development of many formal and technical innovations. An important contributor of these innovations was William Playfair. Playfair was a Scottish economist and politician who developed, along with Joseph Priestly, time-series graphics to show trends in data over time [13,19,48]. Playfair invented the bar graphs, was the first that used line graphs to represent change through time and also invented the pie chart. In his famous example, visible in Figure 2-5, he compares the price of wheat to wages from the 14th century in order to see whether the price of wheat had increased relative to wages. The chart contains 3 parallel time-series: at the bottom are wages represented, a middle bar chart plotting the cost of wheat, and in the top the reigning periods of the English monarchy.

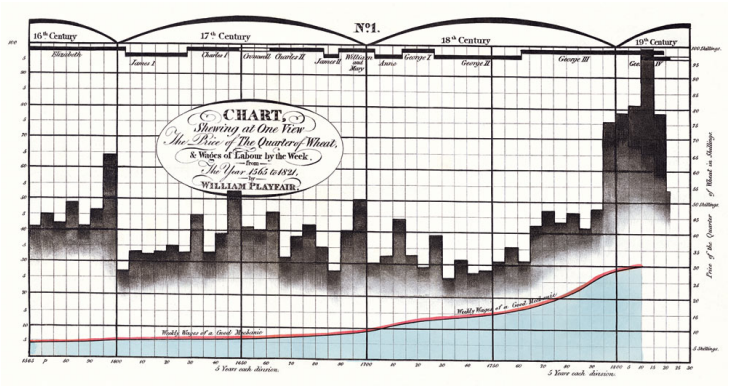


Figure 2-5: Playfair Chart (Source: excelcharts.com)

Another important contributor from the 19th century is Charles Minard. Minard was in the 19th century a civil engineer working in Paris. His most famous work (Figure 2-6) describes the fate of Napoleon's army as it advanced and retreated from Moscow in the winter of 1812. The diagram is a visual timeline and a geographic map containing the size and direction of the army, temperature, landmarks and locations. We see that the army begins the campaign at the Polish border with 422 000 men that decreases the deeper it gets into Russia and the lower the temperature. By the time Napoleon reaches Moscow the army has halved in size. A slow retreat begins in Moscow that moves from right to left culminating at the Polish border with a mere 10 000 survivors. This visualisation manages to show a number of different numeric and geographic facts and is a wonderful example of how visualisations can turn raw numbers into interesting stories about human events.

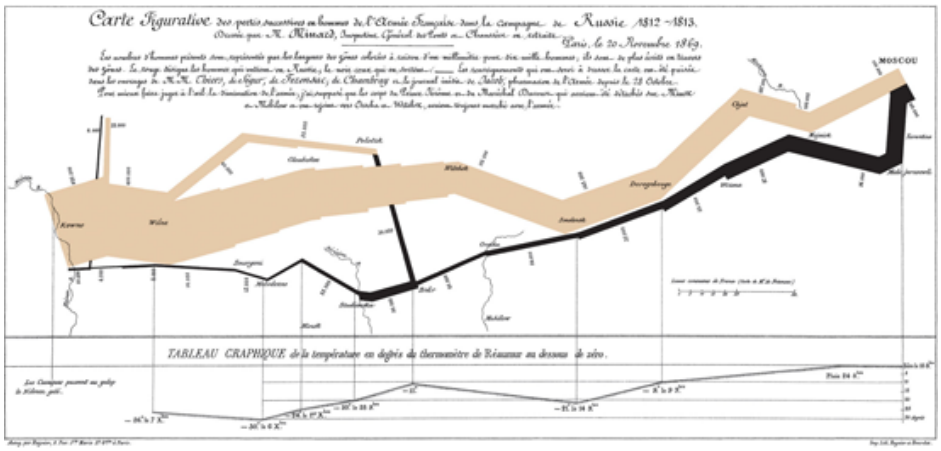


Figure 2-6: Minard: fate of Napoleon's army (Source: `http://www.masswerk.at/minard/`)

Playfair and Minard where not the only ones who created interesting visualisation in the 19th century. Dr John Snow created a visualisation in order to prove his theory about the cholera epidemic. He drew a map of Soho marking each case of cholera with a black dot. By analysing the distribution of the black dots in relation to the water pumps, Snow was able to find the origin of cholera outbreak in Soho [29]. Francis Galton contribution to the history of visualisation was his meteorological maps showing the distribution of air pressures and wind directions [19, 35]. We can go on a while like that. The 19th century had a lot of graphical innovations and is seen as the 'golden age' of statistical graphics and thematic cartography.

In the 20th century information visualisation had become commonplace in magazines, newspapers and books. In the early mid of the 20th century, Otto Neurath designed a visual language that was called the ISOTYPE [26]. With this language he did not only want to show the numbers in a way that was easy to read, but also clearly communicate what they mean. ISOTYPE is the acronym for International System of Typographic Picture Education. Predecessors like Playfair and Snow worked under the assumption that explanatory text would be available with the visualisation, or that someone would be available to explain it. Neurath aimed to make his visualisation self-contained and self-explanatory, so that they would stand on their own, without the explanation of somebody or by text.

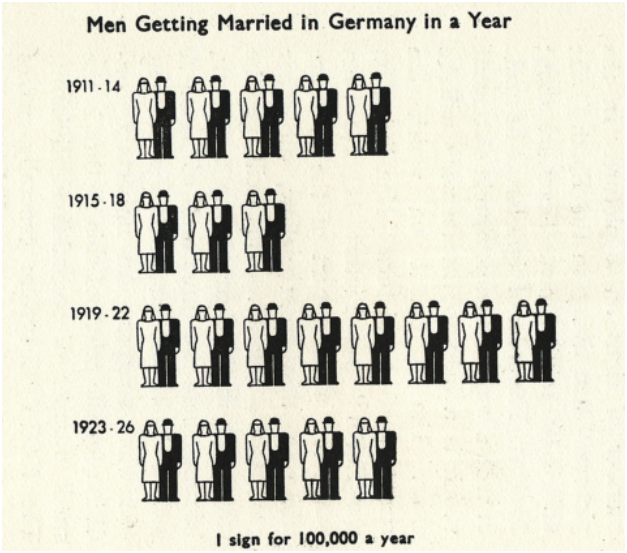


Figure 2-7: Otto Neurath: ISOTYPE (Source: `jdh.oxfordjournals.org`)

From that point on, a lot of researchers started publishing books on how to clearly, accurately and efficiently express information in a visualisation. Jacques Bertin introduced in 1967, with his book *Semiologie graphique*, the principles of visual language and argues that visual perception operates according to rules [2]. In 1983 a new groundbreaking and popular book about information visualisation was published by Edward Tufte, named *The Visual Display of Quantitative Information* [48]. Tufte's book shows us that there is a effective way of displaying information visually, in contrast to some of the visualisations we were creating, which were not effective at all.

The next revolution in information visualisation came with the emergence of computer technology. Larger amounts of data could be visualised very effectively and fast with this new technology. Not much later it was also possible to make interactive visualisations. With these interactive visualisations it became manageable to directly manipulate the visualised information. By the 70s and 80s the first full-colour computer visualisations were being developed and interactive graphics were providing new ways of revealing the information hidden in data by making it possible to navigate through the graph in three or more dimensions. The emergence of the internet, the availability of new software such as Flash[1], new tools for creating visualisation such as D3.js[2] and the increase in available data made a huge increase in different type of visualisations possible. With all these new possibility for creating visualisations of data we see an increase in visualisations created by the general public, whereas in the past most visualisations were created by statisticians and scientist [19].

2.4 The Challenge of Information Visualisation

Creating efficient visualisation is surprisingly difficult. When we create a visualisation, we have to step through a series of choices, including which type of graph we should use and several aspects of its appearance. Most people walk through these choices as if they are not important, with only a vague sense at best of what works and what does not, why one choice is better than another. Without guiding principles, it can easily go wrong for even the most simplest matters. This will be illustrated by the following topics where the visualisation of data can ran into trouble.

2.4.1 Choosing Visualisation Types

With so many visualisation choices available, it is sometimes difficult to know which type of visualisation will show the information in the best possible format and maximise the understanding of the message contained in the data. It is not always possible to say with certainty that one type of visualisation will work best for a given type of data, but some types of visualisations will definitely work better the others on the same data. What follows are some examples of bad visualisations due to incorrect choice in type of

[1] `http://www.adobe.com/products/flashplayer.html`, last accessed on 26/07/2014

[2] `http://d3js.org/`, last accessed on 26/07/2014

visualisations.

Figure 2-8 is a visualisation of the revenue and expenses of the Simon Fraser University in 2006 & 2007. As we can all notice, a pie chart is not the best visualisation to present this data. There are too many pie segments, which makes it difficult to read the smallest segments. The data is also represented as angles, which does not provide precise information and makes it more difficult to see the differences in size between the segments. A better alternative could be a (segmented) bar graph or dot plot, which have better performance of pattern perception [6, 15, 48].

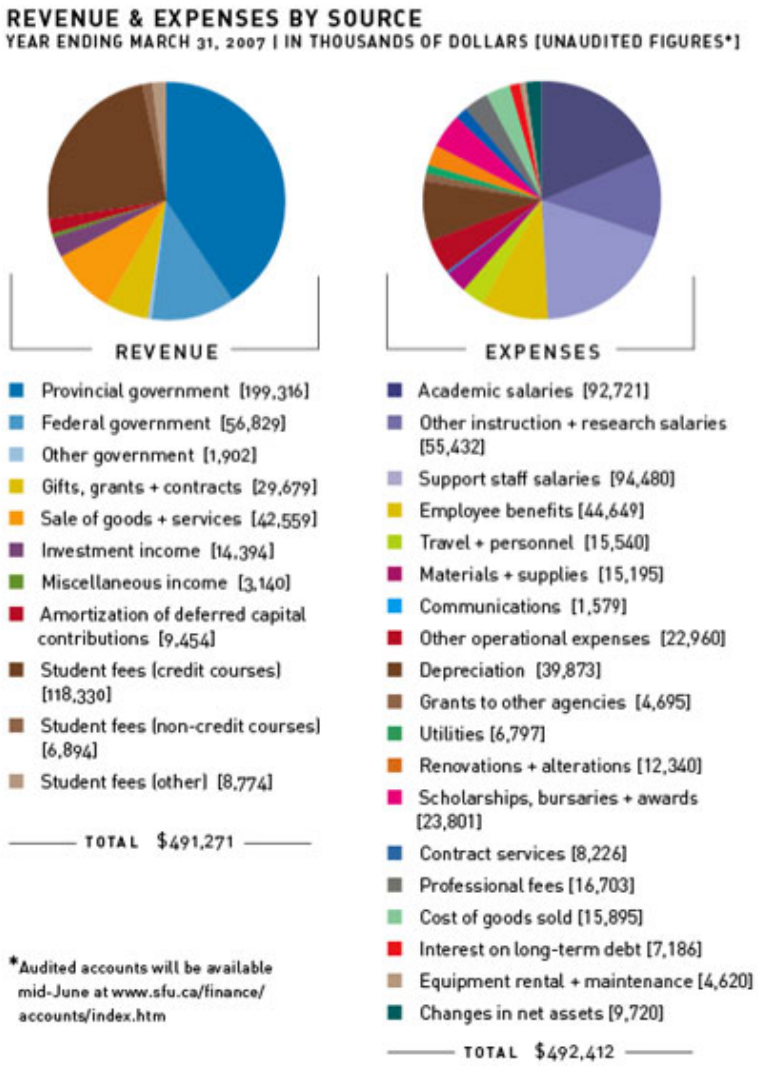


Figure 2-8: SFU 2006 Report from President (Source: `www.sfu.ca`)

The next example of bad visualisation is a three-dimensional representation of data, to compare the attributes of price, number of bedrooms and travel time to work. When you take a closer look to Figure 2-9 you can easily spot the problem. It is for example not clear whether house A costs more than house B. This is a problem mostly encountered when graphs has more then two dimensions. According to Tufte [48], creating three-dimensional graphs is the same as creating 'Chartjunk'. However, not every researcher shares the same opinion as Tufte. Spence and Lewandowsky [43] believe that it is not harmful to add additional dimensions to a graph; irrelevant dimensions make a graph more attractive and can make the processing of it more quickly. Robert Spence states that for 3D graph to be useful, we have got to be able to move it and to interact with it.

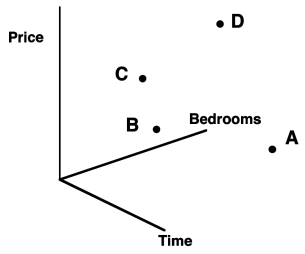


Figure 2-9: Price, number of bedrooms and travel time to work

An alternative representation for three-dimensional visualisations of data is a systematic structure formed from the three possible two-dimensional views of the data, as show in the scatterplot matrix in Figure 2-10. Now two houses can be compared easily with respect to the three attributes. With this type of visualisation you can now easily see that house B cost more than house A. A disadvantage of this visualisation is that there are now three points needed to represent a house, potentially increasing the cognitive load on the user.

Figure 2-10: The scatterplot matrix associated with the data of Figure 2-9

The type of visualisation used to present your data also depends on what you want to show with the visualisation, what your message is to the viewer. In Figure 2-11 we can see a grouped bar graph with the meat production of a fictive country, per season, over 2 years in thousands of tonnes. With grouped bar graphs it can be difficult to tell the difference of elements between groups however it is ideal to compare between each element in a group. So when you want to compare the production of the different types of meat in a season, and that for each season, the grouped bar graph is a good visualisation. In Figure 2-12 the same data is visualised in a line graph. We can notice that the line graph represents the trend of the meat production better than the grouped bar graph.

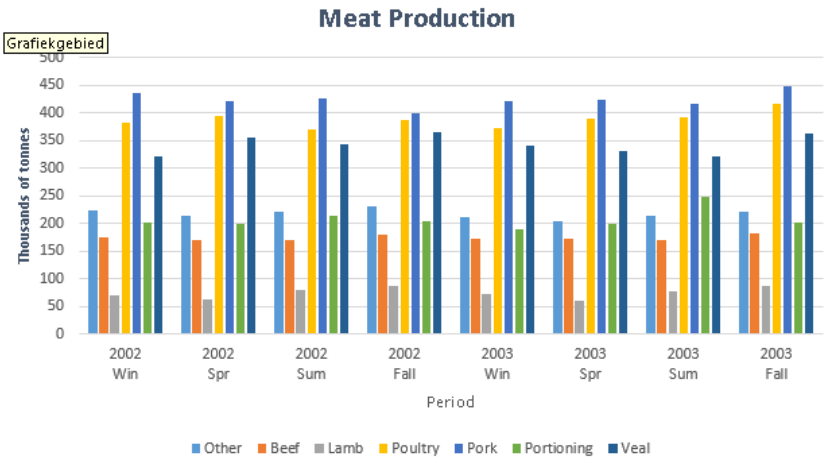


Figure 2-11: Grouped Bar Graph: Meat production

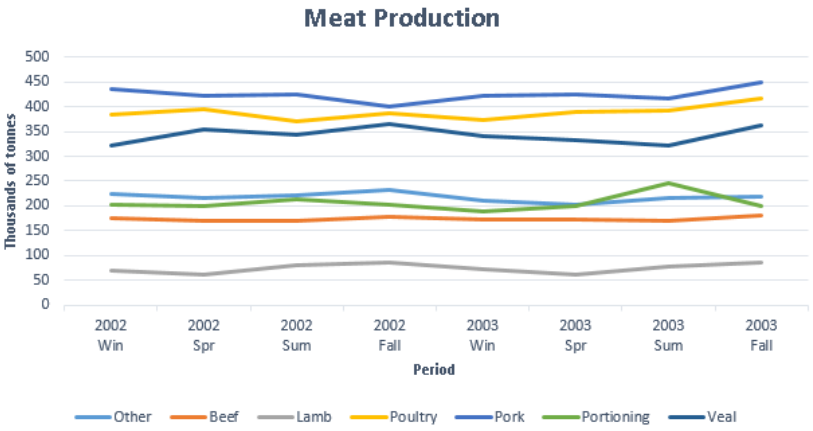


Figure 2-12: Line Graph: Meat production

With so many visualisation choices available, it is sometimes difficult to know which type of visualisations will fit the type of data and maximise the understanding of the message contained in the data. In this section we showed some examples of data visualised in not the best type of visualisation and in some cases we already suggested a type of visualisation that suits the data and message better. In section 3.2 we will have a closer look on how to select the best visualisation type for presenting the data, by going through some steps and principles.

2.4.2 Graphical Integrity

Much of the twentieth-century thinking about visualisation has been preoccupied with two issues [48]. First with the assumption that visualising data were mainly a different technique for showing the obvious to the ignorant. This assumption led to two results in the visualisation branch:

- Visualisation had to be alive, overdecorated and exaggerated (so people do not fall asleep when looking at it). Why this is a fruitless path will be explained in section 2.4.3
- Main task of visualisation analysis was to detect and denounce deceptions. You can of course do more with visualisation then only detect and denounce deception

The second issue is deceptive visualisations, which are visualisations that distort the underlying data, making it hard for the viewer to learn the truth about the data. This type of visualisation may happen when the designer chooses to give readers the impression of better results than is actually the situation. In other cases, the person who creates the graph may want to be accurate and honest, but may mislead the reader by a poor choice of graph type or poor graph construction [40]. A lot of these misleading visualisations have common errors, here are some examples of these common errors.

A lot of deceptive visualisations are created by scale and axis manipulation. Figure 2-13 is an example of such a manipulation, in which you can see that they place the vertical axis upside down. When you look at the graph for the first time, it looks like that the number of murders committed using firearms is mostly descending since 2005. Afterwards when you take a closer look to the y-axis you notice that it is not descending but is actually ascending. So by just creating a upside down y-axis a deceptive visualisation is created.

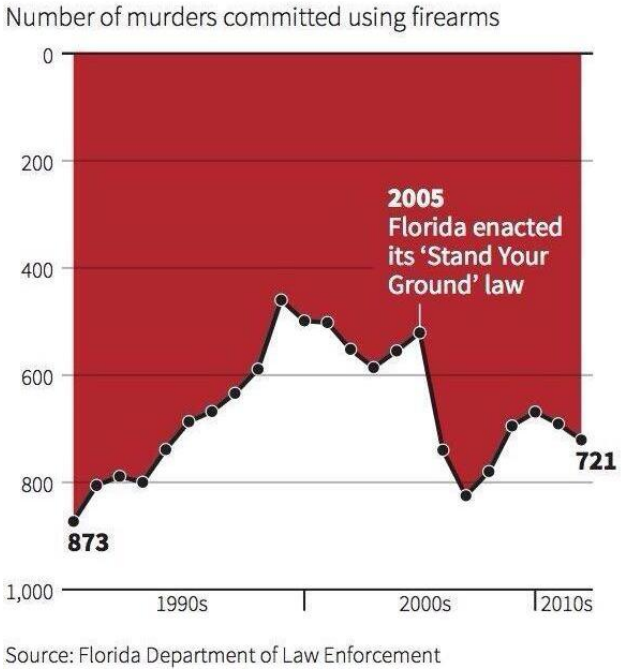


Figure 2-13: Gun deaths in Florida

The next visualisation (Figure 2-14) created by Fox news channel shows that gas prices were consistently rising. On February 20, they displayed this graphic that used three random data points to show the national average cost of gasoline over a year: one data point is the national average gas price from the day the graphic aired, the other two are from the previous week and the previous year. This graph was used to (falsely) blame Obama for rising gasoline prices[1]. The first issue with this visualisation is that the x-axis does not have a consistent scaling which can be confusing for the audience. The second issue, which is probably the biggest one, is the fact that Fox's chart omitted what happened in the 12 months between February 2011 and last week. When we look at Figure 2-15 we see the same data visualised in more details. This more accurate visualisation of gas prices over the one year period shows that gas prices in February 2012, the highest point on Fox's graphic, is not the highest point in Figure 2-15[2]. The gas prices where actually at its peak in April-May of 2011 [1].

[1]Source http://mediamatters.org/embed/clips/2012/02/21/22989/fnc-an-20120220-gaspriceschart last accessed on 02/06/2014

[2]Source: http://mediamatters.org/research/2012/10/01/a-history-of-dishonest-fox-charts/190225 last accessed on 02/06/2014

Figure 2-14: Fox News: Rising gasoline prices

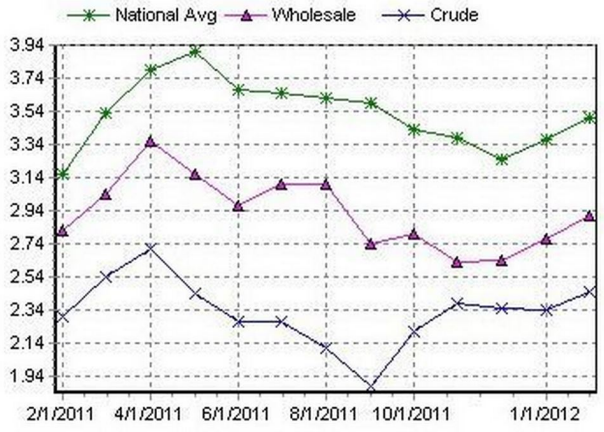


Figure 2-15: Rising gasoline prices detailed (Source: *A History Of Dishonest Fox Charts*, by Hannah Groch-Begley & David Sher

When using three-dimensional effects it can happen that the visualisation becomes misleading. Figure 2-16 might look more attractive than Figure 2-17, but is actually very misleading. There is no grid lines on the vertical axis, and because of the perspective it looks as though the value for c is greater than the value for a. But in fact they are identical to each other, which is shown in Figure 2-17.

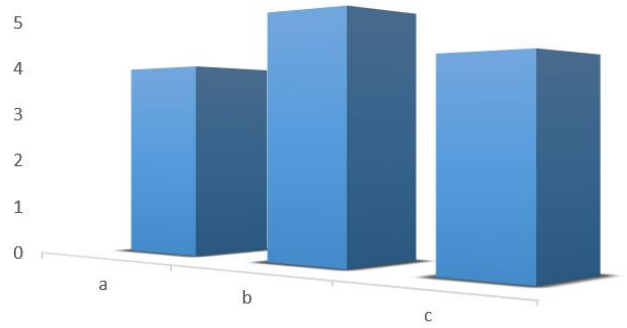


Figure 2-16: Misleading 3D bar chart

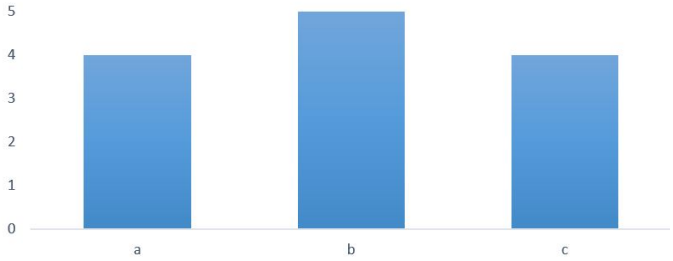


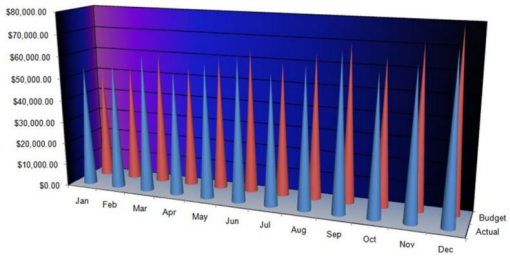
Figure 2-17: 2D Bar Chart visualising same data as Figure 2-16

In this section we saw some common ways that a visualisation can be misleading. We saw that poor graphical integrity can be caused by the designers who chooses to give readers the impression of better results than is actually the case. Of course not all designers are dishonest, some may create visualisation that mislead the reader due to the lack of skills of the designer in selecting the correct visualisation type and designing the visualisation. More about misleading visualisation and how to avoid them in section 3.3.

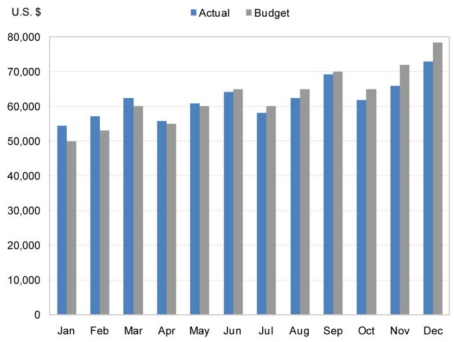
2.4.3 Design Aesthestics

A lot of visualisations that are available to us today do not only show the data but are also decorated to look beautiful. Creating a good visualisation of your data cannot be compared with designing flyers or posters. When overdecorating a visualisation it is possible to complicate the visualisation or even add and show incorrect data which can confuse the viewers. Visualisations are in some cases produced without focus on their main purpose: to enlighten and inform the reader [30]. In following examples you will see what can go wrong when decorating your visualisation extravagantly.

Figure 2-18 is an example of Stephen Few in his book *Show Me The Numbers: Designing Charts for Enlightening Communication* [13]. The reason why Figure 2-18a is an overdecorated graph is that this decoration makes it difficult to determine the values actual and budget, or to compare the actual expenses to the budget across time. When we transform this visualisation to a less decorated one, we could become as possible outcome Figure 2-18b. What can be noticed is that this visualisation communicates more to the viewer then the overdecorated one.



(a) Overdecorated visualisation

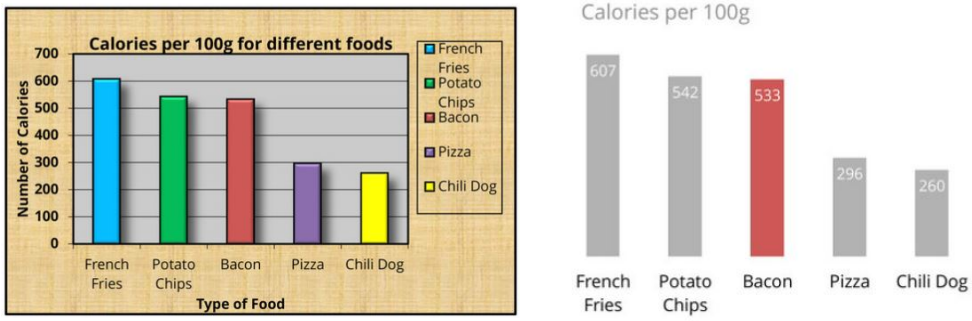


(b) Transformed Visualisation

Figure 2-18: Overdecorated Visualisation Transformed

The website of Darkhorse[1] offers a slideshow [27] that demonstrates the amount of unnecessary decoration that can be removed from a typical Excel- or Powerpointgenerated visualisation to make the presented data become more clear and communicative to the viewer. The slideshow starts with Figure 2-19a and ends with the transformed Figure 2-19b. In between those visualisations from the slideshow they show in each slide a reduction in the decoration. For this they base themselves on a concept that Edward Tufte introduced in 1983, stating that *Data-ink is the non-erasable core of the graphic*. This topic will be discussed in more detail in section 3.4.3.

[1] `http://darkhorseanalytics.com/` last accessed on 02/06/2014



(a) Overdecorated Visualisation

(b) Transformed Visualisation

Figure 2-19: Overdecorated Visualisation Transformed (Source: Darkhorse)

2.5 Overview

In this chapter we have briefly gone over the history of visualisation and some of the challenges of creating visualisations. As we can notice, in the field visualisation we have already come a long way, but low quality visualisations are still being created. It is now more then ever possible to create good visualisations with tools, but good visualisations require skill and some general guidelines. The good news is, although the skills required to present data effectively are not all intuitive, most of them are easy to learn. In the development of a new tool for creating visualisations, we not only want to fix the presented issues because they might cause bad visualisations, but also to lower the skill level and time needed to create a good visualisation.

3
Principles of Graphing Data

In Chapter 2 we have taken a closer look at the history of visualisation which gives us an interesting view on the evolution of visualisation. We also studied the literature concerning the challenges in information visualisations together with some examples. This gave us an extensive look on the way we create visualisations nowadays. In this chapter, we take a closer look at what a good visualisations could mean. Studying the basic ideas, the methods and the principles created by important researchers in the field information visualisation will help us to get a better idea of what creating a good visualisation really is.

We start by introducing the most commonly used and recommended visualisations. Then we try to solve the first challenge introduced in chapter 2, discussing how to select the best visualisation type for a particular data set. To do this, we will have to go through a series of steps, for example identifying the data type and the message contained in the data, choosing the correct relationship type and best visualisation type.

We will also take a closer look to the second problem that afflicts information visualisation by investigating why people create misleading visualisation and how they do it. We will also discuss some principles that will help us detect those misleading visualisations and help us avoid creating this misleading visualisations by ourself.

This chapter will also cover the problem of overdecorated visualisation. We will take a closer look on how to make sure that the visualisation draws the user's attention to the message included in the data and not to something else. First we learn the power of visual perception and how to use that power to create good visualisations. Then we talk about the data-ink principle and how to use it for graphical redesign. After that we will see how to use colour, legends, tick marks and grid lines in the visualisation. To finish this chapter we also briefly mention the design decision of a table.

By looking at all these defined principles and the research done in information visualisation, we can investigate if current visualisation tools suffice for what we really need to present our data in the best visualisation possible.

3.1 Types of Visualisations

In the section we will introduce some visualisation types, also know as chart types or graph types. Most of these types are commonly used and most of them will already be familiar. Still, these types of visualisations will be used as example tools in this document, and a small introduction of them will help to better understand these examples.

3.1.1 Bar Chart

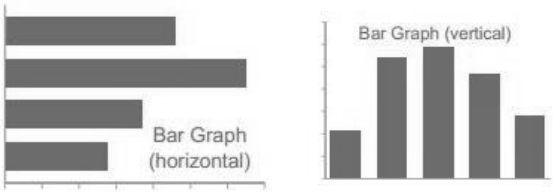


Figure 3-1: Bar graph

Most people know a Bar chart as a graph with vertical bars that have lengths proportional to the values of categories separated by white space. This type of graph is actually named a Column charts or graph. They are almost the most seen graph type in a all different types of publications [28]. A Column graph displays on numeric variable on the y-axis and categorical data on the x-axis. A problem with the vertical arrangement of the bars is that sometimes labels on the x-axis are squashed or turned 90° which makes it hard to read and visually not beautiful.

When having the above defined problem with Column graphs it is better to use Horizontal bar graphs. By placing the numeric variables on the x-axis and the categorical data on the y-axis you do not need to use vertical text labels or abbreviations. The graph shape is particularly useful to present categorical data with longer names.

Two other types of visualisations that use bars are Stacked bars and Multiple-bar graphs, which deal with multiple variables per category. Stacked bars are graphs where several values are stacked making a single horizontal or vertical bar per category. Multiple-bar graphs, also called Grouped bar graphs, are graphs that present several variables plotted as adjacent horizontal or vertical bar graphs belonging to the same category, spread over multiple categories.

3.1.2 Dot Chart

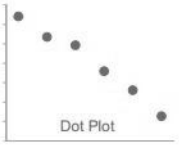


Figure 3-2: Dot plot

A Dot chart, or Dot plot, is a special type of the Stacked bar graph. Cleveland and McGill created the Dot plot after conducting a number of experiments to determine which perceptual task was the most accurate. This was the task of reading position along a common scale, and they discovered that Dot plots where a superior alternative for Stacked bar chart or Pie charts for this task [4]. It uses a minimum of ink to present the data and indicates the value by the position of the dot relative to the axis and not for example by the length of a bar, as in a Bar chart.

3.1.3 Line Chart

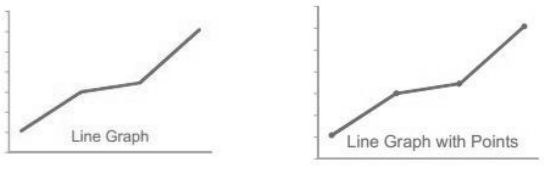


Figure 3-3: Line chart

A Line chart is a graph that displays data as a series of data dots which are connected by straight line segments. You can draw this chart with visible points, to nuance the data points, or without them. They are useful to spot trends of continuous data. Line graphs can have a categorical and numerical scale but also two numerical scales. It is called a X-Y plot when the line graph has two numerical scales, and it shows the degree and pattern of the relationship between two variables. Line graphs are mostly used to show information that is connected in some way, for example change over time.

3.1.4 Scatter Plot

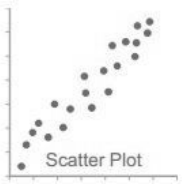


Figure 3-4: Scatter plot

A Scatter plot is also an X-Y plot, usually used for visualising a linear correlation coefficient or fitting a regression line. It gives a good visual picture of the relationship between two variables, and helps to interpreter the correlation coefficient by presenting each value as a point. In Figure 3-17 we can see some examples of Scatter plots, that cover correlations from perfect positive correlation to perfect negative correlations. Sometimes a linear trend line is added, which can show the overall direction of the values.

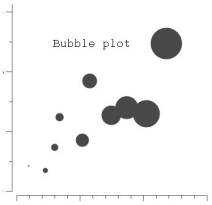


Figure 3-5: Bubble plot

A Bubble chart is a special case of a Scatter plot. By altering the size of the dots it can display additional information. This type of visualisations can visualise three dimensions which makes it useful when trivariate data has to be presented. Bubble charts are also used a lot on maps and are called bubble maps.

3.1.5 Pie Chart

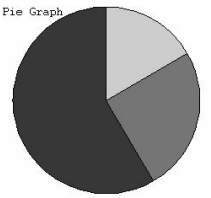


Figure 3-6: Pie chart

Pie charts are used to represent data that only has one numeric and one categorical variable. The numeric values are represented as angles of a slice, while the categorical variables are presented as slices. Due to the fact that the numeric values are presented with angles, it is difficult to provide precise information. The human eye is not naturally skilled in comparing angles. Some graph designers add numeric values or percentages to each slice to get a more detailed view of the data, but this can also create clutter. It is important to use a new colour for each slice and a key slice starts at 12 o'clock. Also Pie graph are most effective when at least most of the slices represent 25% to 50% of the whole.

3.1.6 Histogram

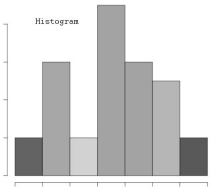


Figure 3-7: Histogram

A histogram always has two numeric axes and are mostly used for showing distributions within a large dataset. In most cases the x-axis is a continuous variable, divided into an arbitrary number of categories. When the graph handles a single variable, the bars of continuous variables touch each other by convention. When using categorical variables in your histogram then the bars are better presented with spaces between them.

A frequency polygon is like a histogram, but uses lines that joins points to illustrate a distribution instead of bars. It has the advantage that it can show a modest number of related distributions clearly on a single chart, using different lines or even symbols.

3.1.7 Box Plot

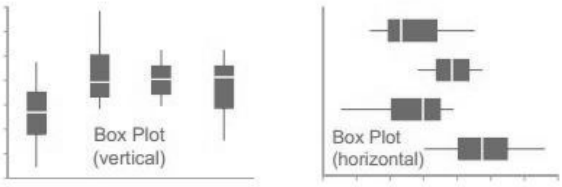


Figure 3-8: Box plot

The Box-and-whisker plot or Box plot is a graph for summarising the distribution of one or more continuous variable. It is more useful than a Histogram for picking up key aspects of the distribution. A Box plot will clearly show where the odd extreme values are and also show where values are systematically located further from the center in a certain direction. In a Box plot the distribution of data is shown by using 5 numbers: The minimum and maximum value, the first and the third quartile and also the median.

3.2 Choosing Visualisation Types

With so many visualisation choices available, it is sometimes difficult to know which type of visualisations will show the data in the best possible format and maximise the understanding of the message contained in the data. How does one select the best visualisation for his data? A possible solution is to try all the visualisation types until one is found that presents the information in the best possible way. With the host of tools for creating and editing these visualisations (more information in section 4) it is nowadays more easy for the creator to try multiple visualisations. Of course, picking a visualisation from scratch by trial and error is not the best way. A more efficient way would be the availability of a book or document with rules that can guide you to the best visualisation for every possible dataset.

Most of these books or documents that have been written in this field do not contain pages full of rules that could help creating visualisations. It is not easy to define visualisations rules that work on every available dataset. Fortunately, a lot of researchers provides us with books and documents that include interesting guidelines. In this section we will take a closer look at some of these basic rules and guidelines. For this we will mostly use the literature produced by the following researchers: William S. Cleveland [5, 6], Edward R. Tufte [48], Stephen Few [13, 15], Michael Friendly [18]. These researchers did not only write guidelines about which type of visualisation suits a certain dataset, but also guidelines about the format of better visualisation (more information in section 3.4). What follows in this chapter is a step-by-step guide with graph selection guidelines for the more commonly used visualisations. The books of the researchers are a good read when you also want information about less commonly used visualisations. The guidelines of the more common visualisations will be used in the implementation. The steps that we will cover in this section are presented in a logical order, but following the precise order of the steps is not required.

3.2.1 Determine Your Message and Identify Your Data

The first step that is essential in the visualisation selection, and which is also the step that is often missed, is determining what you want to say. We must ask yourself, "what is the real purpose of this visualisation?". It is not enough to simply take the data we want to present and transform it into some type of visualisation. We can compare this with communicating with someone; first select about what we want to talk about, then choose our words and arrange them in a way that we communicate with that person in the best possible and understandable way. If we just communicate everything that comes into our heads at that moment, without taking into account the purpose of your communication, someone may question our sanity. Once we know the clear purpose why the visualisation should exist, we will be able to determine what is needed to communicate our message.

Before we can communicate our data, we must know what the data means and what is relevant to visualise. What you want to say about the data also depends on the type of data. When you take a closer look in the book *Visualizing Data* [5] written by Cleveland you notice that it is organised by arranging the global structures around the different types of data. Each chapter treats a different data type: univariate, bivariate, trivariate and hypervariate. The types of visualisation that we can use to represent that

data also depends on the type of the data. What follows is a short resume about these types of data, for more details we recommend the work [5] of Cleveland. For every data type Cleveland discusses multiple usable visualisation types, with a lot of interesting examples and visualisation methods.

Single-variable or univariate data refers to a dataset where we are only observing a single variable. With this type of dataset we cannot show a relationship or compare between multiple variable because it only contains one variable. In univariate data we are more interested in finding a way to summarise information for this single variable. The description of the spread of the data is also a way that helps us to better understand the data. The spread of a data set includes the range, median, upper and lower quartiles, maximum value and minimum value (see section 3.1.7). The commonly used visualisations for univariate data are two dimensional charts like Pie charts, Line charts, Bar charts, Histogram and Box plots. A Box plot would graph the summary of all of the data together. When you want to compare the values with other values from the same variable then a Bar graph or a Pie graph can be helpful.

Double-variable or bivariate data is a data set that includes two variables, and is used to examine the relationship between these variables. In bivariate data you mostly want to show relationships or compare the variables. This type of data is commonly represented in two dimensional charts like a Scatter plot, Histograms and Area charts, but also in the visualisations used in univariate data. By creating a box plot of each variable we can easily compare the spread of the variable.

Trivariate data are data sets that have three variable that are easily representable in a three dimensional space. As you will see in section 3.3.2, some problem arises because we basically can only present data on paper or on a computer screen as a two dimensional representation of a three-dimensional space. Trivariate data can also be visualised on Bubble charts, TreeMap[1](Figure 3-9) and visualisations used in univariate and bivariate data by utilising additional dimensions. Note that these additional dimensions can be encoded as colour, item sizes, annotations, shapes, orientation and so on. More about this in section 3.3.2.

[1] `http://visual.ly/learn/treemaps` last accessed on 10/6/2014

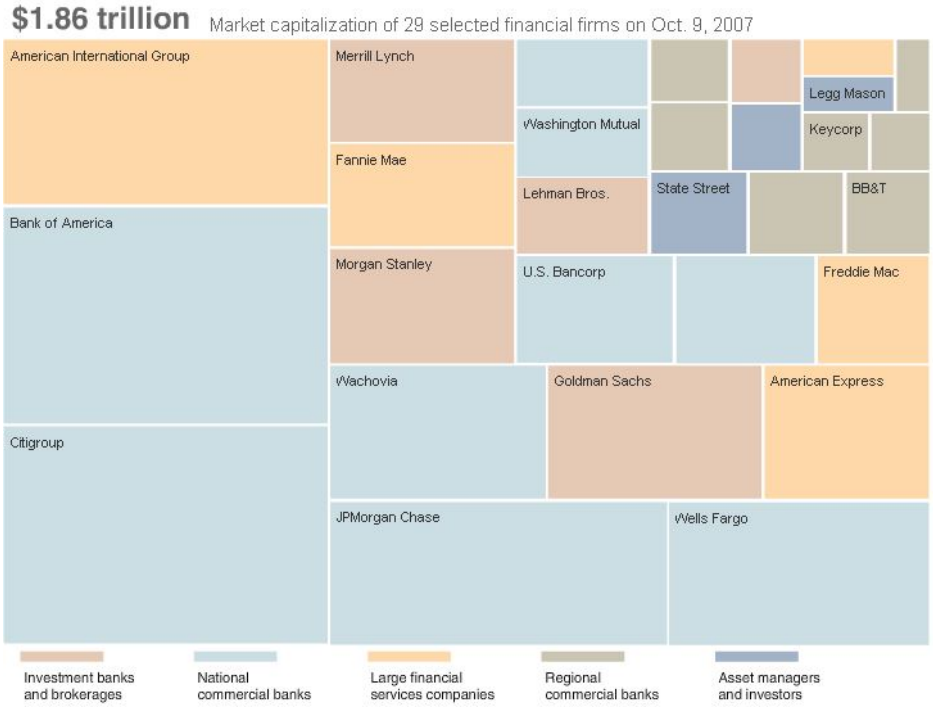


Figure 3-9: Each box represents the market value of that specific company, which is found by multiplying the number of a company's shares outstanding by its stock price (Source: New York Times)

Hypervariate data extends the idea to four or more variables in the dataset. As in trivariate data the challenge here is how to represent these multiple dimensions on a two dimensional screen or paper. Again a lot of the previous data types visualisations can be used, by adding extra dimensions or multiple axes. However, typical graphs used for visualising this type of data are Coordinate plots (Figure 3-12) and Star plots (also know as Radar/Spider plot: Figure 3-10). Another popular technique is interactive linking visualisations that each represent a variable of the dataset. Figure 3-11 is a interactive linked graph that was created by Antony Unwin [51]. The linking shows that the most students who were not good in Maths were also not good in other subjects, but that there are some with higher marks.

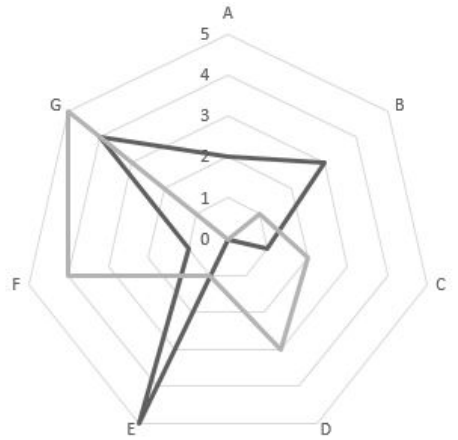


Figure 3-10: Star plot with 2 lines that represent objects and seven letters that could represent attributes of these objects.

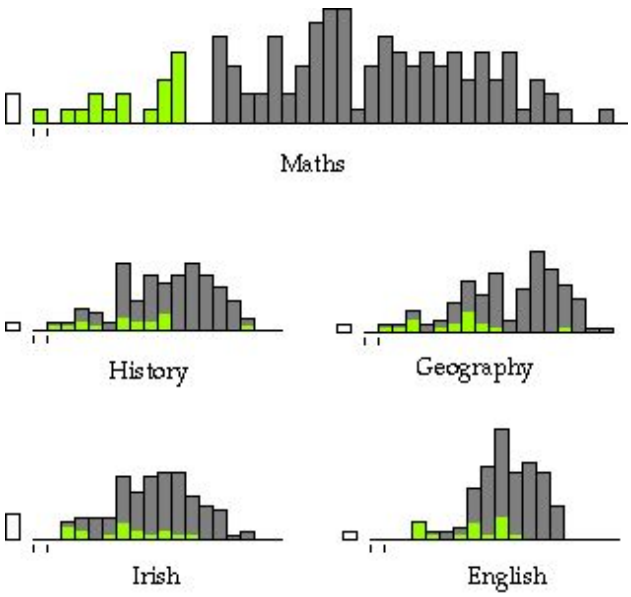


Figure 3-11: Linked histograms of the exam marks of 126 Irish pupils (Source: *Requirements for interactive graphics software for exploratory data analysis*, by Antony Unwin)

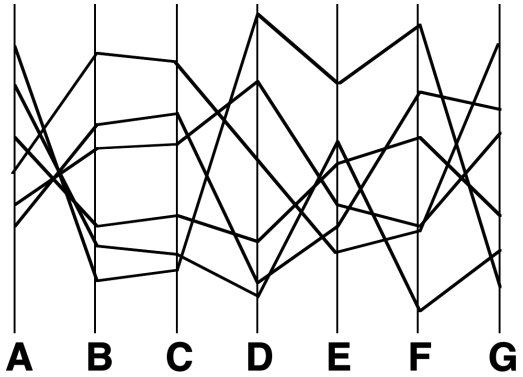


Figure 3-12: Coordinate plot with 6 lines that represent objects and seven letters that could represent attributes of these objects.

As you may have noticed, a data type with a high number of variables can use a type of visualisation for a lower number of variables most of the time. This can be done by adding addition dimensions, as explained above. However, the more number of variables the data type consist of, the harder it comes to represent. It is for example possible to visualise trivariate or hypervariate data with a line graph (Example for trivariate: Figure 3-13), but it can make it harder to analyse the graph. It is always better and more interesting to use the visualisation types that are introduce for that type of visualisation.

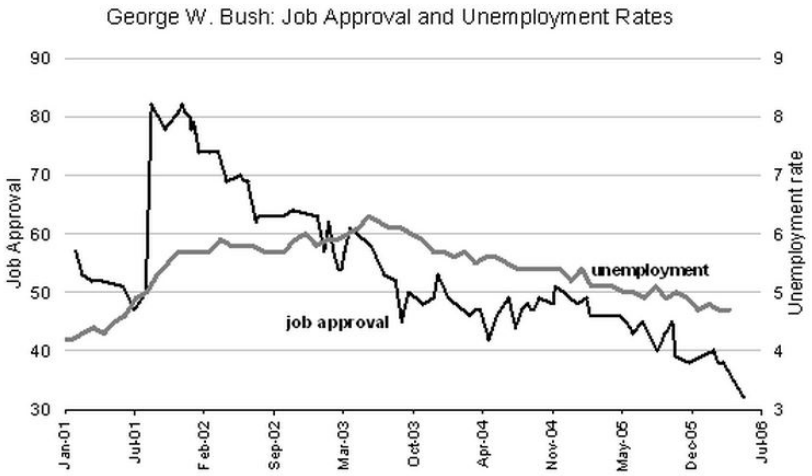


Figure 3-13: Source: Zogby International, unemployment: Bureau of Labor Statistics

3.2.2 Table or Graph to Communicate the Message

A small step that we need to consider is whether it is better to present the data in a table or in a graph. If the data will be used to look up and compare the individual values then it is in most cases better to use a table. It is also better to present the data in a table when it really needs to be precise. If the message is contained in the shape of the data by showing trends, patterns, exceptions or by comparing multiple values then it is better displayed in a graph [12]. The size of the data set also matters. Visualisations are usually outperformed by tables in reporting on small data sets of 20 values or less while visualisations come in handy when having larger data sets. [48]

It is also interesting to take into account the size of the visualisation in relation to the amount of data that will be visualised. For this Tufte uses the term *data density* which yield:

$$\text{data density} = \frac{\text{number of entries in data matrix}}{\text{area of data display}}$$

If there are not enough data points to have a higher data density, it may be better to use a simple table or find a way to shrink the visualisation. More information about data density and shrinking visualisations can be found in *The Visual Display of Quantitative Information* [48] written by Tufte.

3.2.3 Relationship that Best Supports the Message

In this section we will examine seven common relationships we want to show with our visualisation. Every type of relationship has some potential visualisation type that will support the message best.

Comparison
This relationship type means that you want to compare one set of value(s) with another set. An example of this relation type is comparing the performance between Product A and Product B.

The visualisation types that works best with comparing variables are the Horizontal bar graph and Vertical bar graph. A bar graph is good for comparing increases or decreases in values, highest and lowest value/frequency, how many of the same value/frequency. A Single-series bar chart is good for comparing values within a data category, such as monthly sales of a single product. A multi-series or Grouped bar chart is good for comparing categories of data, such as monthly sales for several products.

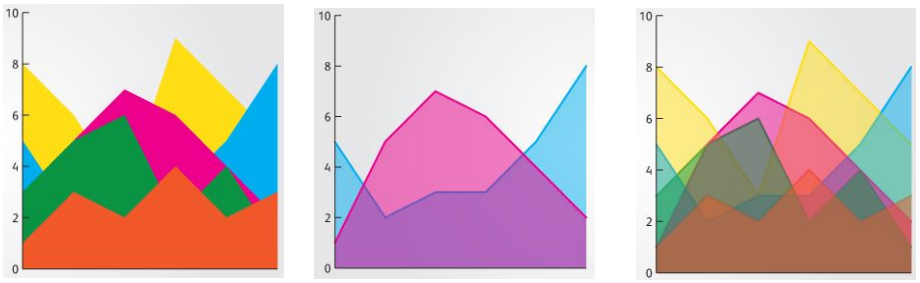
Another popular but not the best visualisation type for comparing variables is the pie graph. Because it is difficult to compare values, represented by slices, within a pie graphs or to compare data between multiple pie graphs, pie graph are commonly used when a general comparison is all that is required. Pie graph are most effective when at least most of the slices represent 25% to 50% of the whole. The human eye is not naturally skilled in comparing angles [9] so when you want to show more than 3 or 4 values, it is better to use a Bar graph. When the Bar graph is clouded with to much bars, you can also replace the bars with dots.

When comparing bivariate data it is interesting to use Scatter plots, which makes it one of the simplest visualisation type for comparing two different variable. When you want to compare trivariate or hypervariate data a Scatter plot matrix can be used, which is a matrix filled with Scatter plots that compares two variable each. With a Bubble plot it is also possible to compare data with more than two variable.

Trend over time
The purpose of trend over time, also called Time series, is to help understand the trend or flow of some variable(s) over time. A popular example in time series is the daily closing value of the Dow Jones Industrial Average (also called the Dow 30). When we search the Dow Jones Industrial Average on Google finance we get the visualisation displayed in Figure 3-14. In this visualisation we have a graph that gives a overview of the trends over all the years and another graph that gives a detailed view with zooming functionality. By zooming in the detailed view we can see the data from one day to all the years that are available.

The visualisations in Figure 3-14 are both Area graphs. Area graphs are based on the line chart except that the area below the plotted line is filled with colour to indicate volume. Area charts are not as good at showing the trend for many variables due to the filled area below the plotted lines which can obscure other data series that are on the same chart, as displayed in Figure 3-15a. Transparency works well with two or three series(Figure 3-15b), but if used with too many series, it becomes unclear(Figure 3-15c).

Figure 3-14: Dow Jones Industrial AverageSource (Google Finance, `https://www.google.com/finance?cid=983582`)



(a) Five Data Series (b) Two Transparent Data Series (c) Five transparent Data Series

Figure 3-15: Area Graphs, Source: Drew Skau (`http://blog.visual.ly/line-vs-area-charts/`)

When we want to show multiple series it is more interesting to use a Line graph. As in Area graph, we use the lines to emphasise the overall shapes of the series. By placing points connected by lines we can also emphasise individual values while still highlighting the overall shapes of the series. A (grouped) Bar or Column chart can also be used to present one or multiple series, if the number of bars needed to visualise the data is not too high.

Part-to-Whole

In Part-to-whole or Parts of Whole relationships we want to show how various parts comprise the whole. An example of this type of relationship is various percentages of religions amongst all the students of a school.

Pie graphs are commonly used to visualise part-to-whole relationships, but they do not work nearly as well as Bar graphs. It is much harder to compare the sizes or degree of the slices than the length of bars. For this type of relationship a Horizontal or Vertical bar graph can be used. Stacked bar graphs can also be used, but only when we must display measurements of the whole as well as the parts.

Another visualisation type frequently used in the Part-to-Whole relationship is the Treemap. In a Treemap you can compare any two parts as well as between parts and groups. For example, in Figure 3-9 we can see that the market value of the Bank of America is greater than the entire Regional commercial banks. With Treemap it is really easy and fast to compare values just by looking to the visualisation. However, Treemaps are not easy to get right. In contrast to basic charts where some researchers like Edward Tufte and Stephen Few have laid down some basic rules, Treemaps obey few of these rules and breaks many of them. As a result, many Treemaps contribute to the information visualisations lawlessness. In this document we will not go into details of Treemaps, but we would like to refer you to the work of Shneiderman [41].

Distribution

With the Distribution relationship we want to study the distribution of one or more categories of measurements of a variable. To be more precise, we want to study where the data for each category lies along the measurement scale. The measurement scale refers to how variables are measured, for example nominal, ordinal, interval and ratio scale of measurement. People who are interested in reading more about these measurements scales should read the work of S. S. Stevens [46].

The Histogram is a popular visualisation tool for showing distribution by summarise discrete and continuous data that are measured on an interval scale. A histogram divides up the range of possible values of a data set into groups that are represented by bars. A Histogram makes it possible to detect outliers or gaps in the data. A Histogram does a good job of showing us individual distributions of data sets, but they are not the best pick when we want to do a comparison of distributions. Sometimes a Histogram also has a line connecting all the midpoints of the top of the bars. In that case the visualisation is called a Frequency polygon. A Frequency polygon gives an idea about the shape of the data distribution.

When you just want to show a summary of the distribution of the data then you can use a Box plot. Box plots are an excellent way to compare multiple distributions because they make it possible to quickly compare the summary of the distributions. Also a box plot can be used even when the number of distributions are to small.

Deviation

When data is visualised to show that one or more sets of values differ from some reference set or norm, then we are dealing with the deviation relationship. These values can be visualised with bars, lines and points while the reference set is always visualised with a reference line or axis. To explain this type of relationship Stephen Few [12] uses a general example in business where a visualisation (Figure 3-16) shows how some set of actuals (e.g. expenses) deviate from a predetermined target (e.g. budget).

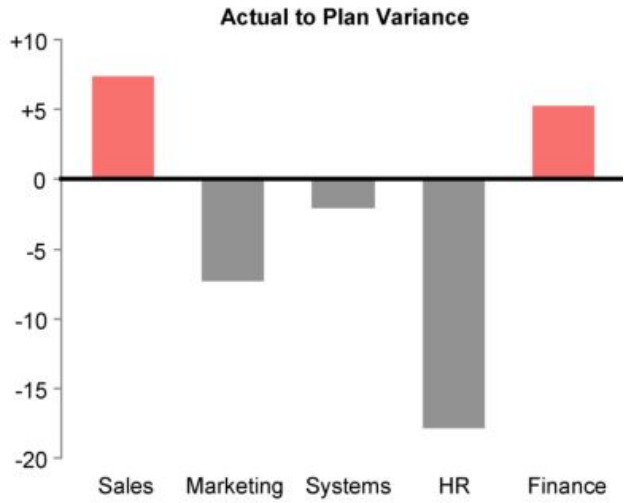
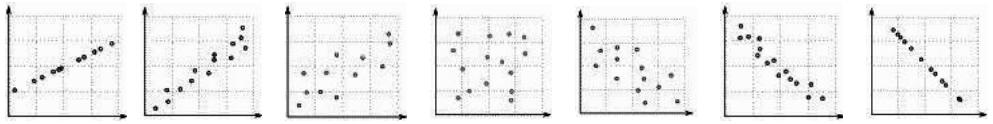


Figure 3-16: Example deviation (Source: *Effective Communicating Numbers*, by Stephen Few)

In Figure 3-16 you can see a Bar chart used by Stephen Few to visualise a deviation, which are mostly used in this relationship type for emphasising individual values. When the overall pattern has to be emphasised it is more clear to use a Line graph with value lines, which are visualised differently from the reference line. Points that are connected with lines in a Line graph can slightly emphasise individual values while also highlighting the overall pattern.

Correlation

There are visualisations that work particularly well for comparing paired variables of one or more sets of measurements to determine if there is a link between these variables. By visualising correlation between sets you can easily see if there is a positive or negative correlation, a perfect or low correlation or even no correlation at all (illustrated in Figure 3-17). Understanding the correlations between variables can help predict and avoid or even take advantage of a particular situations. In for example car insurance, they correlate age of drivers with accidents, so insurance companies know which age to charge more.



(a) Perfect Positive Correlation (b) High Positive Correlation (c) Low Positive Correlation (d) No Correlation (e) Low Negative Correlation (f) High Negative Correlation (g) Perfect Negative Correlation

Figure 3-17: From positive to negative correlation

Most of the time a Scatter plot is used to visualise a correlation, by presenting each value as a point. Sometimes a linear trend line is added, which can show the overall direction of the values. It is also interesting to use a Scatter plot when you want to search for outliers in the data sets that your are comparing. These outliers are points way off to the side from the linear trend line of the other points, making it easy to detect them.

Bar graphs can also be used to present correlation between pairs of sets. A Bar graphs will take more space than a Scatter plot and presenting to much values on this visualisation type can lower the readability of it. When working with Bar graphs it is best to arrange the bars as a Paired Bar graph or a Correlation Bar graph [15].

Ranked
In a ranked relationship the (mostly categorical) values are sequenced by size, from large to small or vice versa. It is often interesting in businesses to see things ranked, such as the the expenses of departments or the sales number of products. This relationship type does not only give a easy overview of the sequence, but also makes it easier to compare the values by placing those that are most similar near one another.

Sometimes people use a Line graph to visualise the rank of the values, which is not the best idea. A Line graph is used when the overall pattern has to be emphasised, however in a ranked relationship it is the individual value that has to be emphasised and not the overall pattern. When the focus has to be on the values it is better to only use bars, vertical or horizontal. To highlight the high values, it is best to sort the bars in descending order. When you want to highlight the lower values, then the bars are better sorted ascending.

3.2.4 Conclusion

By understanding the different data types and the seven types of relationships together with the chart types that present them most effectively, we already passed the first big step in knowing the best means to present data. Knowing how to design the separate components of a graph to communicate your message clearly, without distraction and without damaging the integrity of the data is another big step, which we'll examine in the next sections.

3.3 Graphical Integrity

A visualisation has graphical integrity if it accurately represents the data, so when the visualisation is consistent with the numerical representation. This is less simple than it seems. In chapter section 2.4.2 we already saw some examples of deceptive graphs, and examples of poor graphical integrity abound. Poor graphical integrity can be caused by the designers who chooses to give readers the impression of better results than is actually the situation. Of course not all designers are dishonest, some may create visualisations that mislead the reader by a poor choice of graph type or poor graph construction [40].

It is possible to mislead the reader with any means of communication, so also by representing the data as visualisation. This means that misinformation and disinformation are not unknown terms in the field of information

visualisation. When we follow the definition of the Oxford English Dictionary[1] then we denote misinformation as "wrong or misleading information" and disinformation as "the dissemination of deliberately false information". However, as Bernd Carsten Stahl explained so well in his essay *On the Difference or Equality of Information, Misinformation, and Disinformation: A Critical Research Perspective* [45] is that the distinction of information, misinformation, and disinformation is problematic. As Bernd states on information, misinformation and disinformation:

> "*The most important distinction between information and misinformation and dis-information is the question of truth. Where information is true, misinformation or disinformation are untrue. The distinction is closely linked to the question of truth and we should admit that there is no universally accepted theory of truth.*"

For simplicity's sake, we continue to distinguish between misinformation as accidental deception and disinformation as deliberate deception. Further in this section we will first discuss why people draw visualisations that deceive and why newspapers, magazines or even scientific books publish these deceptive visualisations. Then we will take a look at the most common ways in which visualisations can be misleading. The main source of this section will be one of the biggest contributor in graphical integrity named Edward R. Tufte [48, 49].

3.3.1 Reasons for Poor Graphical Integrity

One of the biggest reasons for poor graphical integrity is the fact that most people who are responsible for producing visualisations for publications lack the proper training. Even if a person is trained in graphic design does not mean that he or she will develop a visualisation with good graphical integrity. Skills and experience in analysis of data is also very important for graphical integrity. Data is useless until we understand what it means and can communicate that meaning clearly and possibly visually to those who require it. Unfortunately it is nowadays more important to visualise your data as beautiful as possible without ever minding the statistical integrity of the visualisation. What we need are graphic designers with experience and skills in analysing data and also know how to visualise this data without loss of data integrity.

[1] `http://www.oed.com` last accessed on 30/06/2014

The second reason is the widespread belief that anything associated with statistics is boring. Also graphic artists know that statistics are usually not a popular topic so they try to make it more interesting and fun by creating (over)decorated visualisations of the information available in there data. Due to people that keep believing in the stereotype that statistics is boring, they are encouraged to let a artist create the visualisation of the data instead of the persons who gathered or/and analysed the data. The message that the analyst wants to show with this data can get lost when visualised by another person.

Another common reason for the poor graphical integrity in some visualisations is that many believe that visualisations should divert and please those in the audience who find the explaining text of the data too difficult. Especially in the early years of information visualisation it was a common belief that visualisations are only for the unsophisticated readers.

In all three reasons above we have cases of misinformation because it is not the purpose of the creator to deliberately deceive the viewer. The reasons above comprises the skills, attitudes and organisational structures prevailing among the creator of the visualisations. Not only misinformation can create a poor graphical integrity, creating disinformation by intentionally spreading false or incorrect information will also have a negative effect on the graphical integrity of the visualisation. There are many reasons why someone would lie in a visualisation, but most of the time it is to give readers the impression of better results than is actually the situations. When you want to read more about how to intentionally spread false or incorrect information with visualisation we recommend the book *How to Lie with Statistics* [25] written by Darrell Huff.

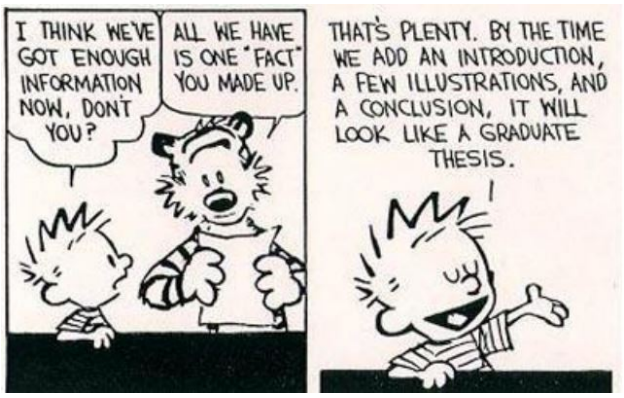


Figure 3-18: Comic Calvin and Hobbes

3.3.2 Tell the Difference

In this section we will discuss some principles of graphical integrity and common ways in which visualisations can be misleading. In this discussion we assume that the data of the dataset aren't altered by the creator. The only thing the creator uses are some visualisation techniques to (deliberately) deceit the viewer.

Perceptions of Area

Some experiments have discovered that perceptions of area versus the actual area can vary per person. James Flannery's research in *The relative effectiveness of some common graduated point symbols in the presentation of quantitative data* [16] on the human perception of circles and other symbols on statistical visualisations tells us that while the circles was scaled accurately, the users misperceive the size of the areas and so also the value linked with it. Later on Flannery developed a method that uses an appearance compensation to the circles to compensate the underestimation (illustrated in Figure 3-19). The problem with this compensation is if one actually measures the compensated perceptual circle one would get the wrong actual values, which makes it a lying visualisation. This overcompensation is also based on an average and does not take into account the viewers that do not have problems to deduce the correct value from the circles.

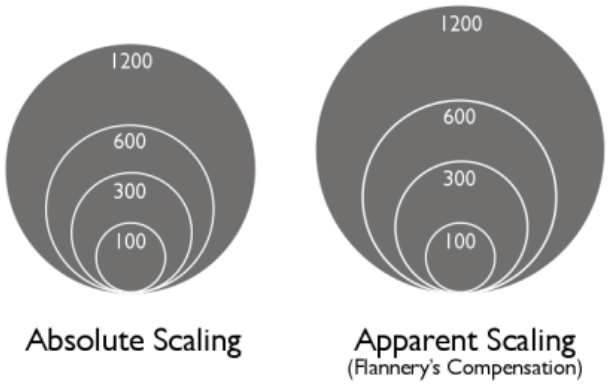


Figure 3-19: Flannery's Appearance Compensations

Edward Tufte stands against the Flannery's compensation and states his principle: "The representation of numbers, as physically measured on the surface of the graphic itself, should be directly proportional to the numerical quantities represented" [48]. Violating this principle means, according to Tufte, distortion in the visualisation. To test if a visualisation does not violate his principle he created a new measurement called *Lie Factor*.

$$Lie factor = \frac{\text{size of effect shown in graphs}}{\text{size of effect in data}}$$

If the Lie Factor is equal to one, then the visualisation is representing the data correctly. Lie Factors greater than 1.05 or less than 0.95 are distorted and can visualise misleading information. Figure 3-20 is an example used by Tufte to explain his principle. On the visualisation we can see an increase of 53% in the fuel economy standards for cars:

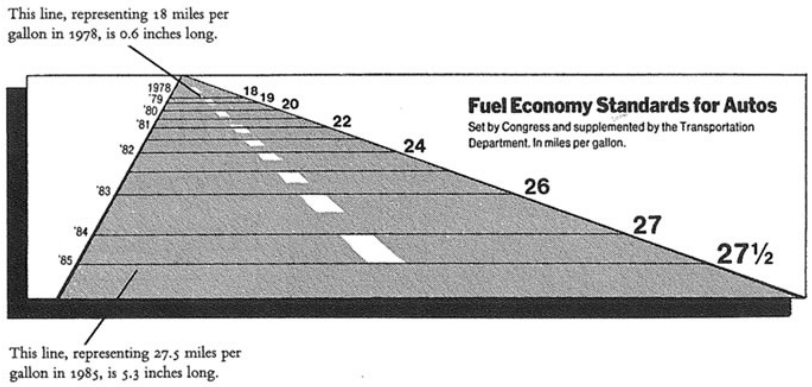


Figure 3-20: Deceptive visualisation (Source: New York Times, August 9, 1978)

$$\frac{27.5 - 18}{18} \times 100 = 53\%$$

The magnitude of the sizes of the horizontal lines between the lines of '78 and '85 is 783%:

$$\frac{5.3 - 0.6}{0.6} \times 100 = 783\%$$

When we now use the formula of Tufte we see a Lie Factor of 14.8. This value is much higher then one, which mean that this visualisation is an extreme case of misinformation or even disinformation.

$$Lie factor = \frac{783}{53} = 14.8$$

Design and Data Variation

When a viewer analyses a part of a visualisation he or she also unconsciously generates a visual expectation about the other parts of the visualisations. When the visualisation is not fully consistent it is possible that the created expectation can create some form of deception. So the problem with not using a consistent design throughout the whole visualisation is that changes in the design can cause viewers to confuse this change with actual data change. Figure 3-21 is a small example that contains one design variation. At first sight the x-scale is moving in a regular interval of 10 years, which is true between 1991 and 2011. A viewer could generate a visual expectation that after 2011 the interval is still 10 years. As consequence is that the viewer will then perceive that there is a slowing increase after 2011. However when taking a closer look we see that after 2011 the interval changed to one year and stops at 2014, which means that we don't have enough information to see if we really have a slowing increase over the whole line.

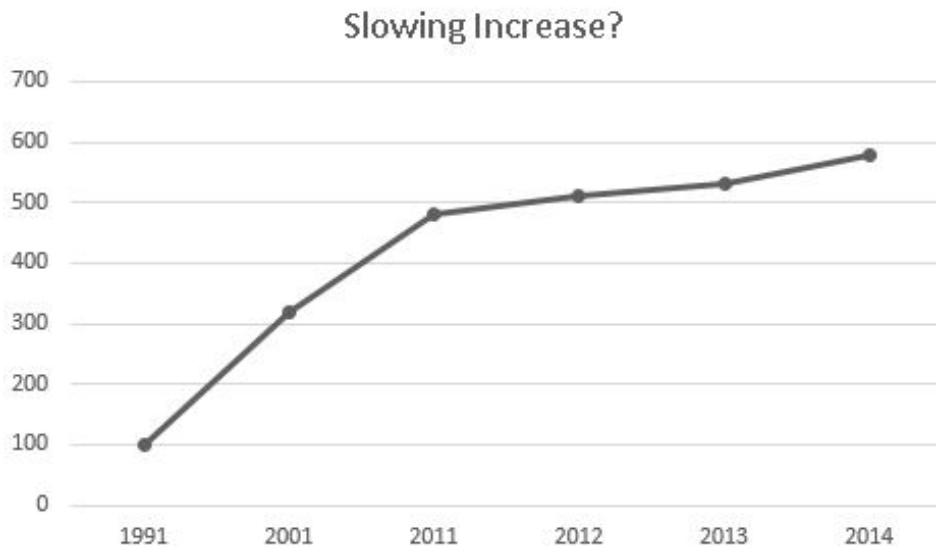


Figure 3-21: Design Variation

When including design variation in the visualisation the viewer may mix up changes in the design with changes in the data. To avoid this type of deception it is also best to follow a principle defined by Edward Tufte: "Show data variation, not design variation." [48].

Dimensions
The use of two or more varying dimensions to show data with a lower number of dimensions is a weak, deceptive and inefficient technique to visualise the data. They are only capable of handling very small data sets and the more dimensions is used to view a variable, the more error in design and ambiguity in perception you will possibly have. Figure 2-16 is an example of a deceptive and inefficient 3D visualisation that exceeds the number of dimensions in the data, which is in this case two dimensions.

That is why it is also interesting to follow the principle stated by Tufte: "The number of information-carrying (variable) dimensions depicted should not exceed the number of dimensions in the data." [48] This principle also includes the fact that you can visualise the number of dimensions in data by a lower number of information-carrying dimensions. In section 2.4.1 we already saw an alternative representation for a 3D visualisation formed as three possible two-dimensional views of the data, show in the scatterplot matrix in Figure 2-10.

Scales
One of the most common ways to (deliberately) misrepresent data is by choosing a particular deceptive scale for the axis. Small differences in scales to present the data can have huge visualisation and perception differences representing that data. Too many mistakes have been made in selecting the correct scale because of the sometimes subtle but also difficult scale issues. Let's take a closer look into these issues:

Let's start with a principle stated by Cleveland: "Choose the range of the tick marks to include or nearly include the range of the data" [6]. The range of the data is the interval from the minimum to the maximum of a set of values in a dataset. Following Cleveland's principle it is a good idea to place the full range of the data on the visualisation included or nearly included in the range of the extreme tick marks. In Figure 3-22 the whole range of the data on the horizontal scale is completely contained in the range of the tick marks while the data on the vertical scale are nearly contained into the range. With this principle Cleveland also tries to state that it is best to visualise the full range of the data. By omitting data from the visualisation some information can be lost which can change what the viewer conclude from the visualisation. This problem is already illustrated in section section 2.4.2 with Figure 2-14 as the visualisation that omitted data and Figure 2-15 that shows the full rage of data. Tufte created for this problem also a principle: "Graphics must not quote data out of context" [48].

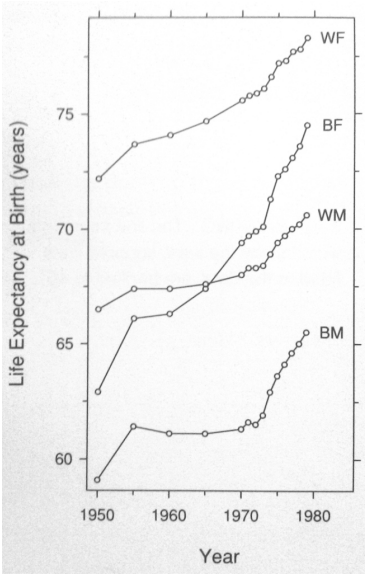


Figure 3-22: Ranges (Source: *The Elements of Graphing Data*, by Cleveland)

Figure 3-22 is an example used by Cleveland to show how to pick the scale for the axis. What can be noticed on that visualisation is that the vertical axis does not include zero on the scale, or in other words does not have a baseline. Cleveland has another principle that goes as follows: "Do not insist that zero always be included on a scale showing magnitude". Cleveland is of the opinion that it is helpful to include zero in the scale to see its value relative to the value of the data, but it should not ruin the possibility to see the variation in the data. Following example are visualisations created by Ravi Parikh [34] that both show the same date. The only difference is that the y-axis of the left visualisation starts from 3.14% and ends on 3.154% while the right one starts at 0% and ends at 3.50%.

Cleveland prefers, in the example below, the left visualisation because it shows a better variations in the data. He is of opinion that viewers will look at the tick mark labels and they will also have it more easy to read the values. The visualisation on the right is, according to Cleveland, a waste of space because the small difference in height between the bars makes it impossible to read the data properly which makes it difficult to see the variation in the data.

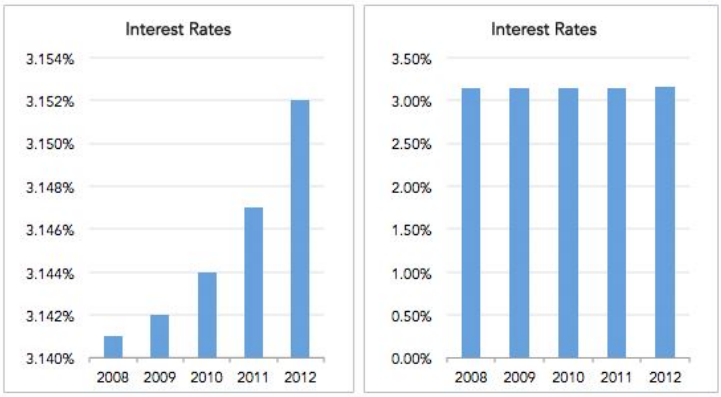


Figure 3-23: Same data on different scales (Source: *How to lie with data visualisation*, by Ravi Parikh)

Tufte follows almost the same reasoning as Cleveland. He states that when the zero values occurs in the data then you definitely need a baseline. Otherwise don't spend a lot of empty vertical space trying to add a baseline at the cost of hiding the variation between the visualised data.

Other researchers do not share the same opinion as Cleveland and Tufte in this topic. For example Darrell Huff, who wrote the book *How to Lie with statistics* [25], states that a visualisation that shows magnitudes without a zero line is dishonest. Viewers of a visualisations sometimes do not look at the tick mark labels and will form a conclusion by only looking to the variation between the visualised data.

Stephen Few states that adding a baseline depend on the type of visualisation that is used. If bars are used to visualise the data then they must start from zero. However, if lines, points or combination of line and points are used the scale should be adjusted so that it extends a little below the minimum value and a little above the maximum value [12]. If for example bars are used to visualise the data, but the message of the data is not clear because of the large scale, then the bars can be replaced by points and the scale can be narrowed down.

With Figure 2-13 we have seen another way of how easy it is to let the viewer misperceive the information shown in the visualisation. In this example they placed the vertical axis upside down which gives you the first impression that the number of murders committed using firearms is mostly descending since 2005. Afterwards you notice that it is not descending but is actually

ascending. Most people are not familiar with a baseline at top of the chart with positive values underneath which can make it confusing. In this case the reverse arrangement is probably to achieve the metaphor of blood effect.

3.3.3 Conclusion

We saw a few reasons under which data visualisations are produced that have a lower graphical integrity. The lack of skills and experience in analysing data and graphic design together with the widespread belief that anything associated with statistics is boring, leads to misinformation. Spreading false or incorrect information to give a viewer the impression of better results than is actually the situations leads to disinformation. These reasons produce visualisations that (1) lie; (2) are overdecorated and have simplistic design; (3) do not present the real information in the data.

In this section we saw some examples and principles that help with achieving high graphical integrity. Of course there is more information available about how to avoid or detect misleading visualisations. More principles and examples for a good graphical integrity can be found in *How to Lie with Statistics* [25], *The Visual Display of Quantitative Information* [48] and *The Elements of Graphing Data* [6]. In this section we picked the examples and principles that apply on the more common visualisations created nowadays.

3.4 Design Aesthestics

In previous sections we have already seen some guidelines and principles to pick the chart that represents the data best and how to avoid creating misleading visualisations. Now we will take a closer look on how to make sure that the visualisation draws the users' attention to the message included in the data and not to something else, for example colourful backgrounds, a design flaw presented in Figure 2-19a.

We start this section by demonstrating the power of visual perception and how to use this power to create better visualisations. Then we will talk about the principles of grouping and how we perceive the world as combinations of visual elements. We will also see five popular principles created by Tufte about non-data-ink in visualisations. Next we talk about the use of colours, legends, tick mark and grid lines. To finish this section we also briefly discuss the design steps for creating clear and comprehensive tables.

3.4.1 Visual Perception

Vision is one of the fastest and most used sense of a human, and is also one of the senses that is the closest connected with cognition [14]. By seeing and thinking we make sense of the world. So with the power of visual perception we can explore, understand and present information. One of the leading researcher in the power of visual perception explains in his book *Information Visualisation: Perception for Design* [53] the importance of visual perception and visualisation:

> "*Why should we be interested in visualisation? Because the human visual system is a pattern seeker of enormous power and subtlety. The eye and the visual cortex of the brain form a massively parallel processor that provides the highest-bandwidth channel into human cognitive centers. At higher levels of processing, perception and cognition are closely interrelated, which is the reason why the words 'understanding' and 'seeing' are synonymous. However, the visual system has its own rules. We can easily see patterns presented in certain ways, but if they are presented in other ways, they become invisible... The more general point is that when data is presented in certain ways, the patterns can be readily perceived. If we can understand how perception works, our knowledge can be translated into rules for displaying information. Following perception based rules, we can present our data in such a way that the important and informative patterns stand out. If we disobey the rules, our data will be incomprehensible or misleading.*"
> Colin Ware

Before we can visualise information effectively, we must first know how to tap into the power of visual perception. However, what is actually meant with the power of visual perception? The next example will help that make clear. The table below (Figure 3-24) shows the data in a text-based manner, which makes it easy to find a precise value. However, with visual perception we go beyond looking up precise values in a table like below. Information that is presented visually allows us to gain insights more easily than would be possible from the same data textually presented.

Sales Revenu
(U.S. dollars in thousands)

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Domestic	1985	2433	2684	2215	2548	2868	2384	2654	2698	2487	2984	3458
International	587	636	675	594	641	679	596	134	300	501	608	658
	2572	3069	3359	2809	3189	3547	2980	2788	2998	2988	3592	4116

Figure 3-24: Table of fictional Sales Revenue

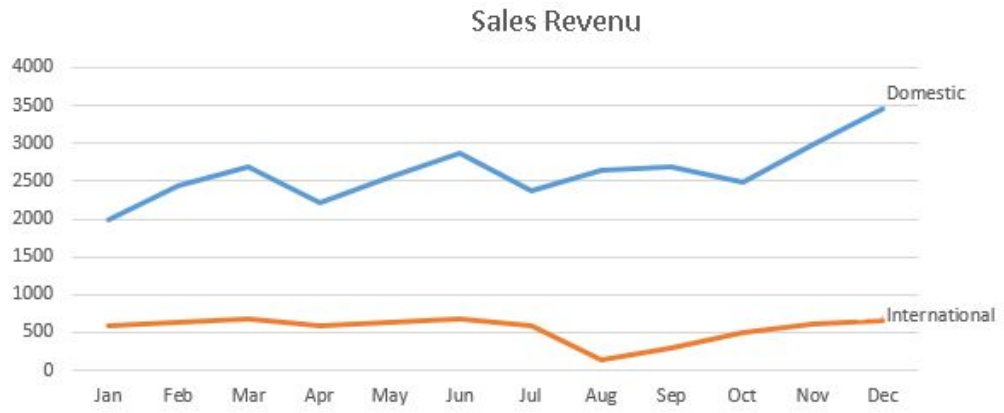


Figure 3-25: Visualisation data in table principlesofgraphingdata/tablePowerPerception

The visualisation in Figure 3-25 presents the same data as the table above but brings several new facts that were not obvious to notice using the table. What follows are some facts that the visualisations makes visible instantly:

- Throughout the year the domestic sales were always much higher than international sales.
- International sales remain relatively the same while domestic sales go upwards during the year.
- August was a bad month for the international sales but relatively good for the domestic sales.

The example above shows that it is more easy with visualisations to see patterns and relationships. Of course it is also possible to become the same conclusions as above with a table, but mostly not as efficient as a visualisation.

A lot of knowledge of perception has been gathered already and we should keep this knowledge in mind if we want to create effective information

visualisations. What follows are some interesting facts from Stephen Few [14] about how we collect and process visual information:

> Fact 1: We do not attend to everything that we see. Visual perception is selective, as it must be, for awareness of everything would overwhelm us. Our attention is often drawn to contrasts to the norm.

This means that we have to create our visualisations in such a way that allows what is interesting and most meaningful to stand out from what's less interesting. Take for example Figure 3-26, which is an adapted picture from a book *Where's Waldo* [21] written by Illustrator Martin Handford. The *Where's Waldo* books challenges readers by showing pictures on where they have to find the cartoon man with the red-striped shirt and red-striped hat. Most of the time it is really challenging because to find Waldo you need to scan the page completely with your eyes.

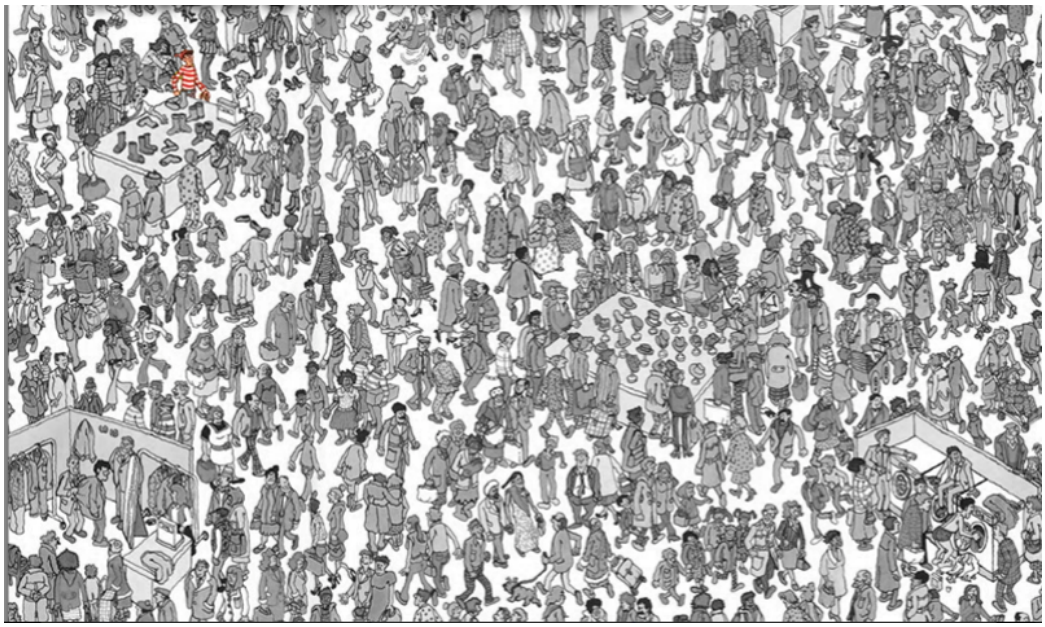


Figure 3-26: Where's Waldo (Source: *Where's Waldo*, by Martin Handford)

You probably found Waldo in an instance (except if you are colour blind). Waldo stands out because he differs from his surrounding, which is only printed in black and white while Waldo himself also has the colours pink and red. What's important to remember here is that information visualisations should stand out the potentially meaningful information in contrast to what is less or even not worth the readers attention.

> Fact 2: Our eyes are drawn to familiar patterns. We see what we know and expect.

Let us explain this fact with an example. When looking at Figure 3-27 our eyes recognise almost instantly the shape of a rose. Afterwards this shape/pattern is also categorised by our minds as a rose. However, the image of the rose also contains another familiar shape, which is not easily noticeable. When you did not see it yet, take a few more seconds to see if you can spot this other familiar shape.

Figure 3-27: Rose with another familiar shape (Source: `www.coolbubble.com`)

Did you spot the shape of a dolphin? It is located above the middle of the rose. With this fact Stephen Few tells us that using familiar and easy to spot patterns to present the information in a visualisation is one of the best techniques.

> Fact 3: Memory plays an important role in human cognition, but working memory is extremely limited.

This fact tells us that we have to keep in mind when we want to remember an object, or if we want to see an object change, that it is necessary to attend to it. We must remember that visual perception does not capture as much of the world as we may think. This problem is know as change blindness [42], the inability to detect changes in photographs, motion pictures and even in real-world environments. Note, when we do not focus on something, we do not see it clearly because only a small part of each eye called the fovea is designed for high-resolution vision. This fact tells us that information visualisation must help the working memory with processing the displayed information.

With these facts it is clear that when we want to create effective information visualisation we have to know and respect how visual perception and cognition works. Jacques Bertin was the first person to recognise and create a basic vocabulary of vision in his work *Semiologie graphique* [2]. Much of Bertin's work is based on creating an understanding of the basic attributes of visual perception. He discovered that some of these basic attributes are perceived *pre-attentive*, which means that these attributes, for example certain shapes or colours, pop out from their surrounding.

By using pre-attentive attributes we can encode information visually in a way that can be perceived easily and instantly by the viewer. And, if some of the data should stand out from the rest it can be encoded by using different pre-attentive attribute(s). Figure 3-28 [11] shows a list of more common pre-attentive attributes that can be used to visualise data, created by Colin Ware [53] for demonstrating the power of visual perception.

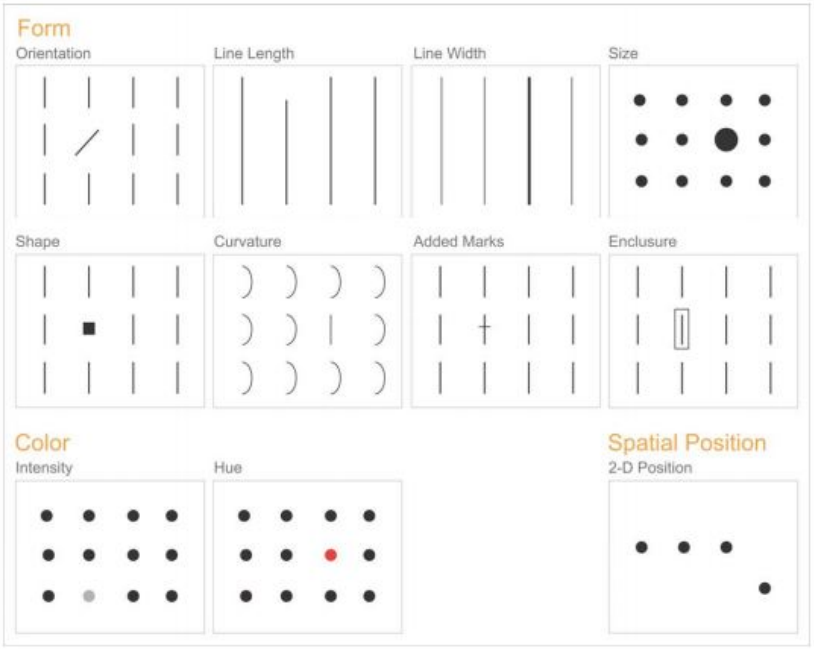


Figure 3-28: Pre-attentive attributes of visual perception most applicable in data visualisation (Source: *Tapping the power of visual perception*, by Stephen Few)

> "*An understanding of what is processed pre-attentively is probably the most important contribution that vision science can make to data visualisation.* "
> Colin Ware

Not all these pre-attentive attributes are perceptually equal. Some attributes are perceptually strong in showing quantitative data while weak in categorical data. Some can be perceived quantitatively, which makes them useful for encoding numeric values, while others are not. For example variation in colour hue cannot be used to show quantitative values. To summarise this, Mackinlay [32] created a ranking of pre-attentive attributes for encoding quantitative, ordinal and categorical data. These rankings are shown in Figure 3-29 with at the top the most accurate and at the bottom the least accurate attributes for that data type.

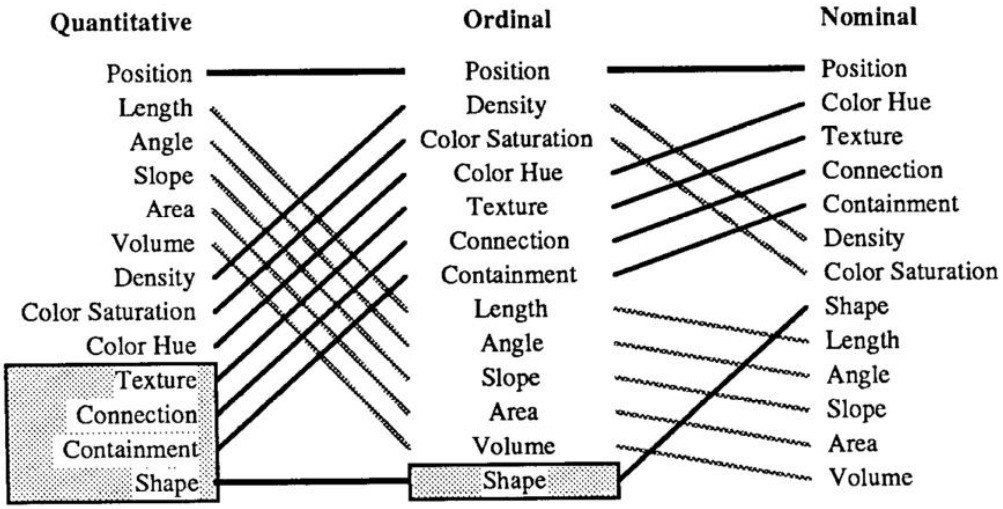
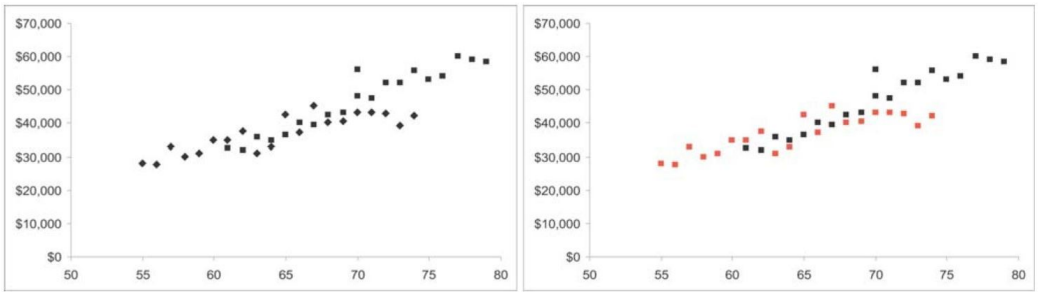


Figure 3-29: Mackinlay's ranking of pre-attentive attributes. Attributes in the gray boxes are not relevant to these types of data. (Source: *Automating the design of graphical presentations of relational information*, by Mackinlay)

In Figure 3-30 we see two visualisations created by Stephen Few [11]. These two Scatter plots are presenting the same data. The only difference is the pre-attentive attribute that they use to differentiate the two sets of data. Which pre-attentive attribute does the best job of grouping the two sets of data? They both work, but hue is doing a much better job.

Fact one and two can be anticipated by keeping in mind the pre-attentive processing of a visualisation, but we still have to overcome the limits of the working memory. Research has shown that our visual working memory can only hold three storage units, better known as chunks, of information at a time [7]. How much visual information that can be contained in a chunk depends on how this information is visualised. When we for example take



(a) pre-attentive attributes: orientation (b) pre-attentive attributes: hue

Figure 3-30: Comparison of the relative strength of two pre-attentive attributes. (Source: *Tapping the power of visual perception*, by Stephen Few)

the table from Figure 3-24 then most of us would need to store each number as a separate chunk. This means that we cannot store much information from the presented data in the table into our working memory. However, when we take the visualisation in Figure 3-25 where the same information is displayed in a line graph, we can store each line in a single chunk, one for the domestic sales and one for the international sales. This means that with a visualisation we can think about more information simultaneously than we were able when relying on a table. This is why visualisation makes it more easy to explore and analyse data.

3.4.2 Principles of Grouping

In this section we will briefly go trough the principles of grouping. These principles are also know as Gestalt laws of perceptual grouping. The Gestalt psychology, founded in Germany, shows that we perceive the world as combinations or group of elements that are different than a collection of elements in isolation. Figure 3-31 is an interesting illustration of several grouping principles, created by the writers of *A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure-ground organisation.* [52].

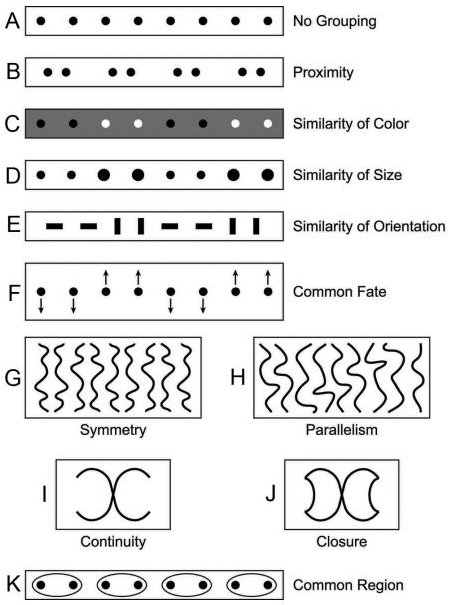


Figure 3-31: Illustration of several grouping principles (Source: *A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure-ground organisation*, by Wagemans et al.

In the first box (A) they demonstrate that equally spaced dots are not grouped together by our mind. However when placing some dots closer then other dots (B), our perception tends to group close together dots as part of the same element. This is why people tend to see clusters of dots instead of a large number of individual dots. This principle of grouping is called *proximity*. This principle does not only has an effect on dots, but also on other shapes likes squares, triangles, lines etc.

The next principle of grouping is called *similarity*. The principle of similarity states that, if all else is equal, the most similar elements in for example colour (C), size (D), orientation(E) etc. tend to be grouped together. We can use the similarity principle to distinguish between elements who lay adjacent to or overlap with each other based on how the visualisation of the elements looks like.

Another grouping principles illustrated in the figure is *common fate*. In common fate, the only difference between elements is the direction they move in (F). When objects are seen moving in the same direction at the same rate they tend to be grouped together. Think about birds flying in the sky, which are seen as a flock when all flying in the same direction and the same speed.

Also more complex elements can be grouped together. For example the elements that are symmetric (G) or parallel (H), both principles of grouping. Another principle of grouping in *continuity* (I). When two or more elements are intersecting, we tend to perceive each element as a single uninterrupted element.

The grouping principle *closure* (J) refers to our mind having the tendency to see complete elements or forms even if it is incomplete. These elements or forms can be partially hidden behind other elements, or have some parts that are not drawn. The part missing that is needed to make a complete picture will be automatically completed by our mind.

The last grouping principle that is less know is *common region* (K). This principle states that elements that lie within the same bounded area will be grouped together. All else being equal, the elements being inside of or contained by some larger surrounding contour tend to be grouped together.

These were some of the more common principles of grouping that we also can encounter in information visualisation. There are of course some additional principles that we did not discuss here. If you want to know more about these principles, we advice you to read the paper *A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure-ground organisation.* [52].

3.4.3 Data-Ink and Graphical Redesign

> "*Above all else show the data.* "
> Edward Tufte

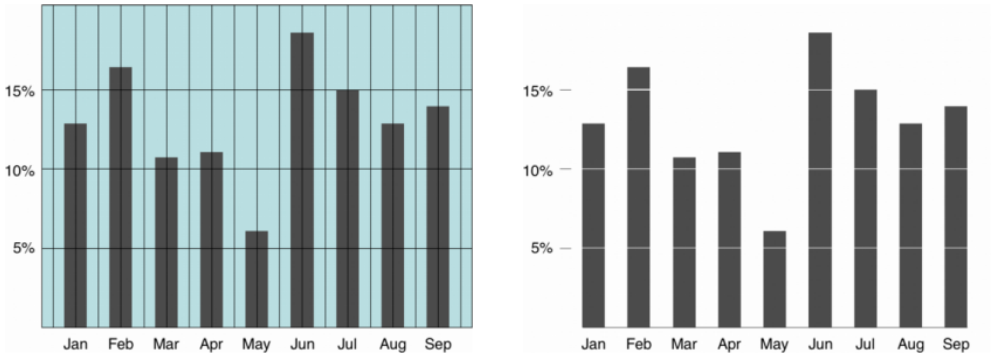
We started this section with one of the most fundamental principle for good statistical visualisations. This principle means that the visualisation of the data should draw the viewer's attention to the message contained in the data, not to something else. Most of the time spent on a visualisation has to go to presenting the data and not decorating the data that is presented in the visualisation. It is also important that the message is visualised efficiently and correct, which we discussed by going through some principles and techniques in section 3.1 and section 3.3.

Tufte also tells us that a good visualisation presents the message with a clean and simple design without extra decoration, or in other words without extra non-data details. However, what is a clean and simple design and how do you measure it? For this he created another four principles which will make these clean and simple design more concrete. We will shortly discuss these principles in this section. More information and examples can be found in Tufte's book *The visual display of quantitative information* [48]

To turn a clean and simple design into a more practical idea, Tufte defined *data-ink* and *data-ink ratio*. Data-ink is the non-erasable core of a graphic that consist of non-redundant ink arranged to represent the data. When a visualisation only contains data-ink, nothing can be removed without losing information. Data-ink ratio is defined as following:

$$\text{Data-ink ratio} = \frac{\text{data-ink}}{\text{total ink used to print the graphic}}$$

In other words data-ink ratio is the proportion of a visualisation's ink used for a non-redundant representation of the data. "Maximise the data-ink ratio, within reason" is one of the more general principles that will also motivate Tufte's remaining principles, due to the fact that they all follow from the idea of maximizing the share of data-ink. This principle tells us that the more ink of a visualisation is used to only present data, the more the data is emphasised, and the better the visualisation. Keep in mind, almost all graphics require some non-data ink, such as the axis lines with the tick marks, labels, legend etc. This is why you can only maximise the data-ink ratio within reason. Figure 3-32 are two visualisations presenting the same data but one has a low data-ink ratio while the other has a high data-ink ratio.



(a) Low Data-ink Ratio (b) High Data-ink Ratio

Figure 3-32: Low vs high data-ink ratio (Source: wikipedia)

The following principle of Tufte is more a erasing principle: "Erase non-data-ink, within reason". Ink that does not help to spread the message in a visualisation is not interesting for the reader and can in fact even clutter up the data. With saw an interesting example in Figure 2-19 created by Darkhorse in which they transform a visualisation with low data-ink ratio into a visualisation with high data-ink ratio. They improved the visualisation by removing non-data element like the background, redundant labels, special effects, bolding and changing some elements like reducing colours, lighten labels and grid lines. They even removed the grid lines an thick marks and replaced them with direct labels.

> "*Perfection is achieved not when there is nothing more to add, but when there is nothing left to take away.*"
> Antoine de Saint-Exupery

Another erasing principle of Tufte is: "Erase redundant data-ink, within reason". To explain this principle Tufte uses the partial Bar graph in Figure 3-33 where he states that the altitude of the bar can be located in six separate ways. (1) Height of the left line, (2) height of the right line, (3) height of the gray shading, (4) position of horizontal line, (5) position of the number above the bar and (6) the value of the number itself. Removing five of the six ways to located the altitude will result in still having the sixth indicating the altitude. This is of course only possible if we assume that more than one bar is presented in the graph.

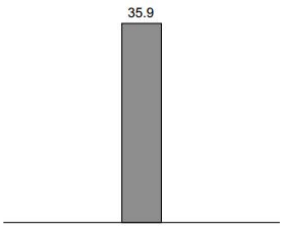
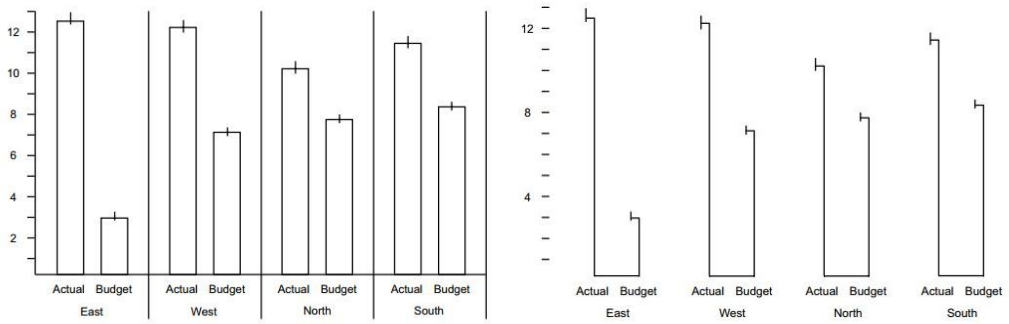


Figure 3-33: Redundant data-ink (Source: *The elements of graphing data*, by Tufte)

Figure 3-34 is an example recreated by Stephen Few [13] that follows the examples and principles created by Tufte. The original visualisation with the redundant data-ink is located at the left. On the right is Tufte's improved version. As you can notice, the left side of the bar has been removed and also the horizontal and vertical axis has been adapted. The vertical lines that divide the regions has also been removed because the white space between the regions already does the perfect job of dividing the regions.



(a) Low Data-ink Ratio

(b) High Data-ink Ratio

Figure 3-34: Reduced Bar graph using Tufte's principles (Source: *Now You See It: Simple Visualization Techniques for Quantitative Analysis*, by Stephen Few)

It is possible to do a lot more reduction in Figure 3-34b. The short vertical lines at the top of each bar can be removed and in the spirit of principles of grouping (see section 3.4.2) we could argue that the proximity between the actual and budget pairs of bars are enough to notice that they are grouped and do not need a horizontal line connecting them. Next we could also only show the labels "Actual" and "Budget" once and remove them on al the other locations. The result is Figure 3-35 [13].

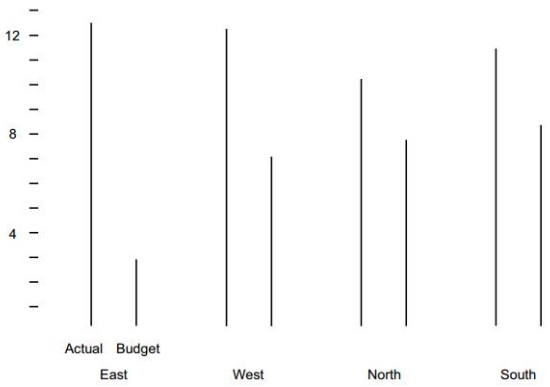


Figure 3-35: Bar graph with no redundant data-ink (Source: *Now You See It: Simple Visualization Techniques for Quantitative Analysis*, by Stephen Few)

As you can notice, the visualisation in Figure 3-35 has been reduced too much. Sometimes a little bit of data-ink redundancy is needed in a visualisation to make it easily readable. When comparing the values some will notice that it is more difficult to see the difference between the values with the lines then with actual bars. Sometimes removing too much redundant data-ink can make the visualisation confusing. It is also what Tufte states in his principle: Stay within reason.

The last principle of Tufte is: "Revise and edit". It is important to create multiple (draft) visualisation of your data. An initial version of a visualisation can almost always be clarified and improved by revision and editing. Sometimes only small changes are needed, in other cases it is possible that you have to go through al the steps and principles discussed in this chapter. Nowadays, computers make it very easy and fast to revise and edit visualisations.

3.4.4 Colours

Lets start this section with an interesting quote from Edward Tufte in his book *Envisioning Information* [47]:

> "*The often scant benefits derived from coloring data indicate that even putting a good color in a good place is a complex matter. Indeed, so difficult and subtle that avoiding catastrophe becomes the first principle in bringing color to information: Above all, do no harm.*"
> Edward Tufte

When using colour well it can really enhance and clarify a visualisation. However, it is also easy to use colour poorly which will probably obscure the data presented in the visualisation and confuse the viewer. Colour is mostly used in information visualisation to distinguish a set of elements from another, to present the elements into separate categories or in other words: to label the data. The reason why colours are often preferred is that it is mostly more effective than the alternatives (see Figure 3-30). Sometimes colour is also used for background, grids and axis, which can work together effectively as long the data-ink principles are considered and it does not confuse the viewer. It is for example interesting to give the background a white colour, the grids a gray colour and de axis a black because people are mostly familiar with this colour use. Colour can also be used to let a viewer be dazzled by a display of data created in a rainbow of colours, what does not help to bring the message ot the data to the viewer. This type of colour use should be avoided.

Colour in computer graphics are specified by the three following dimensions: hue, saturation, and value (HSV) [53]. *Hue* is typically what we mean when we refer to colour, which is defined by terms like blue, magenta, green, yellow and so forth. There is a circularity to our perception of hue which means that we can measure hue in degrees from 0° to 360°. A few examples of hue colours are: red = 0°, green = 120°, blue = 240°. The dimension *saturation*, often also called chroma, refers to the level of intensity and richness of a certain colour, in other word: how pale or dark a colour appears. The lower the saturation the paler the colour becomes. The value dimensions refers to the degree of lightness or darkness. The lower the value the darker the colour becomes. The HSV dimensions are also illustrated in Figure 3-36[1].

[1]source: `http://ie.technion.ac.il/CC/Gimp/node51.html`

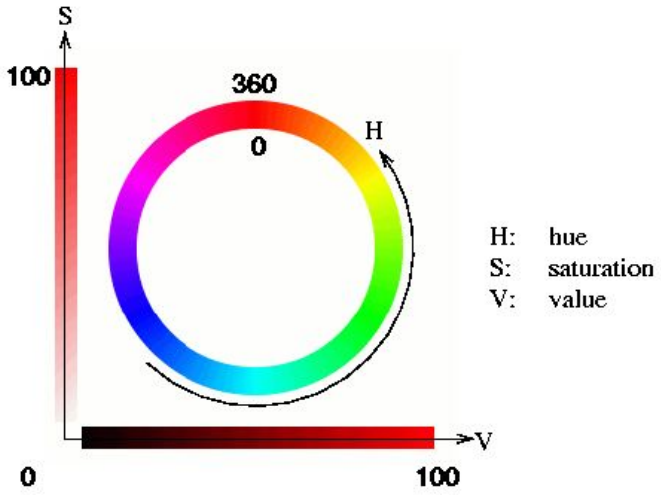


Figure 3-36: Range of hue, saturation and value (Source: *An Appropriate Color Space to Improve Human Skin Detection*, by Ennehar et al.)

Another method in which colour can be specified is CMYK, abbreviation for cyan, magenta, yellow and black. This is a method used by printers to produce colour on paper. We are not going to discuss this further in the paper because the focus on visualisation in this document lays with visualisations on a display. If you want to read more about the HSV and CMYK systems we recommend reading: *Computer Graphics: Principles and Practice* [17] and *Four colors/One Image* [33].

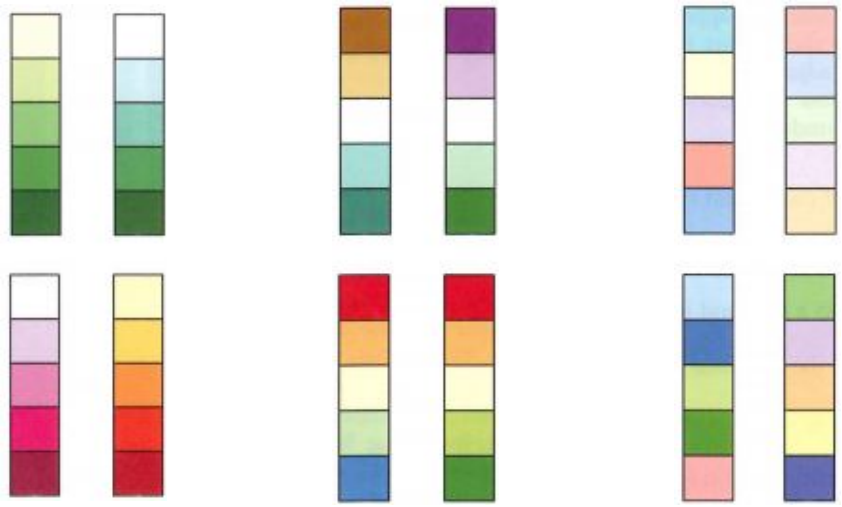
Colour can be used to imply a categorical or ordinal and quantitative differences. This can be done by using different colour schemes. The paper *ColorBrewer.org:An online Tool for Selecting Colour Schemes for Maps* [23] discuss three kinds of colour schemes: sequential, diverging and qualitative schemes.

A *sequential colour scheme* (Figure 3-37a) is used to imply order and best used for representing data that ranges from lower to a higher value like ordinal and quantitative data. This scheme mostly uses the colour values to differentiate the data values, usually light colours for low data and darker colours for high data values.

The second colour scheme is *diverging colour scheme* (Figure 3-37b) and is mostly used when a break point needs to be emphasised. The breakpoint is emphasised by a hue and lightness change in the colour scheme and are

mostly used to present a zero value, median or even the mean. This scheme is also mostly used in ordinal and quantitative data.

The last scheme is the *qualitative colour scheme* (Figure 3-37c) in which mostly only the hue colour changes. This scheme is used when there is no order implied in the data which means that it is mostly used to present categorical data.



(a) Sequential (b) Diverging (c) Qualitative

Figure 3-37: Examples of colour schemes

We only talked about a small part of the use of colour in information visualisations. However, we covered enough to start using colour in visualisation without doing harm to the information that should presented by the visualisation. When interested in a more advanced knowledge of using colour in information visualisation we recommend the work [53] of Colin Ware.

3.4.5 Legend

Almost every visualisation has a legend, however it is better to directly label the data [12]. It is for example possible in a line graph with multiple lines to label them directly by placing the label close to the line. In Grouped bar graphs it can be more difficult to place the labels directly with the bars because there is less room to arrange them properly. In this case you usually need a legend, but you can arrange the labels in the legend to match the

arrangement of the bars to make it much more readable. To address this issue Stephen few created the visualisation shown in Figure 3-38 [12].

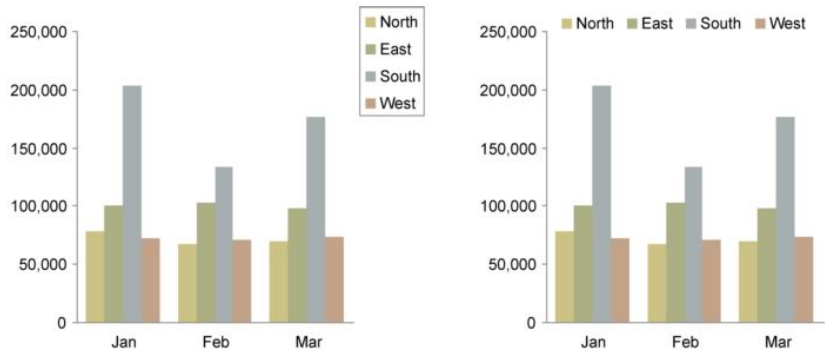


Figure 3-38: Range of hue, saturation and value (Source: *Effectively communicating numbers: Selecting the best means and manner of display*, by Stephen Few)

As you can notice it is easier to read the labels on the right graph where they are arranged the same way as the bars. It is also interesting to see that the legend does not need a border to be identified as a legend. The border is non-data-ink that easily can be removed without distorting the visualisation. The most important thing to remember in this section is to place the label as close as possible to the corresponding element.

3.4.6 Tick Marks and gridlines

It is not mandatory to always use tick marks on the axis. Most of the time tick marks are only necessary on quantitative scales and do not have a real purpose on categorical scales [12]. When using tick marks on quantitative scales it is best to stay between 5 and 10 ticks marks, too many can clutter the visualisation. Using less than 5 tick marks can fail to give the level of details that is needed to get the correct information from the visualisation. The number of tick marks also depends on the size of the visualisation, which means that using between 5 and 10 tick marks is not mandatory, but are more common in the visualisations.

The use of grid lines in visualisations survived from the old days when people had to draw these visualisation by hand on paper. Today, we let computers generate our visualisations which makes us wonder if a grid line is still useful. Stephen Few states that grid lines are only useful when one of the following conditions are met [12]:

- Values cannot be interpreted with the necessary degree of accuracy

- Subset of points in multiple related Scatter plots must be compared

Why remove grid lines while it makes the visualisation more accurately readable? The purpose of a visualisation is not to communicate data within a high degree of accuracy but to show patterns and relationships. For a high degree of accuracy it is better to use a table. When using grid lines make sure that they are lighter then the rest of the visualisation.

3.4.7 Tables

When we decide that only a table is needed to present the data, then we also have to make a series of design decisions:

- Before we decided that we want to present the data in a table we already identified the purpose of the table (steps in section section 3.2.1)
- The next step is to determine which part of the data we have to present to fully bring the messages to the viewer. Sometimes it is not necessary to show all the data
- Designing the table:
  - Do not provide all the cells with a border
  - Do not distort the table by using to much colour
  - Always provide a header row.
  - Provide a clear title for the table
- If needed, highlight the most important data values

An example of a comprehensive table can be found in Figure 3-24. For guidance on this topic and a great deal of more detail about table design, it is interesting to consult the book titled *Show Me the Numbers: Designing Tables and Graphs to Enlighten* [15] and *Envisioning Information* [47].

3.4.8 Conclusion

In this section we took a closer look on how to make sure that a visualisation draws the viewers attention to the message contained within the data. To succeed we need to be familiar with the power of visual perception that will help visualise the data. It is also important to know how we perceive the world as a combination of visual elements, which was explained by using the principles of grouping. When creating the visualisation it is also important to follow the principles created by Tufte, covering the data-ink ratio to help decide if the visualisation is overdecorated. We also covered how to use colour, legends, tick marks and grid lines in visualisations. At the end of this section we also discussed some design steps for creating tables.

4

Graphing Data in Existing Presentation Tools

At the start of the previous chapter we introduced the most commonly used types of visualisations. After that we had a closer look at how to pick the visualisation type that could fit the data in the best way and how we could create this visualisations without misleading the viewer. We also covered how to make sure that the visualisation draws the users attention to the message contained in the presented data.

The previous chapters gave us a lot more knowledge on how to create our visualisations. But what if we did not have al this knowledge and want to create a visualisation? What if we did not read this document or any book, paper or other resource covering information visualisation? Can we still create visualisation that are useful? We already have proven in previous chapters that to create good visualisations, we require skill and knowledge about visualisation rules and guidelines.

There are many tools available today that can help the users with visualising their data. But does these tools provide the right type of help for the users? In this chapter we will investigate some of these tools that help users creating visualisations. Especially in which degree they help the user, who can lack the skills for creating efficient visualisations, in following the visualisations principles and guidelines while creating a graph.

Due to the fact that this thesis focusses on visualisations in presentations we will only investigate how visualisations are created in presentation tools. To be more specific, we will investigate the presentation tools Microsoft PowerPoint, Apple Keynote and Libre Office. We will also briefly talk about Beamer, a LaTeX document class for creating slides for presentations.

4.1 Common Presentation Tools

4.1.1 Microsoft PowerPoint

Microsoft PowerPoint was first created in 1984, but not under the same name and it was not invented by Microsoft[1]. It was created that year by a small start-up called Forethought, Inc. and based on a product proposal from PhD student Robert Gaskns [20]. He was recruited by Forethought to develop the presentation tool called Presenter. This program was later renamed to PowerPoint after trademark problems. It was originally created for Apple Macintosh and allowed users to create slides with text in different fonts. It provided easy editing and there was even the integration for charts from external applications such as MacDraw and MacPaint.

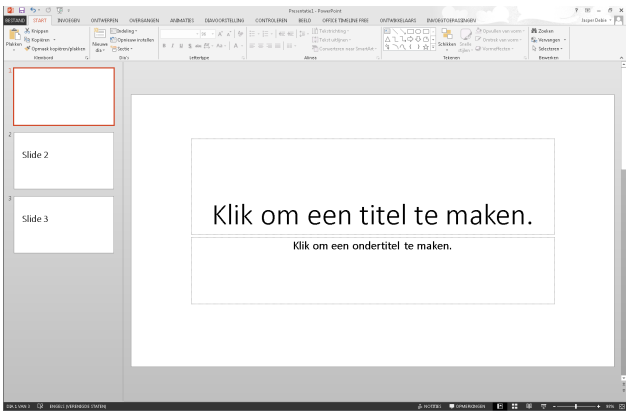


Figure 4-1: Microsoft PowerPoint 2013

In 1987, Microsoft took over Forethought and continued the development of PowerPoint. In 1990 the first version of Microsoft PowerPoint was released and as technology advanced, PowerPoint was further update with more options to create visualisation, new audio possibilities and animation of

[1] http://www.microsoft.com last accessed on 02/07/2014

content and slides. Nowadays, Microsoft PowerPoint, visible in Figure 4-1, is one of the most widely used presentation tools and has also been designed to be usable to everyone. PowerPoint can be used on Windows, Apple and Linux operating systems.

4.1.2 Apple Keynote

Keynote is also a presentation tool that is created by Apple Inc.[1] and was originally used by Apple CEO Steve Jobs to use it for creating presentations used in Apple's conferences and expo's. Later on, the tool was first sold publicly in 2003 as Keynote 1.0, competing against already available presentation tools like Microsoft Powerpoint. It has been maintained and updated up to this day.

When viewing the bigger picture, you can notice that Keynote and PowerPoint are very similar. Keynote follows the same basic concepts as PowerPoint in slides, multimedia and creating visualisations. Even the graphical editor, visible in Figure 4-2, has a lot of similarities. Keynote is mostly used on Apple devices and is not available for Linux and Windows.

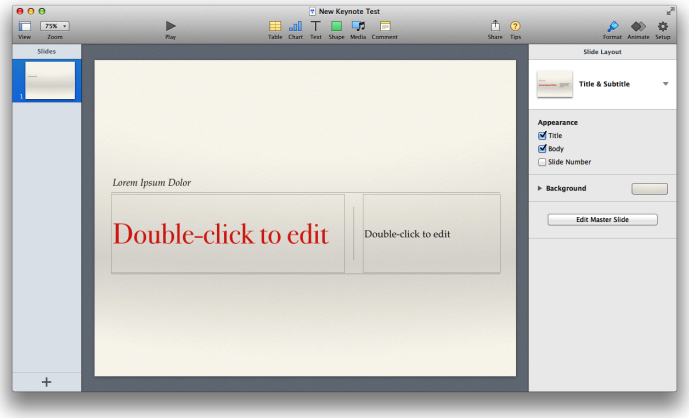


Figure 4-2: Apple Keynote 6.0

[1] https://www.apple.com/ last accessed on 02/07/2014

4.1.3 LibreOffice Impress

Impress is a presentation tool included in the free and open source office suite, which is developed by The Document Foundation[1]. It was forked from OpenOffice[2] in 2010, which is also a free and open source office suite.

Also LibreOffice has a lot of similarities with Keynote and Powerpoint. It has a similar looking looking graphical editor (Figure 4-3) and it also provides the basic concepts like slides, multimedia support and visualisation functionalities. LibreOffice can be used on Windows, Linux and Apple's operating systems.

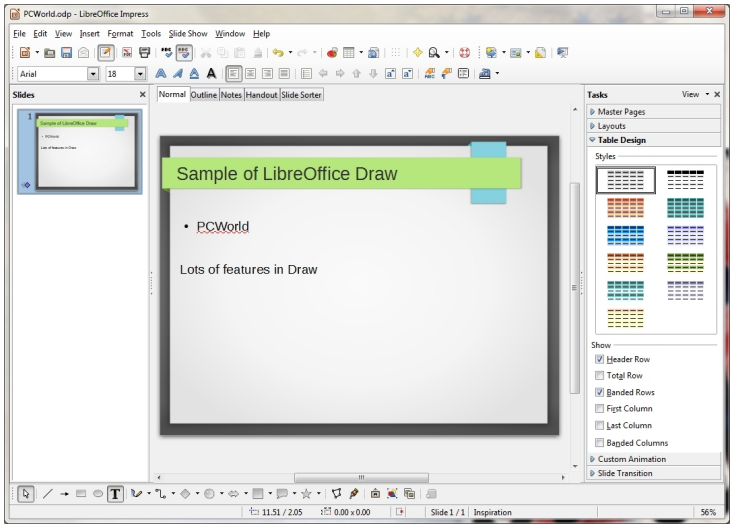


Figure 4-3: LibreOffice Impress

4.1.4 Beamer

Beamer[3] is also a tool for creating presentation, but is different from the common presentation tools. Beamer is a LaTeX document class for creating presentations and also follows the main ideology of LaTeX: *Place the focus on content, not on presentation.* Beamer does not provide a graphical editor, which mean that you have to write source code with a text editor. Beamer can provide the same functionalities as the presentation tools above, but it

[1] `http://www.documentfoundation.org/` last accessed on 02/07/2014

[2] `https://www.openoffice.org/` last accessed on 02/07/2014

[3] `http://texdoc.net/texmf-dist/doc/latex/beamer/doc/beameruserguide.pdf` last accessed on 02/07/2014

needs to be written in source code. Following source code snippet shows how to create a presentation with two slides.

```
\documentclass{beamer}
\title{A Tiny Example}
\author{Andrew Mertz and William Slough}
\date{June 15, 2005}

\begin{document}
\maketitle

\begin{frame}
\frametitle{First Slide}
Contents of the first slide
\end{frame}

\begin{frame}
\frametitle{Second Slide}
Contents of the second slide
\end{frame}

\end{document}
```

Listing 4.1: Sample Beamer code

As you can notice, Beamer is a presentation tools for more advanced computer users. However, even these users do not always have the skills to create the best visualisations. This makes it also interesting to briefly investigate how this tool helps the user, who is more familiar with working on computers, to create visualisations.

4.2 Visualisations in Presentation Tools

In this section we will investigate how common visualisation types can be created by the presentation tools that we introduced in section 4.1. We will take a closer look at what these presentation tools do to compensate some people's lack of skills in creating visualisations. By looking how much these presentation tools follow the principles and guidelines defined in chapter 3 we can get a better idea of how these tools really help creating good and efficient visualisations. We will only investigate the visualisation types in these presentation tools defined in section 3.1.

Note, to test if the principles and guidelines are respected we will create some visualisations. We will not fully investigate if it is possible to create visualisation that follow the principles and guidelines defined in chapter 3. What we will do is investigate if the presentation tool helps you choose the correct type of visualisation and if we have, after completing the wizard for creating these visualizations, a good visualisation without the need of any tweaking afterwards.

4.2.1 Picking the Visualisation Type

The first thing we investigated is if the presentation tool helps us with choosing the best visualisation type for our data. What we notice in Impress, Keynote and Beamer is that they do not provide any functionality or help to pick the best suiting visualisation for the data the user wants to present. In Impress and Keynote we can pick every visualisation type for every type of data. It does not matter how the data set looks like, these two presentation tools will almost always try to visualise it. This can give strange and/or incorrect visualisations of the data. In beamer we have to write source code to create a visualisation, which has as consequence that we will notice more early that the data does not fit the visualisation. However, it is also possible with Beamer to create a visualisation that does not fit the data.

It is not only situations like too few/many dimensions or variable in your data to create a good graph with that particular visualisation type. As explained in section 3.2.1, the message you want to present with your visualisation is also important. So the correct visualisation type has to be picked if you want to present your message in the best possible way. For this there is also no support in Impress, Keynote and Beamer.

In contrast to the other tools, Microsoft PowerPoint does help the user with picking the best visualisation type. Depending on how the data set looks like, PowerPoint proposes some possible visualisation types. For every proposed visualisation type it provides some information about what the visualisations does and in which situation you have to use it. What it does not tell us is what type of relationships (section 3.2.3) the visualisation will provide, which does not help us bring the message in the best possible way to the outside world. Also, this functionality is not mandatory to use in PowerPoint, which means that a user can also pick a visualisation that is not recommended by Microsoft PowerPoint. An example of this functionality is visible in Figure 4-4, with in the left the recommended charts and at the right more details why to pick that type.

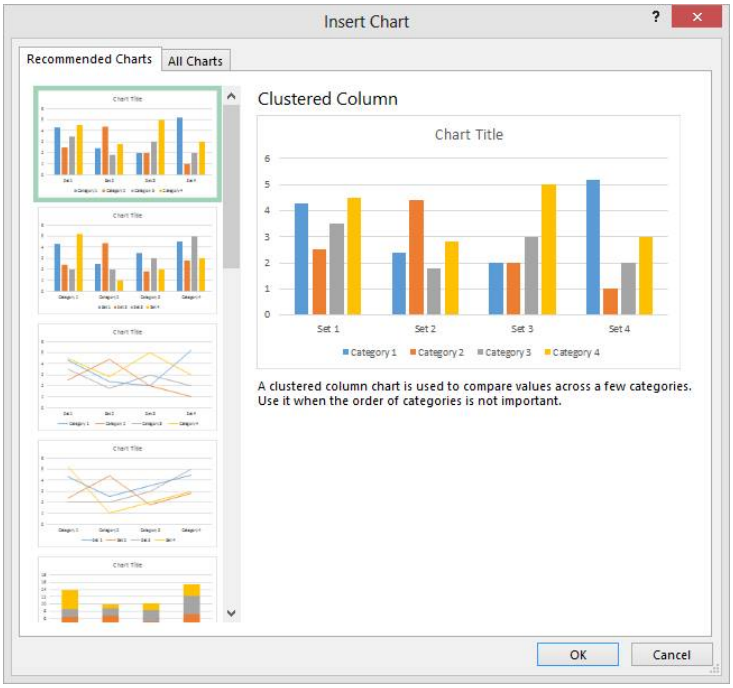


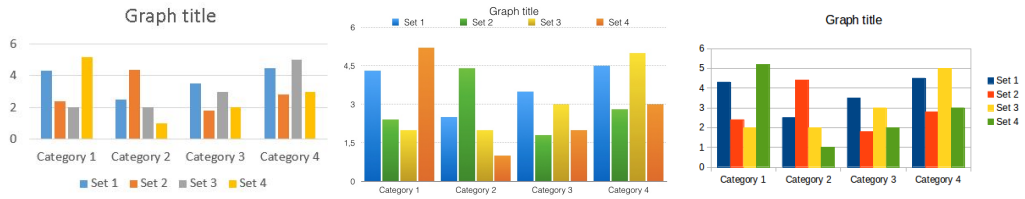
Figure 4-4: Recommended Graphs in Microsoft PowerPoint

4.2.2 Testing Visualisation Type

In this section we will investigate how the visualisation types are created in the presentation tools presented in section 4.1. For this we will only use the wizard that is provided by the presentation tools. When the wizard created the visualisation, we will not do any tweaking on the visualisation afterwards. The reason for this is that we want to test if the wizard takes into account the guidelines and principles create by many researchers like Tufte, Cleveland, Few and so on. When this is not done by the wizard, the user has to tweak the visualisations afterwards to implement some of these principles and guidelines. As we have seen before, a lot of peoples lack the skill and knowledge to implement this. Beamer, which does not have a wizard for creating visualisation will be discussed afterwards.

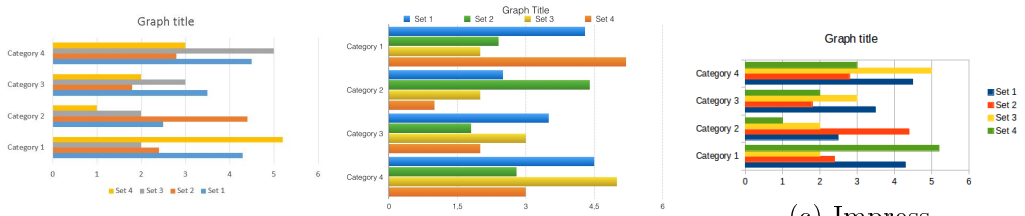
Bar Graph

PowerPoint, Keynote and Impress all provide the creation of horizontal and vertical bar chart in their wizard. Also grouped and stacked bar graphs are available. Figure 4-5 is an example of how Grouped column bars are created in these presentation tools and Figure 4-6 is a example of the Grouped horizontal bar graphs.



(a) PowerPoint (b) Keynote (c) Impress

Figure 4-5: Grouped Column graph



(a) PowerPoint (b) Keynote (c) Impress

Figure 4-6: Grouped Horizontal bar graph

The grouped horizontal and vertical bar graph created by the wizard are pretty good chart. But there are some small remarks:

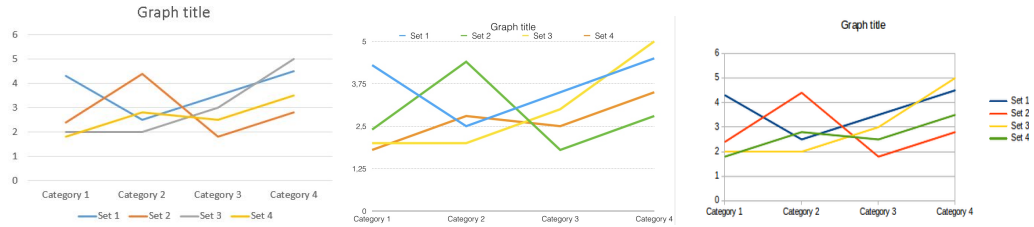
- The grid lines in the PowerPoint and Keynote versions are lightened and draw not too much attention (use of grid lines in section 3.4.6). The bar graphs created in Impress are much darker and in Figure 4-5c there are some vertical lines that are not necessary.
- By placing white space between the categories they are using the principles of grouping (section 3.4.2). In the Keynote versions this white space is rather small which makes the principles of grouping less precise in these visualisations.
- It was also possible in the wizard to take something else then bars to visualise the data, for example 3D pyramids and ovals
- You could argue the use of colour, but they present different sets, which is best shown by different hues. It is of course possible to lighten the colours a bit, what is done slightly in the PowerPoint versions
- The Legend is better placed above or below the bars in Figure 4-5c and not at the right side. In Figure 4-6c they are placed on the correct size while for Keynote(Figure 4-6b) and PowerPoint(Figure 4-6a) it would be more interesting to place the legend on the right side.

Dot chart

Dot charts or Dot plots are not a default chart option in PowerPoint, Impress and Keynote. There are some workaround to create a Dot chart in these presentation tools by taking another type of visualisation and tweak it until it looks like a Dot chart. This is not the way that this type of visualisation should be created. Cleveland and McGill showed in their research [4] that the locations of dots on grid lines are more accurately than the size of bars, which are used in bar graphs. This is a type of visualisation that should be available in presentation tools.

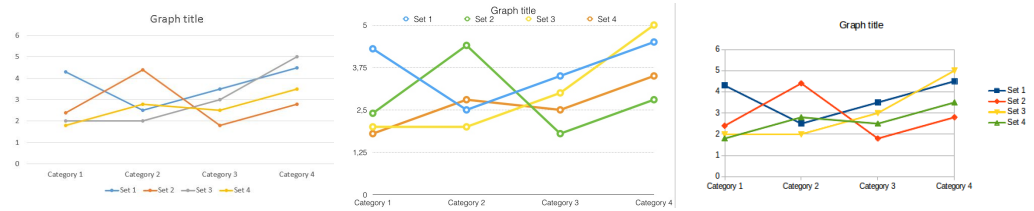
Line chart

Figure 4-7 are examples of Line charts that do not highlight the individual values while Figure 4-8 are Line graphs that do highlight the individual values by using symbols on the location of these values. As you can see, this visualisation type is provided by all three presentation tools.



(a) PowerPoint (b) Keynote (c) Impress

Figure 4-7: Line graph



(a) PowerPoint (b) Keynote (c) Impress

Figure 4-8: Line graph with highlighting of individual values

These graphs that are generated by the wizard are good visualisations, but we have still some remarks:

- The same case as of the bar charts: the grid lines in the PowerPoint and Keynote versions are lightened and do not draw too much attention (use

of grid lines in section 3.4.6) while those created in Impress are much darker. Also some vertical lines in Figure 4-9c that are non-data-ink.

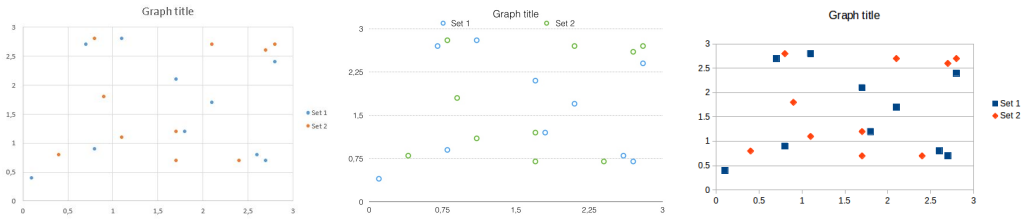
- In section 3.3.2 we saw some guidelines on how to pick the correct scale for the axis. The scales picked in the example are good, but PowerPoint picked the best scales. By leaving some white space on the left and right of the lines it is clear that all the data is visible on the visualisation.

- In Figure 4-8c you can see that Impress uses different symbols to highlight the individual values. This is not necessary and it can distract the viewer from what is really important in the visualisation, which is the data.

- For line chart it is better to use direct label the visualisation (section 3.4.5). So instead of using a legend next to the graph, place the labels close to the corresponding Line.

Scatter plot

In Figure 4-9 you can see how Scatter plots are created in PowerPoint, Impress and Keynote after completing the wizard for creating visualisations.



(a) PowerPoint (b) Keynote (c) Impress

Figure 4-9: Scatter plot

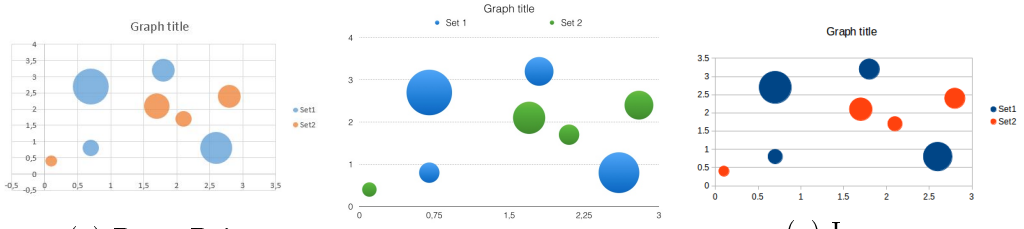
We also have some remarks for this visualisation type created by the presentation tools:

- We keep repeating it: the grid lines in the PowerPoint and Keynote versions are lightened and draw not too much attention (use of grid lines in section 3.4.6 while those created in Impress are much darker. Also some vertical lines in Figure 4-9c that are non-data-ink.

- Powerpoint also uses vertical grid lines while the other presentation tools only horizontal grid lines. Stephen Few explains in his work [12] when to use grid lines (see section 3.4.6). In this case, when you really want to compare the dot values it is interesting to add both grid lines.

- In Figure 4-9c you can see that Impress uses to much different symbols to show the different variables in the visualisation. This is not necessary because colour is already used to separate the two sets. In section 3.4.4 we saw that colour is more effective than the alternatives to distinguish variables.

In Figure 4-10 we see some examples of Bubble plots created in the presentation tools Powerpoint, Keynote and Impress. A bubble chart is a special case of a Scatter plot with the only difference the size of the dots, just for adding an extra dimension.



(a) PowerPoint (b) Keynote (c) Impress

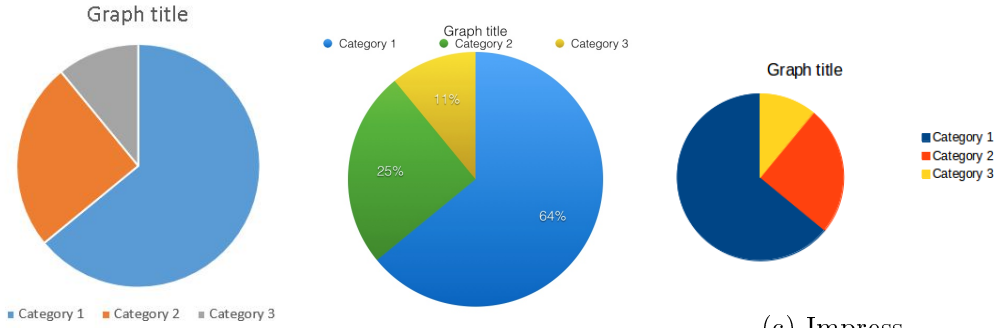
Figure 4-10: Bubble plot

For this visualisation type we also have three remarks:

- For this chart type you still have the problem with Impress creating dark grid line and vertical lines that are not necessary
- In PowerPoint a negative x-axis and y-axis appeared, probably because they add white space around the bubbles. This should be avoided
- If you present a third dimension by using the size of the bubbles then do not forget to label that third dimension. Otherwise the viewer will have no idea what the size of the bubbles present.
- Try to avoid overlap of bubbles.

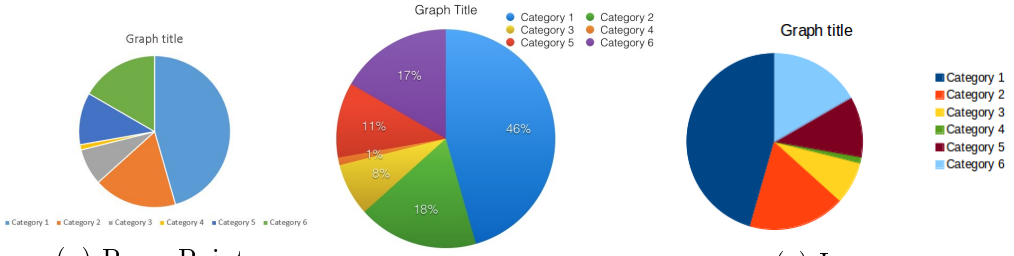
Pie chart

Figure 4-11 and Figure 4-12 are examples of the Pie chart, a visualisation type that is not popular by a lot of researchers in the information visualisation field but loved by many others. This famous visualisation type is a default chart option in PowerPoint, Impress and Keynote.



(a) PowerPoint (b) Keynote (c) Impress

Figure 4-11: Pie chart



(a) PowerPoint (b) Keynote (c) Impress

Figure 4-12: Pie chart with to many variable

Some remarks about the creation of Pie charts in the presentation tools:

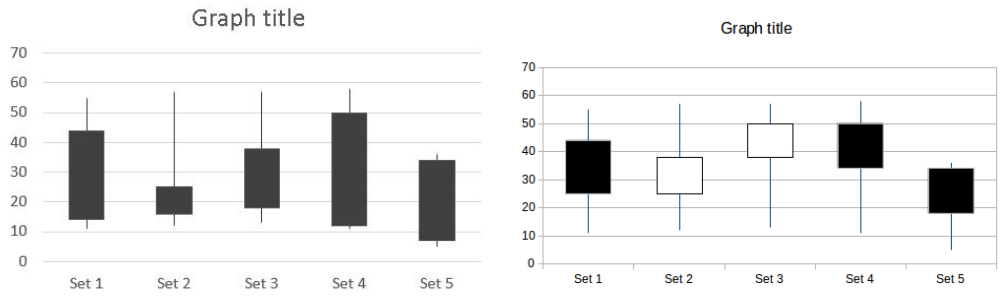
- It is not easy to read the degree from a slice, so it is interesting to place a percentage in the slice (e.g. Figure 4-11b).
- Do not use a legend, it is better to directly label the data [12].
- If using Pie charts, make sure you do not have to many variables. Pie charts are most effective when at least most of the slices represent 25% to 50% of the whole. The Pie charts created in Figure 4-12 are visualisations that are better not created.
- The key slice should starts at 12 o'clock, but the wizard only follows the order of de data. Changing the order of the data will change the order of the slices.

Histogram

The histogram is also not a default chart option in PowerPoint, Impress and Keynote. There are some workarounds to create a Histogram in these presentation tools by counting how many items there are in each numerical category and present that in a Bar graph, in which there is no white space between the bars. Then you still need to do a few tweaks with the scales until it looks like a Histogram. This is of course not the way it should be done. A Histogram is a visualisation that is used a lot and should be a default chart option in the presentation tool.

Box plot

The box plot is only a default chart option in PowerPoint and Impress, visible in Figure 4-13. It is again possible with a workaround to create this visualisation type in Keynote.



(a) PowerPoint

(b) Impress

Figure 4-13: Box Plot

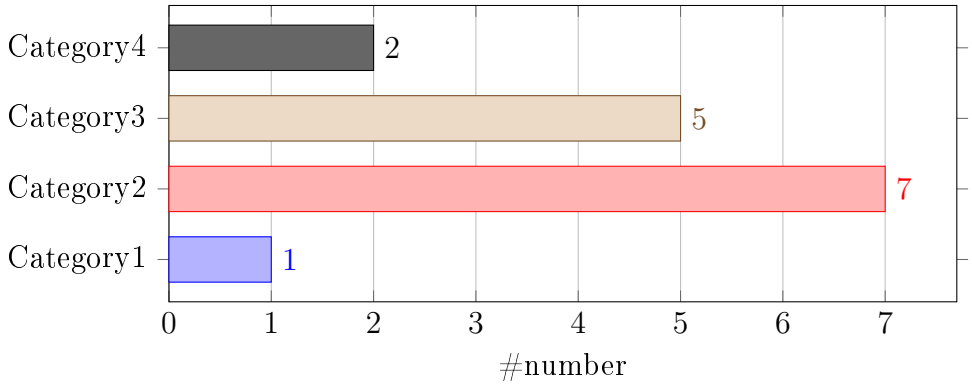
The Box plots created in the presentation tools can be improved a lot:

- It is not possible to see the mean value in Figure 4-13. Better remove the black fill of the boxes, which is non-data-ink, and indicate the mean with a horizontal line.
- The Box plots would be better readable with direct labelling. When using direct labelling you do not need the grid lines and y-axis.
- There is no need for colour in this visualisation type.

Beamer

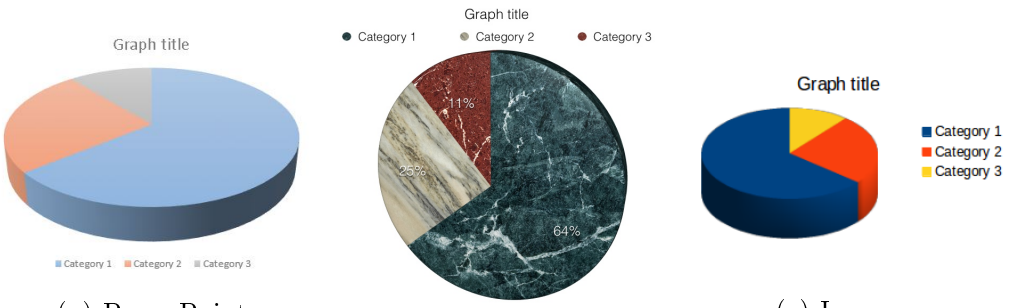
In beamer there is no wizard available to create al these visualisation types. This does not mean that it is not possible to create these visualisation types in Beamer. On the contrary, it is possible and there are even more tweaking possibilities than in the other presentation tools. This is because you create

the visualisation with source code that you have to write by yourself, which is not an easy task. This means that the quality of the visualisations fully depends on the skills of the creator. Next visualisation is a Horizontal bar[1] graph created in Beamer, or to be more precise, created in LaTeX .



4.2.3 Misleading Visualisations

In the wizards of the presentation tools you cannot really find options to create deceptive visualisation on purpose. However, it is possible to select some options that can accidentally lead to misleading visualisation. It is for example possible in a lot of presentation tools to create a 3D version for some of the visualisations. Figure 4-14 are examples of 3D visualisations of a Pie Chart. These 3D examples are not very misleading, but remember from section 2.4.2 how misleading they can be.



(a) PowerPoint (b) Keynote (c) Impress

Figure 4-14: 3D Pie chart

[1]Source: http://tex.stackexchange.com/questions/128028/bar-chart-single-colored-bar, last accessed on 23/07/2014

With the presentation tool wizards it is not possible to create an inconsistent scale for the axis to get misleading visualisation like Figure 2-14. But it is possible to choose the start and the end value of the scale, which was also done in Figure 3-23. So it is possible to create misleading visualisations with the wizard, but most of the time you have to make an effort for it.

4.3 Conclusion

The visualisations created by the wizard in the presentation tools PowerPoint, Keynote and Impress are overall not bad visualisations. However, some of them still need some improvements, which can be achieved by following the principles and guidelines defined by researchers in the visualisation field. The biggest problem is still that the users do not get a lot or any help concerning picking the correct visualisation type for the data. Picking a bad visualisation type can result in visualisations that do not present the message from the data correctly. It is also still possible to create misleading visualisations in current presentation tools, which we want to avoid when presenting to an audience.

5 Towards Better Visualisations

In previous chapters, we have discussed the common problems that can be found in current visualisations, followed by research on the principles and guidelines that define how a good visualisation should look like. We also investigated how these principles and guidelines are implemented in current presentation tools and noticed that there is still room for improvement in these tools. As we have stated earlier, visualisations in presentations can be very powerful and useful to transfer knowledge to other people, but they can also convey the wrong information or confuse the viewers. The quality of the visualisation has an influence on how fast and understandable this knowledge is transferred to the other people. However, presentations are all about transferring knowledge to other people. This implies that a presentation tool has to provide the necessary functionality to let the user create data visualisation that enhance this knowledge transfer. This is however only possible if the presentation tool provides a way to create efficient and effective data visualisation.

In this chapter we step away from the classical visualisation approach in common presentation tools and discuss how our ideal process of creating visualisation for in a presentation tool would look like. We will start by defining some requirements a presentation tool should support when it provides the user with functionalities to create visualisations of their data. Afterwards, we present our tool that supports these requirements in an ideal visualisation process as a theoretical design. The actual implementations of

this tool, which is named Visualisation Picker, will be described in Chapter 6. We will also introduce a presentation tool called MindXpres, which we will extend with quality data visualisations that can be used to present to an audience during presentations.

5.1 Requirements for Visualising Data

5.1.1 Understand the Data

The first step that is essential in creating visualisation of data, and which is also the step that is often missed, is determining what the user wants to say or show with the data. In most common presentation tools the user never has to define what type of message or knowledge he or she want to bring to the audience. We only have to select the range of data we want to visualise and choose a visualisation type to represent the data. However, we have seen in Section 3.2.1 that only when we know the clear purpose why the visualisation should exist, we will be able to determine what is needed in the visualisation to communicate the message and transfer the users' knowledge to the audience.

As we have already discussed, every type of relationship has some potential visualisation type that will support the message best. It is not possible for an application to always detect which message the user wants to bring to the audience. When the presentation tool cannot detect the message contained in the data, then it should be at least possible for the user to define what he or she wants to show with the visualisation of the data. This can be done by categorizing the message in one of the seven common relationships defined in Section 3.2.3.

The data type (Section 3.2.1) and the variables types (Section 2.2.6) of the data set also have a great influence on the resulting visualisation. This information is for example very important for picking a visualisation type. However, sometimes it also needs some user input to be analysed correctly from the data set. For example, zip codes are numeric and could be seen by the tool as quantitative, while they are actually categorical variable. The message, data and variable types of a data set is important information that is necessary for picking a visualisation type and creating a suiting visualisation of the data. This information should always be analysed from the data set and presented to the user as verification.

5.1.2 Guidance in Picking a Visualisation Type

We cannot expect that all the people that want to create a visualisation of their data know which visualisation type can be used for each message, data and variable types. That is not all, other factors also play a role in picking a visualisation type, for example the size of a data set, range between minimum and maximum, etc. There are too many guidelines and principles for picking a visualisation type which makes it impossible for a human to always take them into account when creating a visualisation. However, we have seen in Section 3.2 that it is very important to pick the best possible visualisation type to maximise the understanding of the message contained in the data and to make the knowledge transfer as effective and efficient as possible.

There are so many factors we have to take into consideration when creating a visualisation of the data that it may be interesting for presentation tools to guide the user to the correct visualisation type. When the presentation tool has the information that is described in the previous requirement, it can already help the user a lot in the process of picking a type of visualisation. It is not always possible to say with certainty that one type of visualisation will work best for a given data set, which means that the presentation tool may suggest multiple visualisation types. However, the presentation tool should definitely avoid giving the user the ability to create visualisations using a visualisation type that does not suit the data at all. In Section 2.4.1 we have seen what too much freedom in picking visualisation types does with the quality of the visualisations.

5.1.3 Suitable Design

In a presentation we also have to make sure that the visualisation draws the users' attention to the message included in the data and not to something else, for example colourful backgrounds. It is important in a presentation that the visualisation is very effective and efficient, so that the audience does not have to spend too much time on analysing the visualisation. When the audience has to place its focus too long on the visualisation, they could miss other information on the slide or information given by the presenter. A good visualisation presents the message with a clean and simple design without extra decoration, or in other words without extra non-data details. Adding extra non-data details only results in a less effective and efficient visualisations.

There are also a lot of guidelines and principles about design aesthetics for data visualisations. The more common guidelines and principles are already discussed in Section 3.4.1, which are only a fraction of what can be found in the field of information visualisation about design aesthetics. The common presentation tools give the users to much freedom in designing their visualisations while it actually should help the user apply these guidelines and principles in their visualisations. We are of the opinion that a presentation tool is responsible for the design of the visualisation. This means that the user does not have to spend time on designing the visualisation and in this way cannot create visualisation with extra non-data details. When the presentation tool is designing the visualisation, it has to take the design principles and guidelines into account.

5.1.4 Extensibility

When we were experimenting with common presentation tools in Chapter 4 we noticed that some of the common visualisation types are not supported, for example the Histogram and the Dot plot. Some of these visualisation types do not have an alternative visualisation type that presents the data with the same efficiency and effectiveness. That is why we are of the opinion that a presentation tools should be extensible with new visualisation types created by the user.

It should also be possible to adapt the implementation of a supported visualisation type. When new guidelines or principles are found they need to be implemented in the corresponding visualisation type to further improve the quality of the visualisation.

5.2 The Ideal Visualisation Tool

Our implementation called Visualisation Picker is an extensible visualisation-driven web-based tool that helps the user create visualisations of the data that can be used as content for a presentation. The Visualisation Picker satisfies all the requirements that are defined in previous section and provides a graphical user interface in which the user can easily follow the process of visualisation creation. How these requirements and visualisation process are implemented in Visualisation Picker, is described in Chapter 6. Now we will first take a closer look in what we see as the ideal process of visualisation creation.

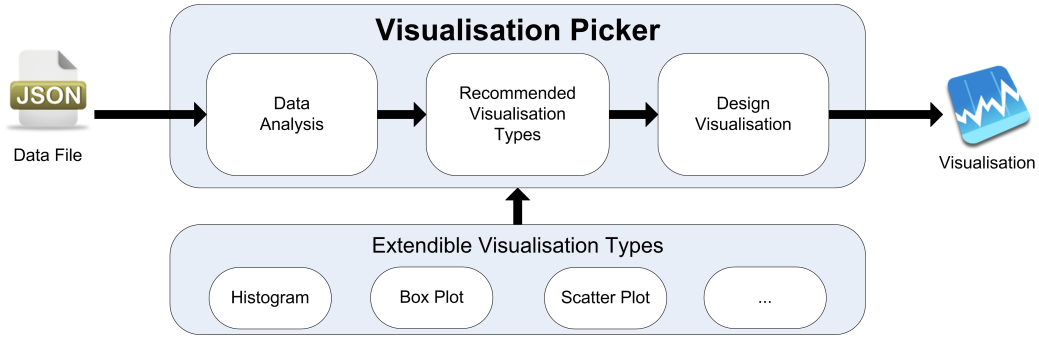


Figure 5-1: Visualisation Process in Visualisation Picker

As shown in Figure 5-1, the visualisation process we use in Visualisation Picker consists of the following components:

- JSON Data File as input for Visualisation Picker
- Data Analysis
- Extendible Visualisation Types
- Recommended Visualisation Types
- Design Visualisation
- Visualisation as output of Visualisation Picker

Together these components span the entire process of using the Visualisation Picker, from reading the input data file to the actual creation of the visualisation. It all begins when the user provides the Visualisation Picker with a JSON data file. The data of which the user want to create a visualisation is stored in this JSON file. This data file has to be readable by the Visualisation Picker, which means that we also have provided a Schema to check the format of the file and the used syntax. This Schema also allows the user to define properties of visualisations in the file, for example the title, label for X and Y-axis, description and even the type of the data, message and variables.

Once the JSON data file has been created and loaded into the Visualisation Picker, it will first have to go through the built-in compiler, which will check if the data file matches with the Schema. Afterwards Visualisation Picker starts analysing the file and shows the result of the analysis to the user by means of a input form. This makes it possible for the user to change incorrect or incomplete information and even adapt information that is incorrectly

analysed. It is also the location where the user can define his message that he or she wants to present with the data. It is very important that all the information in this step is defined correctly, knowing that it will have big influences on the resulting visualisation.

After analysing the data it is time for the Visualisation Picker to start the next process in the visualisation creation: generating a list of recommended visualisation types that suit the data. In this step it is important to match the properties of the data with the available visualisation types and generate a list of the visualisation types that can present the data and the message in the best possible way. The number of visualisation types the Visualisation Picker has to choose from, depends on the number of types that has been implemented as plug-ins in our implementation. The Visualisation Picker can be extended with new visualisation types as plug-ins to improve the quality of this process. Our implementation also makes it easy to try the proposed types of visualisations on the data and switch between the types with only using one mouse-click.

By using a plug-in system, of which each plug-in represents a visualisation type, we make Visualisation Picker an extendible tool. This makes it possible to extend the ideal visualisation process in Visualisation Picker with new types of visualisations. These plug-ins can be created or extended by users with the necessary skills in information visualisation. This way we do not need workarounds to use a visualisation type that is not supported by the tool or wait until the tool developers decide to support the visualisation type. These plug-ins contain the properties and de design aesthetics of the visualisation type.

After the user picked a visualisation type our implementation can start creating a visualisation of the data. For creating these visualisations the Visualisation Picker will follow the guidelines and principles that we have defined in Section 3.4. It is the intention that the visualisation that is created in our ideal visualisation process does not need to be adapted afterwards by the user. Our Visualisation Picker can provides multiple different designs of the same visualisation type, which makes it possible for the user to pick a design he or she finds most interesting.

The visualisations that are created by the Visualisation Picker can also be used in the presentation tool MindXpres, which will be discussed in next section. The user does not have to adapt the visualisation to add it to this presentation tool. All the functionality and interaction possibilities of the created visualisation in the Visualisation Picker is also available when the same visualisation is placed in the zoomable user interface of MindXpres. This makes it possible to interact with the visualisation during a presentation.

5.3 The MindXpres Presentation Tool

We will now introduce a presentation tool that is called MindXpres, which describes itself as a "web technology-based extensible platform for content-driven cross-media presentations" [37]. MindXpres has a modular architecture and a very interesting plug-in mechanism which enables the reuse and integration of components for new visualisation and interaction possibilities. This way it introduces a radical new way of creating presentations that can resolve issues and lack of features in current presentation tools like Microsoft PowerPoint, Apple Keynote, and LibreOffice Impress.

Some of these issues and lack of features are:

- Linear traversal of the slides without the possibility to easily navigate between slides.
- Not possible to display multiple slides at once.
- Mostly presenter focussed, which means that most of the functionalities are only for the presenter and not for the viewers.
- Most of them not easily extendible.

In MindXpres these problems are already solved without any loss of the usability of the presentation tool. That MindXpres also solves the last issue is for us of most importance. In most of the common presentation tools it is not possible to extend the way charts are created. However, MindXpres gives us the opportunity to extend itself with the functionality to create great visualisations of data that follow the principles and guidelines of many researchers in the information visualisation field.

MindXpres is similar to a LaTeX documents, which also separates the content and visualisation of the content. The visualisation is created by a compiler by using a predefined theme and the content outputted in HTML5[1] format which can be viewed in almost any browser.

In the following sections we will introduce ourself in the base and the building blocks of MindXpres. We will discuss the architecture of MindXpres and also the plug-in mechanism that we will use for our implementations.

5.3.1 MindXpres Architecture

In this section we will discuss how MindXpres is build, without going into detail. We will first discuss the MindXpres document format, together with the authoring Language. Afterwards we will talk about the compiler, which will validate the MindXpres document and generate HTML code. We will also briefly discuss the Presentation Bundle. For more details about MindXpres, we want to refer to [37] and [38]. Figure 5-2 shows us the general architecture of MindXpres.

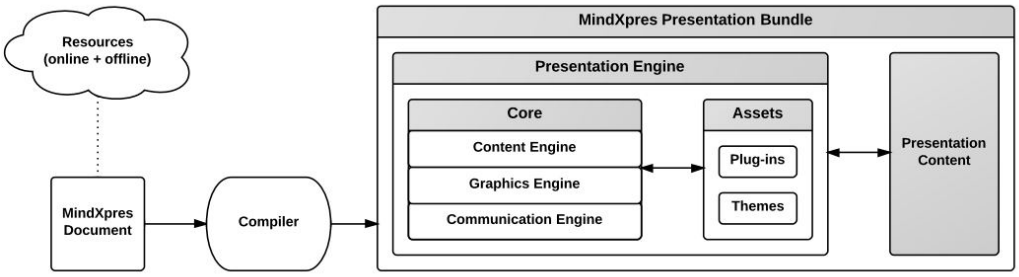


Figure 5-2: MindXpres architecture (Source: *MindXpres - An Extensible Content-driven Cross-Media Presentation Tool*, by Roels and Signer)

5.3.1.1 Document Format and Authoring Language

The content and the content structure are stored in a XML format into a MindXpres document. These MindXpres documents can, next to their own content, also refer to external content that has to be included. This document works on the same idea as a LaTeX document, a document that can be written in a text editor and only contains content. In latest work

[1] `http://www.w3schools.com/html/html5_intro.asp`, last accessed on 24/07/2014

they also created a graphical authoring tool [24], which can also create these MindXpres documents. The authoring language is created to eliminate unnecessary rules and attributes of HTML and XML so that our plug-in can easily define new vocabulary to provide support for new media types and structures. More about this later in section 5.3.2.

Following code snippet is a very small example of a MindXpres document. With this document a presentation can be generated with 2 slides. The first slide contains a image while the other slide contains a bullet list. Every slide also has a title. The theme in which the presentation will be created is also defined, in this example it will be the VUB theme.

```
<presentation theme="VUB"  author="Jasper Debie">

  <slide title="Slide1">
    <image src="foto.jpg">
  </slide>

  <slide title="Slide2">
       <bulletlist>
         <item>item1</item>
         <item>item2
           <item>item2a</item>
           <item>item2b</item>
         </item>
       </bulletlist>
    </slide>
</presentation>
```

5.3.1.2 Compiler

The compiler transforms the MindXpres document into a collection of other document formats. This collection is called the MindXpres *presentation bundle*. The compiler allows to create different types of presentations, from dynamic and interactive presentation to a more static output such as PDF documents. The compiler can also solve incompatibility issues, such as unsupported video formats.

Following code snippet show how the compiler can transform the previous example into the following HTML5 code:

```
<div data-type="presentation" data-theme="vub" data-author="
    Jasper Debie">
  <div id="element_1" data-type="slide" data-title="Slide1">
    <img  src="foobar.jpg">
  </div>
  <div id="element_2" data-type="slide" data-title="Slide2">
    <ul>
     <li>item1</li>
     <li>item2
      <li>item2a</li>
      <li>item2b</li>
     </li>
    </ul>
  </div>
<div>
```

5.3.1.3 MindXpres Presentation Bundle

The MindXpres presentation bundle consists of compiled presentation content, with the content still the same as the original MindXpres document but transformed in another document format. Note, the content might have been modified by the compiler to be compatible with the platform the presentation will be executed on. The core of the presentation bundle consist of 3 modules, as you can see in Figure 5-2. The *content engine* is responsible for processing the content and linking the content to the corresponding plug-in. The *graphic engine* is responsible for the rendering graphic-related functionalities, for example zoomable interfaces. The *communication engine* consists of a communication API that can be used by the plug-ins to communicate. Then we also have the *Assets* component in the presentation bundle, which consist out of plug-ins and themes. *Themes* can be used to style your presentation.

5.3.2 Plug-in Mechanism

In this section we will take a closer look at the plug-in mechanism in MindXpres. This feature of MindXpres is of great interest to us, because it allows us to add a new media type to the presentation tool. That is exactly what we need, a way to add a new media type for graphs to visualise information.

Due to the plug-in mechanism in MindXpres there is no need to hardcode the core with non-core modules. These modules can be implemented as plug-ins which allows third parties to replace, modify or even create new

modules. Basic types such as text, bullet lists and images are all implemented in MindXpres as plug-ins. This feature makes it really easy to extend MindXpres, something that is often limited or even lacking in other existing presentation tools.

The plug-ins in MindXpres are divided in three major categories:

- *Components* are the basic building blocks, represented as plug-ins that provides functionalities for the specific content types such as text, video, images etc. When we want to add graphs to MindXpres we will have to create a new component plug-in to support this new media type.
- *Containers* are plug-ins that are responsible for organising and grouping the components plug-ins. An example of a container is a slide, which has to organise the slide title, slide number and other components displayed in the slide. You can of course create other containers, such as a container with scrollable images.
- *Structures* are the high-level structures and layouts, used for organising components and containers. While Containers only mark the beginning and end of a set of elements that has to be organised, structures are more complicated than that. Structures may also influence the navigational path trough the content.

5.3.3 How to create plugins

In MindXpres, every plug-in is placed into individual folders that are located in the component, container of structure folder. A plug-in folder contains all JavaScript, Cascading Style Sheets, libraries and other resources that are necessary to make the plug-in work. Due to the fact that every plug-in has it own folder, it is easy to add a new plug-in or remove a plug-in. As mentioned before, MindXpres uses naming conventions to detect plug-in files in the file system. Every plug-in has a file called plugin_info.js that is loaded first and contains some plug-in specific information for MindXpres, for example tags and the name of the plug-in. When MindXpres knows the name of the plug-in, it loads the 'name'.js file to execute the functionalities of the plug-in. Figure 5-3 illustrates the folder that contains the plug-ins.

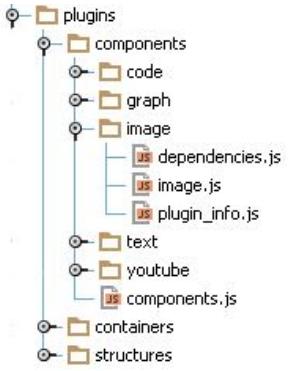


Figure 5-3: The plug-in structure

5.4 Conclusion

In this chapter we described what we think are the minimum requirements a presentation tool has to meet when providing functionality for visualisation creation. Afterwards, we discussed our ideal visualisation creation process and how our new tool called Visualisation Picker supports these requirements and the visualisation process. At the end we also introduced a presentation tool called MindXpres, for which our Visualisation Picker will create quality visualisation. In the next chapter we discuss the actual implementation of Visualisation Picker and the integration of a visualisation plug-in in MindXpres. We will discuss the programming details and how the requirements and visualisation process is developed into a functional implementation.

6

The Visualisation Picker Tool

In Chapter 2 we have defined the more common challenges and problems of presenting data in a visualisation, and in Chapter 3 we explained how to avoid these challenges and problems by following guidelines and principles created by researcher from the field information visualisation. Afterwards, in Chapter 4, we investigated how well these guidelines and principles are used in common presentation tools. We noticed that these presentation tools follow some of the principles and guidelines, but also expect that the user knows how to pick the best visualisation type and how to increase the quality of the visualisation created by the wizard. With our implementation, called the *Visualisation Picker*, we want to make sure that a user can create a visualisation without having to pick a visualisation type on his or her own. We also want to avoid that a user has to improve the visualisation after it has been created by our implementation.

In Chapter 5 we have defined important requirements and explained our approach to fulfil those requirements using the ideal visualisation process. Now that we have established our approuch, we can take this high-level design and turn it into a working implementation. Our implementation will consist out of two parts. The first part will be the implementable of a Visualisation Picker. This Visualisation Picker will help the user pick the correct visualisation type for his data. When the best visualisation type is chosen, the Visualisation Picker will then create a visualisation that will follow the guidelines and principles defined in Chapter 3. The second part of

the implementation is creating a new media type called Graphs in MindXpres. As explained in section 5.3, this can be done by creating a new plug-in. This plug-in will support all the visualisation types created by the Visualisation Picker and will make it possible to present these visualisations in MindXpres.

We start this chapter by defining some goals. These are some features that we wish to achieve with the two parts of our implementations. Next, we describe our Visualisation Picker, the libraries we used for the first part of our implementation and the choices we had to make. After that we will discuss the second part of our implementation, how we implemented this new media type into MindXpres.

6.1 Goals

We start this chapter by defining some goals that we want to achieve with our implementation. By listing these goals, we hope to give the reader an idea of which technical features and requirements we deemed important for our implementation.

6.1.1 Structure JSON Data Files

- Define a JSON Schema for validating the data files.
- Pick a validator for validating a data file with the JSON Schema.
- Create a basic data file that succeeds on validation.

6.1.2 GUI

- Allow reading and manipulating the properties of the data file.
- Calculate a visualisation type that suits the data best.
- Generate a visualisation for the data that follows guidelines and principles.
- Generate code to create visualisation in MindXpres.

6.1.3 Architecture

- Build a plug-in architecture for different visualisation types.

6.1.4 Plug-ins

- Demonstrate how to create a visualisation type in Visualisation Picker.
- Demonstrate how to create the same visualisation type in MindXpres.

6.2 Structure JSON Data Files

Before we can choose a visualisation type or create a visualisation, we first need to read the data from the data file. However, before we can read the data, we first have to know what the file format is and how the data is encode in the data file. In this section we will first choose which file format we want to support in the Visualisation Picker and we will also create our own Schema that can be used with a validator to validate the structure of the file.

The first choice we had to make in our implementation was which type of file format we will support. A file format, or file type, defines the way that the information is encoded for storage in a digital file. There are over thousands different file formats available, too many to implement them all. Also, the focus of this thesis is not on file format which means that we will only implement one format. In our implementations we wanted to use human readable and easy to parse file format.

The hardest decision was choosing between the file formats *JavaScript Object Notation*[1] (JSON) and *Extensible Markup Language*[2] (XML). They are both human readable, easy to parse and also have a lot of other advantages. However, we chose to only implement JSON due to following advantages:

- JSON is lightweight in comparison to XML. Having fewer characters makes it faster to parse.
- JSON is easier to handle and parse with JavaScript due to the fact that it can be loaded as a JavaScript object.

[1] `http://json.org/`, last accessed on 29/07/2014
[2] `http://www.w3schools.com/xml/default.asp`, last accessed on 29/07/2014

There are more then two advantages when using JSON, however these two are very interesting for our implementation: Some visualisation type can visualise large data set, for example a Histogram, in which we first have to calculates the frequency of values between certain value intervals and then visualises it. We also decided to implement the Visualisation Picker in JavaScript, which also makes it interesting to use JSON and not XML.

Now that we picked a file format, we also have to create a JSON Schema which makes it possible to validate the correctness of the file structure and to describe what is allowed, not allowed and required in the file's content. To create our Schema we first took a closer look at the Schema create by json-schema.org[1] and JSON-stat[2]. At first the JSON-stat seemed the most interesting basis for our Schema, but it can be pretty complicated to transform the data set into a format that is supported by JSON-stat. So we decided to create a new Schema and base it on that of json-schema.org by extending it to support a format better suited for information visualisation.

With our JSON-Schema we can check if the data file contains for example a title for the visualisation, a description, when it was created, the label of the x-axis and y-axis and so on. It is also possible to define if the data is univariate, bivariate, trivariate or hypervariate. We added a constraint that only these four terms can be used to describe the number of variables in the data set. It is for example also mandatory to add a data set into the data file, otherwise the file is useless to visualise data. At the end of this section we will create an example data file that satisfies the structure defined in our Schema.

What we still needed was a JSON validator written in JavaScript to validate the data file using our Schema. There are many good JSON validators available, for example tv4[3], JaySchema[4] and z-schema[5]. For our implementation we chose to use tv4, also called *Tiny Validator*, which is very easy to use. When the validator fails to validate the data file it gives a readable explanation why. After importing tv4 in our project the only code we still had to write is following code snippet:

[1] `http://json-schema.org`, last accessed on 29/07/2014
[2] `http://json-stat.org/`, last accessed on 29/07/2014
[3] `http://geraintluff.github.io/tv4/`, last accessed on 29/07/2014
[4] `https://github.com/natesilva/jayschema`, last accessed on 29/07/2014
[5] `https://github.com/zaggino/z-schema`, last accessed on 29/07/2014

```
(tv4.validate(dataFile, Schema)) ?
 console.log("Data file OK") :
 alert("Validation  error: " + JSON.stringify(tv4.error, null,
    4));
```

When loading a data file into the Visualisation Picker it is first checked by the validator. When tv4 validation returns true then the validation succeeded and the file structure is consistent with our Schema. When the validation fails, we show a message box that explains what is wrong with the structure of the data file. The next code snippet is an example of a data file that will pass the validation.

```
{
  "title": "Car Accidents Time Serie",
  "description": "Car accidents cases per year between
     1940 and 1970",
  "updated": "2014-07-30",
  "nr_sets": 1,
  "xaxis":"Years",
  "yaxis":"Accidents",
  "nr_encoding_variable":"bivariate",
  "encoding_variables":["nr_accidents", "year"],
  "data": [
    { "set1": [
        {
          "nr_accidents":120042,
          "year":1940
        },
        {
          "nr_accidents":130421,
          "year":1941
        },
        ...
      ]
    }
  ]
}
```

In the example above we defined the title and the description of the visualisation and also when it was last updated. We also defined that there is only one data set available in the file, however it is possible to have multiple data sets in one data file. This is interesting if you want to compare multiple data sets with each other. We also defined the labels for the axis and the type of the data by providing *nr encoding variable* with the correct term for the data type (e.g. univariate, bivariate, ...). Our Visualisation Picker can find the different variable by itself, but if we want it is possible to already define them in *encoding variables*. In *data* we can place the data set(s). In the example we shortened the data set by using three dots (...).

6.3 GUI

After a JSON data file passed validation, it is ready to be processed by our Visualisation Picker. Thanks to the JSON-Schema validation, we can safely assume that the file is valid and contains the necessary elements needed for the Visualisation Picker to create the best possible visualisation(s) of the data. In this section we will first discuss how the file is read and presented in the Visualisation Picker. Afterwards we will explain how the Visualisation Picker uses the data and some user input to choose one or more visualisation types to present the data.

6.3.1 Analysis and Manipulation of JSON file

We already mentioned how easy it is to read a JSON file in JavaScript. The only thing that needs to be done is load the file in the Visualisation Picker as a JavaScript object. This can be done by simply one line of code:

```
1 var dataObject = JSON.parse("locationToDataFile/nameDatafile");
```

Now that the file with its content is loaded in the Visualisation Picker as a JavaScript Object we can start by reading and analysing the data in the file. In the Visualisation Picker we provide a form in which the user can view the result of the analyse on the content in the data set(s). Figure 6-1 is an example of a filled in form after the file is loaded into the Visualisation Picker.

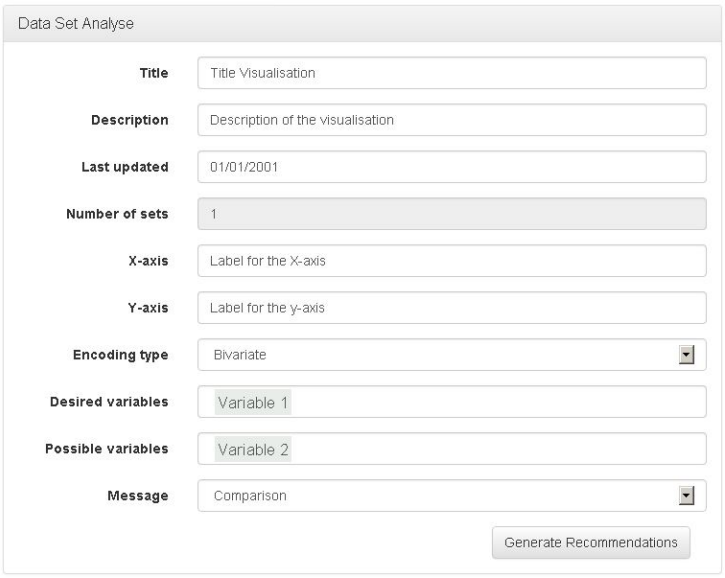


Figure 6-1: Form with the results of the analyse of a data file.

As you can see in Figure 6-1, the title, description, date of last update and the labels for the axis are automatically extracted from the data object. The user can also fill in those fields if these are not available in the JavaScript Object. It is also possible to change these fields, even if they are automatically extracted by the Visualisation Picker. The number of data sets can be defined in the data file itself, but the Visualisation Picker also uses some checks on the data object to know the number of available sets. The Visualisation Picker can also find the variables that are used in the data set and summarises these variables in the field 'Possible variables'. The user can drag and drop the variables from the 'Possible variables' to the 'Desired variables' field. The variables in the 'Desired variables' field will be used in the visualisation of the data. This way it is not mandatory to always use all the variables available in the data sets. The encoding type, or in other words the data type, is also automatically calculated by the Visualisation Picker, but can later on be changed by the user. A last feature in this step of the visualisation process is that it is also possible for the user to define the message he or she wants to bring with the data, by selecting one of the options in the 'Message' field. The types of messages that can be picked and why this is important for the visualisation of the data is explained in section 3.2.1.

For each variable in the data set Visualisation Picker will also try to correctly indicate the type of the variable. A variable can be quantitative, categorical or ordinal (section 2.2.6). It is possible for the user to change an incorrectly

indicated type of variable by adjusting the type in the form displayed in Figure 6-2. The type of a variable can also partially help decide the type of visualisation that will be used to present the data (section 3.4). If for example the variable for the X-axis is of the type quantitative and is presented by bars in the visualisation then it is probably a histogram. When the variable is categorical and presented with bars then a vertical bar graph is normally used.

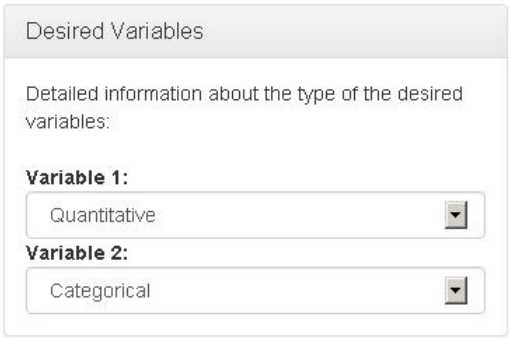


Figure 6-2: Form in which a user can change the types of the variables.

The goal of these two forms is to help the Visualisation Picker to make a correct choice in picking the best visualisation type for the data. All the information presented in the forms will be used the pick a suiting visualisation type and design. With this form the user can check if the data is correctly analysed before going to the next step: picking a visualisation type.

6.3.2 Picking a Visualisation Type

In this section we will discuss how the Visualisation Picker chooses a visualisation type that will probably present the data in the best possible way. The Visualisation Picker also needs some input from the user to pick a visualisation type. When a visualisation type is picked, our implementation will visualise the data in one or multiple visualisations of the type that is carefully chose.

The Visualisation Picker first presents the user with a list of visualisation types that are currently supported. Figure 6-3 shows how this list of supported visualisation types is presented to the user. With this form the user can select which types of visualisation the Visualisation Picker has to take into consideration while choosing a fitting visualisation for the data.

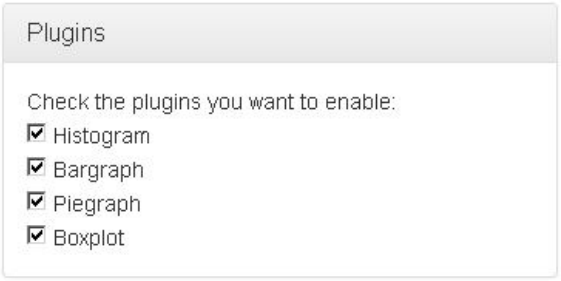


Figure 6-3: List of supported visualisation types

When the user presses the button *Generate Recommendations* in the form *Data Set Analyse*(Figure 6-1) the Visualisation Picker starts to search for the visualisation types that corresponds best with the information provide in the data file and previously described forms. When the Visualisation Picker has found the visualisation types that suits this information, it will provide the users with one or multiple new small forms for each of the found visualisations types. A possible result is visible in Figure 6-4.

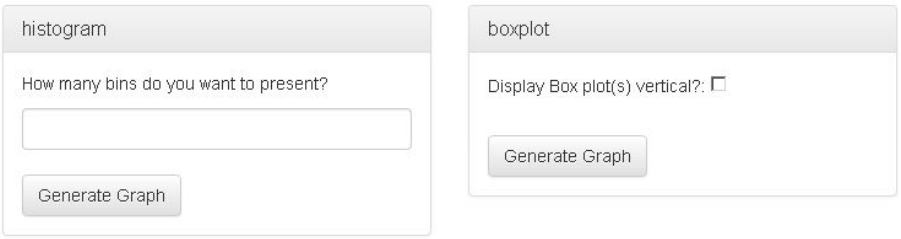


Figure 6-4: Visualisation types that are recommended by the Visualisation Picker for showing a distribution of an univariate data set with a quantitative variable.

Before the Visualisation Picker starts to visualise one of the visualisation types representing the provided data, some extra input of the user is sometimes required. In Figure 6-4 we can for example see the question in how many bins (data between the thick marks) the data has to be visualised. Afterwards the user can press the *Generate Graph* button which will start the creation of a visualisation.

6.3.3 Visualisation of the Data

When the file is loaded, the data is analysed and corrected, the recommended visualisation types are proposed and additional information has been filled in, then our implementation can start creating a visualisation of the data. The visualisation picker will create one or more visualisations of the same

visualisation type by using the data from the file and the extra information from the forms. It is possible that the Visualisation Picker will create more then one visualisation of the same type representing the same data. How these visualisation are presented to the user can be viewed in Figure 6-5. In this example we used the container that contains the generated visualisation for the visualisation type Histogram.

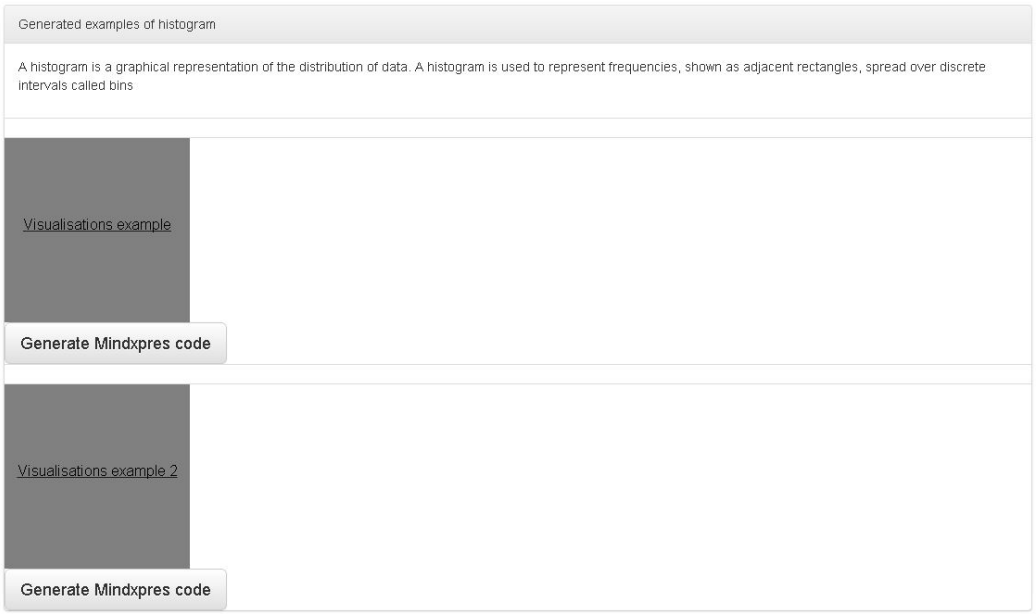


Figure 6-5: Containers for the generated visualisations

We replaced the example visualisation of the Histograms by smaller gray boxes to make the figure smaller. A fully illustrated examples will be available in Chapter 7. Every visualisation type also provides a small description in the left top corner about when this visualisation is commonly used.

The button *Generate MindXpres code* can be used to generate code for MindXpres. An example of this code can be found in Figure 6-6. When importing this code snippet into MindXpres it will present the same visualisation in the presentation tool, including the same data.

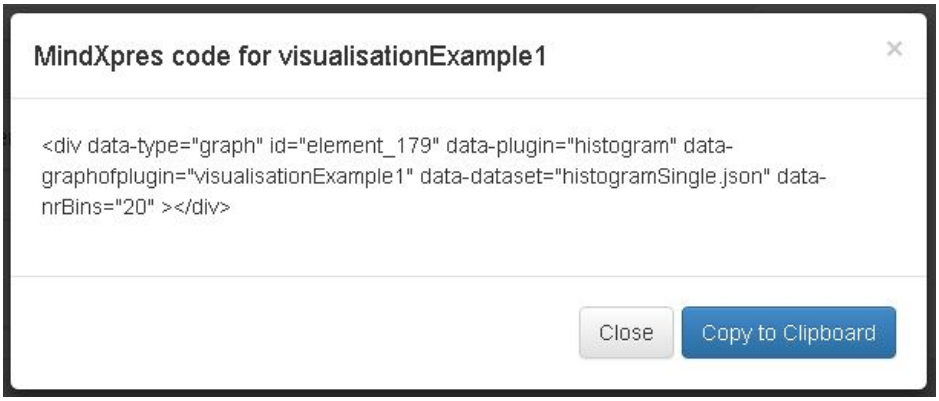


Figure 6-6: MindXpres code to present the same visualisation with the corresponding data set.

6.4 Architecture

We have seen how users can use the Visualisation Picker to help choose the visualisation type and create a visualisation of the data. We have also explained how you can transfer the visualisation of the data from the Visualisation Picker to MindXpres. These features are all presented in an intuitive graphical user interface to make the use of the visualisation process in the Visualisation Picker efficient and easy.

In this section we will discuss the architecture of our implementation. An overview of the architecture used for the Visualisation Picker is illustrated in Figure 6-7. In section 6.3 we already discussed a fraction of the architecture: the body of the HTML page. In the next subsections we provide more detail about what the other components of our implementation are and how they work.

6.4.1 RequirejS

RequireJS[1] is a JavaScript file and module loader that can be used in JavaScript environments (e.g. Node[2]) and in-browser use. RequireJS is one of the most popular modular programming frameworks for managing dependencies between modules. It is mostly used for larger applications that require a number of JavaScript files. It is possible to load JavaScript files in a application without using RequireJS but by using the <script>

[1] `http://requirejs.org/`, last accessed on 01/08/2014
[2] `http://nodejs.org/`, last accessed on 01/08/2014

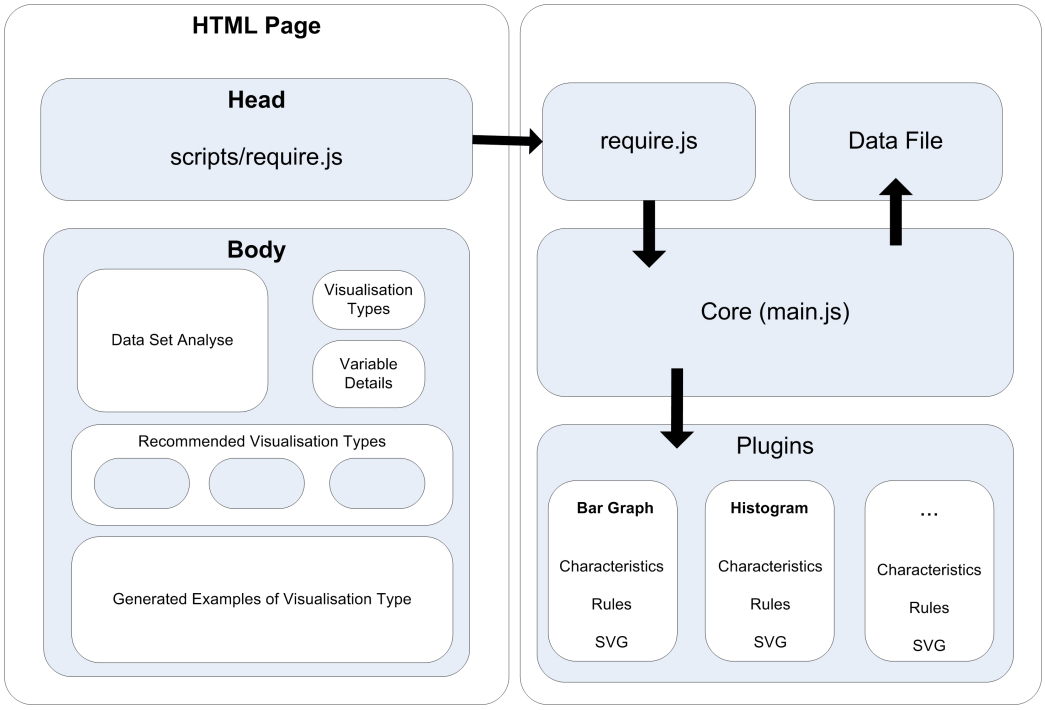


Figure 6-7: Architecture of the Visualisation Picker

tag. However, each of these files can potentially be dependent on other files, which means that the JavaScript files have to be loaded in the correct order. This can become very complicated when dealing with a lot of JavaScript files. That's where RequireJS comes into picture.

When using RequireJS, all code is wrapped in the functions require() or define(). The require() function has two parameters and is used to execute the code or functionality immediately. The first parameter specifies the dependencies and the second is an anonymous function. This anonymous functions takes as parameter(s) one or multiple objects that is used to call the functions that are inside the dependent file. An example is visible in following code snippet:

```
1 require(["dependentfile1", "dependentfile2"],function(
     dependentFile1, dependentFile2){
2   dependentFile1.executeFunction();
3   dependentFile2.executeOtherFunction();
4 });
```

As you can see, the dependent files are called dependentfile1 and dependentfile2 and have respectively the objects dependentFile1 and dependentFile2.

The define() function has only one parameter and is used to define modules that can be used in multiple locations.

```
1 define(function() {
2   console.log("Example of define()");
3 });
```

RequireJS waits until all the dependent modules or files are loaded before executing the code in these functions. By using these functions you are sure that the external files are fully loaded before our code, that is dependent of the these files or modules, is executed. However, even when each dependent file or module will start loading through asynchronous requests in the given order, there is no guarantee that the first file is loaded before the second file due to the asynchronous nature. RequireJS solves this by allowing us to use the shim config, in which the sequence of files which need to be loaded in correct order can be defined. Following code is an example snippet of this configuration option.

```
1 requirejs.config({
2   shim: {
3     'source1': ['isDependentOfThisFIle','andDependentOfThisFIle'
        ],
4     'source2': ['source1']
5   }
6 });
```

Before we can use RequireJS we have to reference require.js with a require call in the HTML page, which can be viewed in following code snippet. The data-main attribute defines the initialisation point of our Visualisation Picker, which is also the core of our implementation. The require.js is in our implementation located in the libs folder.

```
1 <script data-main="main" src="libs/require.js"></script>
```

6.4.2 Core

The JavaScript file main.js is the core of our implementation and provides most of the functionalities. The core is responsible for loading the data file, the Schema and the compiler. It is also responsible for loading the plugins, analysing and showing a summary of the data, and making this data editable for the user. After analysing the data from the data file and the user input in the forms, it also has to decide which visualisation types will present this

data and information in the best possible way, and propose these visualisation types to the user. When the user selected one of these proposed visualisation types the core has to present some example of the data presented in the selected visualisation type. At the end it also has to provide MindXpres code to create these visualisations in MindXpres.

The first task the core has to accomplish is to load all the necessary JavaScript files. By using the require() function of RequireJS we can be sure that all these files are loaded before the rest of the functionalities of the core are executed. The code snippet below is the biggest part of the loading process, which loads the JavaScript files: jquery.js (jQuery[1]), plugin_list.js, schema/standard.js and tv4.js (section 6.2), jquery.bootstrap.js[2] and d3.js[3].

```
1  require.config({
2      baseUrl: '',
3      paths: {
4          "jquery": "libs/jquery-2.1.0.min",
5          "text": "libs/requirejs-text",
6          "tv4": "libs/tv4",
7          "jquery.bootstrap": "libs/bootstrap.min",
8          "d3": "libs/d3.v3.min"
9      },
10     shim: {
11         "jquery.bootstrap": {
12             deps: ["jquery"]
13         }
14     }
15 });
16 require(['jquery', 'plugins/plugin_list', 'schema/standard', '
       libs/tv4', 'jquery.bootstrap', 'd3'],
17     function ($, plugin_list, schema, tv4, bootstrap, d3) {
18         // core functionality
19     });
```

JQuery is a small but fast and feature-rich JavaScript library that makes it easy to do HTML document traversal and manipulation, animation, event handling and few more features. The jquery.bootstrap.js contains a dozen custom jQuery plug-ins. One of these plug-ins is used to create the modal, visible in Figure 6-6.

[1] http://jquery.com/, last accessed on 01/08/2014
[2] http://getbootstrap.com/javascript/, last accessed on 01/08/2014
[3] http://d3js.org/, last accessed on 01/08/2014

D3.js is a JavaScript library for Document Object Model (DOM[1]) manipulation and uses HTML, Scalable Vector Graphics (SVG[2]) and Cascading Style Sheet (CSS[3]) to create visualisations of data. D3.js allows to bind data to a Document Object Model, and then apply data-driven transformations to the DOM. We will use this library to create our visualisations in the Visualisation Picker and MindXpres.

There are many JavaScript libraries available that are comparable with D3.js. A few of these JavaScript libraries are: Google Charts[4], CanvasJS[5], Raphaël[6] and JavaScript InfoVis Toolkit[7]. Why we decided to use D3.js instead of the other JavaScript libraries is explained in following reasons:

- To use Google Charts you always need an internet connection. In order to make a visualisation you first have to make a connection with the `http//www.google.com/jsapi`. Their terms of service[8] do not allow to download the google.load or google.visualisation code, which are needed to create visualisations. D3.js can be used without the need of a working internet connection.
- CanvasJS creates canvas-based visualisations. This makes it less suited for creating dynamic and interactive visualisation. Another disadvantage is that when zooming in on a visualisation you will notice that the visualisation gets pixelated. MindXpres has a Zoomable user interface which makes it possible to zoom in on a visualisation. D3.js creates vector-based visualisations which does not have this problem.
- After experimenting with the Raphaël library for a while we noticed that it still has more bugs than the other tools we have tested. Some of these bugs[9] are even a few years old and still not fixed.
- JavaScript InfoVis Toolkit provides fewer different types of visualisation than D3.js. And a few of these visualisations that can be created, are not really suited for visualising data in a presentation.

[1] `http://www.w3.org/DOM/`, last accessed on 02/08/2014
[2] `http://www.w3schools.com/svg/`, last accessed on 02/08/2014
[3] `http://www.w3schools.com/css/`, last accessed on 02/08/2014
[4] `https://developers.google.com/chart/`, last accessed on 02/08/2014
[5] `http://canvasjs.com/`, last accessed on 02/08/2014
[6] `http://raphaeljs.com/`, last accessed on 02/08/2014
[7] `http://philogb.github.io/jit/`, last accessed on 02/08/2014
[8] `https://developers.google.com/chart/terms`, last accessed on 02/08/2014
[9] `https://github.com/DmitryBaranovskiy/raphael/issues`, last accessed on 02/08/2014

D3.js is released under a BSD licence, which means that we may use, modify and adapt D3;js for non-commercial and commercial use. It is also perfect to use in an application with a zoomable user interface such as MindXpres and bugs[1] are almost instantly fixed. The abbreviation D3 refers to the full name Data-Driven Documents, in which the data is provide by the user, the documents are web based and driving means that, in a sense, D3.js connects the data to the documents.

Next to leading files, the core is also responsible for analysing the data file and displaying all the obtained information in a form (Figure 6-1) visible for the user. The information in this form can be adapted by the user and will then be used by the core to compare it with the properties of the visualisation type. Afterwards the Visualisation Picker picks the best corresponding visualisation types and presents the data in one of these visualisation types. It is also responsible for generating the MindXpres code so that the visualisation created in the Visualisation Picker is presented the same in MindXpres. This process was already explained in more detail in section 6.3.

6.4.3 Plug-in Architecture

Information visualisation with different visualisation types is a major component of the Visualisation Picker, and is mostly driven by plug-ins. Every plug-in in the Visualisation Picker represents a different visualisation type. Each plug-in is responsible for holding the properties of a visualisation type and helps the core to present the data in their own visualisation type. This way, a developer can easily create a new plug-in representing a visualisation type without having to know what happens in the other plug-ins. This makes the Visualisation Picker easily extendible and everyone with programming skills and skills in information visualisation can add new visualisation types.

As you may have notice, the core also loads a JavaScript file called plugin_list.js. This file, of which the content is visible in the next code snippet, contains a list of all the plug-ins available at the moment. As you can see, each plug-in is named as the visualisation type that they represent. Every time a new plug-in is created it needs to be added to this list.

[1] `https://github.com/mbostock/d3/issues`, last accessed on 02/08/2014

```
define(function () {
    plugin_list = [
        "histogram",
        "bargraph",
        "piegraph",
        "boxplot" ];
    return plugin_list;
});
```

JavaScript does not provide any functionalities to read filesystems, so to achieve our plug-in mechanism we use naming conventions. Plug-ins are contained into individual folders and have to includes minimum one JavaScript file called plugin.js. This file contains the properties of the visualisation type and the functionality to visualise the data that will be presented by the core. The name of the folder corresponds with the visualisation type that the plug-in will represent. In section 6.5 we describe how to create a plug-in for the Visualisation Picker.

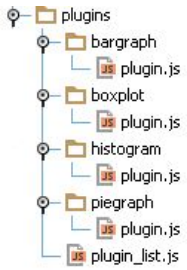


Figure 6-8: The plug-in architecture in Visualisation Picker

The core is responsible for loading al the plug-ins that are listed in plugin_list.js. Following code snippet demonstrates that this is also done with the require() function of requireJS. We push all the plug-ins as objects in an array so that we can later on access each plug-in which contains the visualisation types properties and functionalities to create the visualisation.

```
    $.each(plugin_list, function (index, plugin) {
        require(['plugins/' + plugin + "/plugin"], function (
            module) {
             plugins.push(module);
        });
    });
```

6.5 Plug-ins

In this section we will demonstrate how to create new types of visualisations in the Visualisation Picker and MindXpres. Plug-ins are implemented differently in the Visualisation Picker than in MindXpres, and will be discussed independently. We will mention the most important details about the structures of the plug-ins. However, the description about the structure will be fairly brief, so we want to refer to the extended template that provides the necessary information to create plug-ins for implementing a new visualisation type. With this template it is easier for a user to start implementing new visualisation types.

6.5.1 New Visualisation Type in Visualisation Picker

The next code snippet shows the global structure of a plug-in in the Visualisation Picker. It is mandatory to have the function *init*, *minimumRequirements*, *rules* and *generateGraphs* in the plug-in file. Without these functions the newly created visualisation type defined in this plug-in will not work in the Visualisation Picker. In this section we will briefly discuss the purpose of each function in our plug-in implementation.

```
1  define([], function () {
2
3      var init = function () {
4        ...    };
5
6      var minimumRequirements = function (datasetDetails,
           formDetails) {
7        ...    };
8
9      var rules = function (ruleset, datasetDetails, formDetails)
           {
10       ...    };
11
12     var generateGraphs = function (ruleset, datasetDetails,
           formDetails) {
13       ...    },
14
15     return {
16         init: init,
17         rules: rules,
18         minimumRequirements: minimumRequirements,
19         generateGraphs: generateGraphs
20     };
21 });
```

Init The init function is loaded at start-up of the Visualisation Picker. This function is responsible for providing the core with the name, description and the properties of the visualisation type that the plug-in represents. One of these properties is the data type that can be used with the visualisation type, for example univariate, bivariate, trivariate and hypervariate data. Another property is for what message type the visualisation type is normally used. This can be comparison, trend over time, part-to-whole, distribution, deviation, correlation, ranked.

minimumRequirments When the user presses the button *Generate recommendations* in the GUI, the core start to compare the properties of the visualisation types with the information from the dataset and the form. If the core found some visualisation types that will work well for visualising the data it executes the function minimumRequirements from the plug-ins that represent these visualisation types. The developer can define some test in the function minimumRequirements to check if the data is really compatible or will be displayed correctly with the visualisation type the user wants to create. For example, if the developer created the visualisation type Pie chart he or she can first check with an algorithm if the data does not contain more then 4 values and every slice, that will represent a value, has an angle between 25 and 40 degree. If this function returns the boolean value true to the core then the proposed visualisation type will be visible in the GUI. In other case this visualisation type will not be visible because the algorithms implemented by the developer concludes that visualisation type does not suit the data.

rules When the list with proposed visualisation types is created and displayed (Figure 6-4) you can notice that there are extra input fields the user has to fill in before the Visualisation Picker generates the visualisations of the data. A developer of a plug-in can define some questions in the rules functions of his/her plug-in that will be displayed in the form of the developer's proposed visualisation type. When the developer's visualisation type is picked afterwards to visualise the data, he or she can use this input to make the visualisations better or decide which of the visualisations of the same types are useful.

generateGraphs This function is responsible for creating one or multiple visualisations of the same data and visualisation type. The developer can for example decide to create two visualisation, one with colour and one without. Or a visualisation of the data without thick marks and another with thick

marks and so on. This function is also responsable for creating the MindXpres code that the user can copy into MindXpres to create the same visualisation. To create the visualisations in this function, the developer has access to the data file, the answers on the questions in the rules set and the information filled in the forms by the user.

6.5.2 New Visualisation Type in MindXpres

In MindXpres we use almost the same plug-in architecture as the Visualisation Picker. To register a new visualisation type for visualising data we have to follow the same steps as in section 6.4.3. However, we did have to write a different core for loading all the visualisation plug-ins in MindXpres. We have seen in Chapter 5 that when we want to add a new media type in MindXpres, we have to add it as a plug-in. In MindXpres the plug-ins are implemented as JavaScript bundles which are folders that contains JavaScript files, CSS files, JavaScript libraries and so on. We created a new media type called *graph* in MindXpres and placed our plug-in architecture into the new JavaScript bundle called graphs. We created an adapted core called graph.js and also placed it in this bundle. This adapted core will be responsible for visualising the created visualisation of the Visualisation Picker into MindXpres. The visualisation plug-ins we created in MindXpres also have a different structure then the plug-ins used in the Visualisation Picker:

```
1  define([], function () {
2
3      var init = function () {
4          return ["NameOfVisualisationType"];
5      };
6
7      var generateGraphs = function (datasetDetails, svgElement,
           graphs) {
8        ..
9      }
10
11     return {
12         init: init,
13         generateGraphs: generateGraphs
14     };
15 });
```

The init function in the visualisation plug-ins we created for MindXpres only contains the name of the visualisation type. It does not provide any other properties of the visualisation type, due to the fact that we only needed the other properties for selecting the best visualisation type for a particular data set. However, in MindXpres it is already defined which visualisation type we will use for the data set. The information about which visualisation type suits the data best can be found in the MindXpres code that is generated by the Visualisation Picker. The name of the visualisation type that is returned by the init function is used to identify the plug-in.

The purpose of the MindXpres code created by the Visualisation Picker is to place it in the HTML document of the presentation. When the visualisation library of MindXpres starts transforming the HTML document into a zoomable user interface it will stumble upon the code created by the Visualisation Picker. Our visualisation core, which is also a plug-in in MindXpres, starts searching the plug-in of the visualisation type that is mentioned in that code fragment and then calls the function generateGraph of that plug-in. This function generates a SVG element that will be displayed in MindXpres' zoomable user interface.

7 Use Case

In Chapter 6 we discussed how we implemented the information visualisation guidelines and principles in our prototype called Visualisation Picker and claimed that we solve many of these issues that are inherent in the functionality for creating visualisation in common presentation tools. Due to the time restriction we could not evaluate our solution by means of user trials, in which we could see if our solution is effective, usable and efficient. In this chapter we wish to validate our implementation by means of a use case. By walking through two scenarios that span the process of picking a visualisation type and creating a visualisations that follows the design aesthetics, we wish to demonstrate the quality of the visualisation that is created by our implementation.

7.1 The Scenarios

First of all, we want to define the scenarios that will be used in this section to demonstrate our implementation. These scenarios can be used to illustrate how we respond to the needs and (limited) skill of the user to create a good visualisation. In the first bigger scenario we will work with an univariate data file that the user wants to present to the audience. At first the user does not know in which type of visualisations the data can be presented in. What the user does know is the message he or she wants to show with the data, in this case the distribution of the data values. In the smaller second

scenario the user has three sets of test results in a bivariate data file and wants to present to the audience which set has the highest correlation. With these scenarios we hope to illustrate the ease of use and extensibility of our implementation the Visualisation Picker.

7.2 First Scenario

In the next scenario we will demonstrate how easy it is to visualise the data of a data file in the Visualisation Picker. The used data file contains all the necessary information and can be used directly by our implementation to pick a visualisation type and generate a visualisation of the data. The user will have the choice to pick between multiple visualisation types and also between multiple visualisation of these types. Afterwards we also demonstrate how to create the identical visualisation in MindXpres.

7.2.1 Sneak Peak in Data File

First we will take a sneak peak in the data file to see which data the user wants to present. Following code snippet shows the content of the data file, which contains test results of an enrolment exam. Most of the data in the set itself is left out in this code snippet, but clearly shows an example of how the unanimous results are stored in the file. In this scenario the user want to show a distribution of these result to the audience. Note that we size the code down to the relevant parts and the three dots (...) indicate the irrelevant code that is left out.

```
{
    "title": "Results Enrolment Exam",
    "description": "Results enrolment exam with more
       then thousand unanimous individuals results",
    "updated": "2014-07-30",
    "nr_sets": 1,
    "xaxis":"Results",
    "yaxis":"Frequency",
    "nr_encoding_variable":"univariate",
    "encoding_variables":["result"],
    "data": [
        { "set": [
            {
                "result":47.14
            },
```

```
15                {
16                    "result":71.65
17                },
18            ...
19          ]
20      }]
21  }
```

7.2.2 Analysing Data File

When loading the data file in the Visualisation Picker it is instantly analysed and the results are nicely displayed in the forms, displayed in Figure 7-1. The form at the right side, called Desired Variables, also displays the type of the variable. The variable *result* in the data file is indeed of quantitative nature, which means that our implementation analysed this correctly.

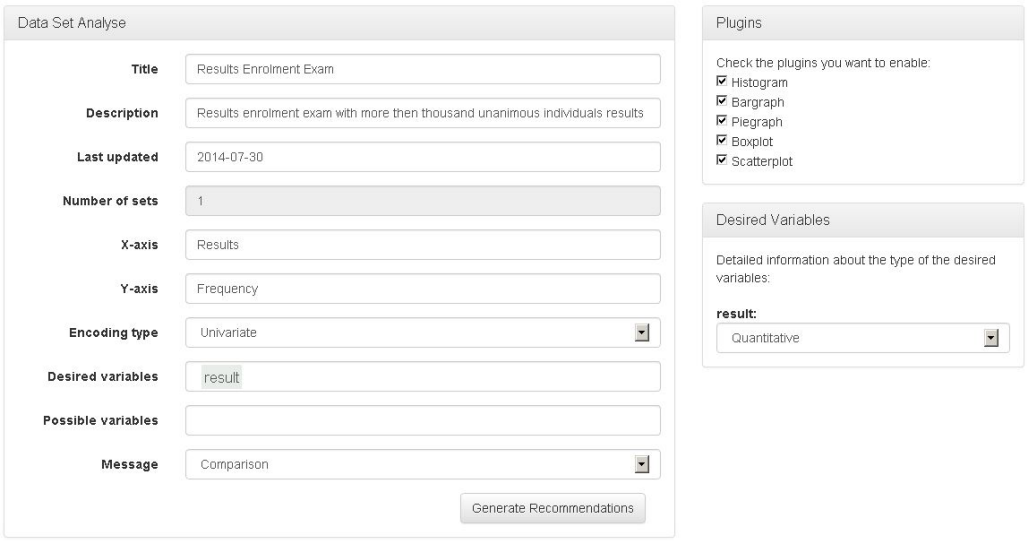


Figure 7-1: Results analyse data file

However, at the moment the message type *comparison* is picked while the user want to show the distribution in the data. Before the user presses on the 'Generate Recommendations' button for generating the visualisation types that will suit his data and message, he or she first has to change the message type to *distribution*. Another message can be picked from the dropdown list that contains the different message types.

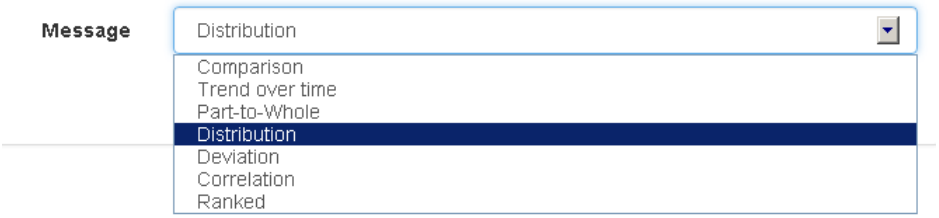


Figure 7-2: Dropdown list with message types

7.2.3 Generated Recommended Visualisation Types

The Visualisation Picker carefully selected one or more visualisation types and displays them to the user. For the data and the message type of this scenario has the Visualisation Picker proposed two visualisation types: Histogram and Box plot. When the user is not sure which visualisation type he or she wants to use, it is possible to try them both.

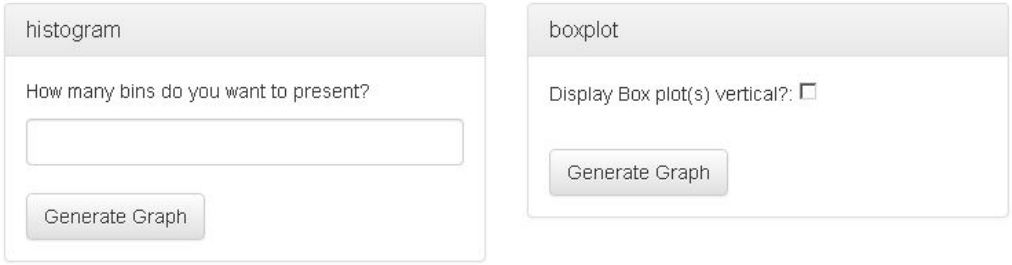


Figure 7-3: Two different visualisation types recommended by the Visualisation Picker

7.2.4 Create Box Plot

When the user wants to display the data in a Box plot, he or she just has to press 'Generate Graph' in the Box plot form. It is possible to generate a vertical and horizontal Box plot. In this scenario the user went for a vertical Box plot, which is displayed in Figure 7-4.

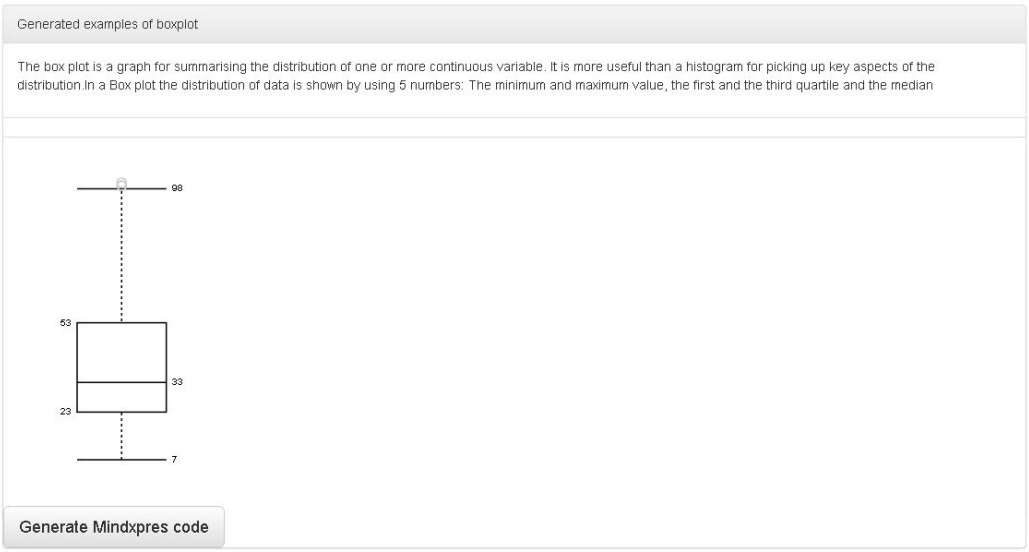


Figure 7-4: Box plot generated by the Visualisation Picker

On the visualisation we see that the first and third quartile are displayed by the 2 horizontal borders of the rectangle. The median is the horizontal line dividing the rectangle. The white dots are the *suspected outliers* and the black dots, which are not visible with this data set, are the *outliers*. The horizontal lines at the ends of the vertical lines is the minimum and the maximum.

7.2.5 Create Historgram

To display the data in a Histogram, the user first has to state in how many bins he or she wants to present the data. In this scenario the user picked ten and pressed 'Generate Graph'. As result, the Visualisation Picker generated two Histograms with a different design.

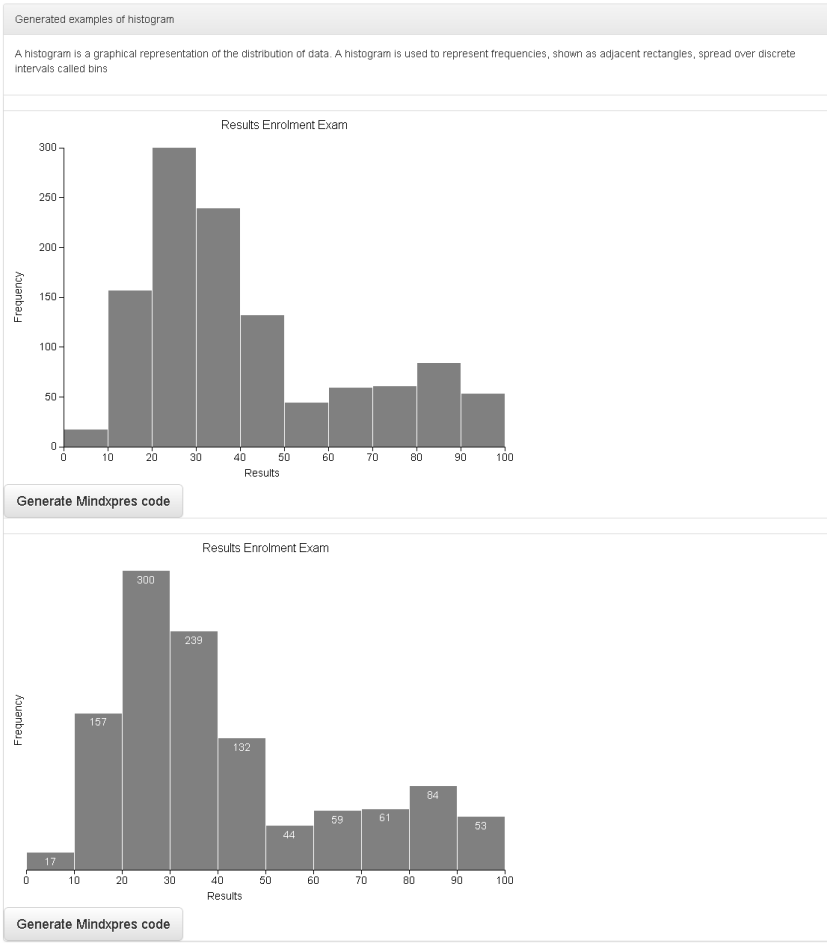


Figure 7-5: Box plot generated by the Visualisation Picker

7.2.6 MindXpres Code

By pressing the 'Generate MindXpres code' the user will get a code snippet that can be used in MindXpres to create the identical visualisation. The user can copy the code to the clipboard and then paste it in the HTML page of the MindXpres presentation.

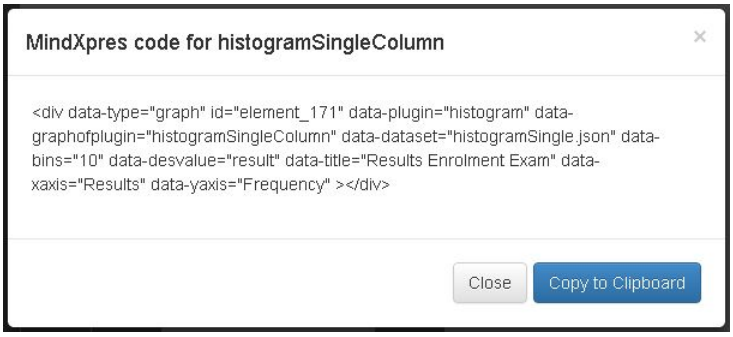


Figure 7-6: MindXpres code for creating the identical Histogram in MindXpres

Figure 7-7 shows what happens when the code snippet in Figure 7-6 is placed in a slide component in MindXpres. The identical visualisation created by the Visualisation Picker is now visible in a slide in MindXpres. How to create a presentation in MindXpres is described in detail in the master thesis [36] of Reinout Roels.

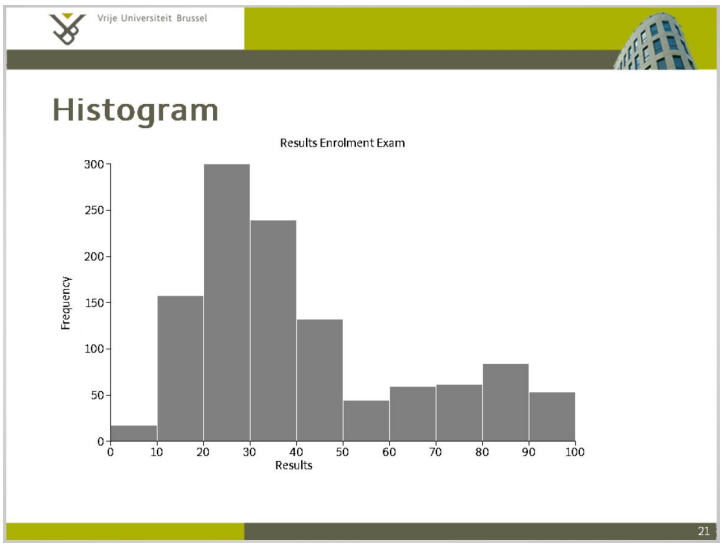


Figure 7-7: Histogram displayed in a slide component in MindXpres

7.3 Second Scenario

In this scenario we show what happens when a user gives an incomplete data file to the Visualisation Picker. This data file contains multiple data sets that will be displayed in the same visualisation, which is picked and designed by our implementation.

7.3.1 Sneak Peak in Data File

The code snippet below is also a possible data file provided by a user. As we can see, this data file contains three sets in total. Some extra information is not defined, for example: the title of the visualisation, the number of sets in the data file, the type of the data and the variables (encoding_variables) the user wants to present in the visualisation.

```
{  "description": "Test results of three different
   tests",
   "updated": "2014-08-07",
   "xaxis": "X Values",
   "yaxis": "Y values",
   "set_names": ["Set 1", "Set 2", "Set 3"],
   "data": [{"set": [
                  {
                      "Results_A": 89,
                      "Results_B": 18
                  },
                      ...
              ]},
               {"set": [
                  {
                      "Results_A": 91,
                      "Results_B": 24
                  },
                  ...
              ]},
               {"set": [
                  ...
              ] }]
}
```

7.3.2 Analysing Data File

Figure 7-8 shows the analysis result of the datafile by the Visualisation Picker. The title and the variables the user wants to show in the visualisation are not filled in. However, our implementation discovered how many sets of data there are available in the data file and that the type of the data is bivariate. The possible variables that the user can present are also discovered by the Visualisation Picker.

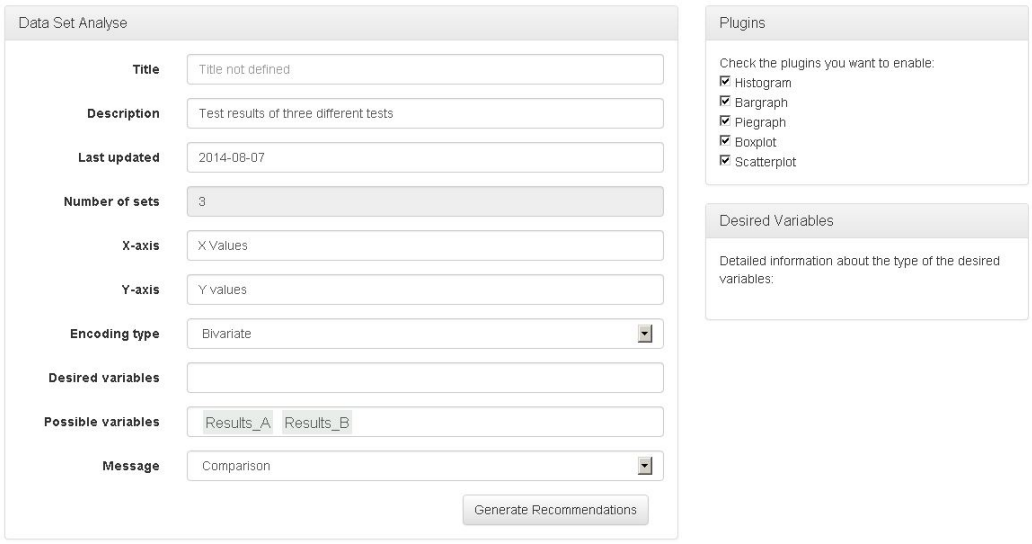


Figure 7-8: Results analyse data file

It is possible to drag and drop the possible variables into the box of the desired variables. All the variables dragged into the desired value box will be used in the visualisation. This way the user can decide which variables he or she wants to present in the data visualisation. The user also has to type the title for the visualisation in the Title box: *Correlation in Test Results*.

Figure 7-9: Drag and drop the desired variables

7.3.3 Generated Recommended Visualisation Types

In this scenario the user places all the possible variables in the desired variables box. As message type the users picks correlation and presses the button 'Generate Recommendations'. The only resulting visualisation type is the Scatter plot. The extra information this visualisation type needs is which variable has to be placed on the x-axis.

Figure 7-10: One visualisation type recommended by the Visualisation Picker

7.3.4 Create Scatter Plot

After stating that the *result_A* variable has to be placed on the x-axis, the Visualisation Picker creates the Scatter plot visible in Figure 7-11. When we take a closer look to the visualisation we see that only 'Set 3' has a correlation, to be more precise: a low positive correlation.

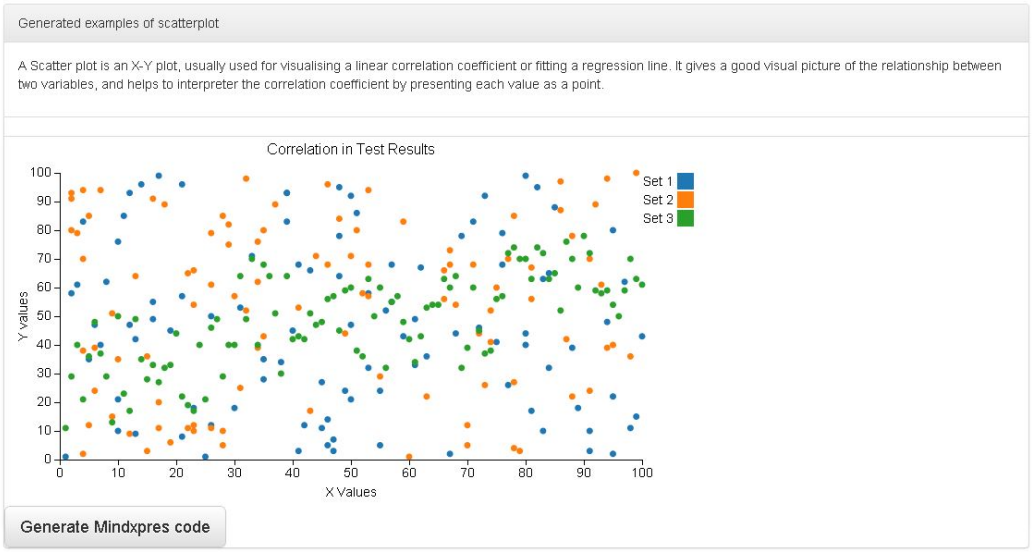


Figure 7-11: Scatter plot generated by the Visualisation Picker

7.4 Conclusion

In this chapter, we have demonstrated how to use Visualisation Picker in combination with MindXpres by means of two simple scenarios. As you have probably noticed, Visualisation Picker tries to handle as much as possible automatically. The user only needs to provide a data file with correctly formatted data and fill in some forms along the way, while other tedious work is done by the Visualisation Picker. Note, we only demonstrated three visualisation types in this chapter, when more visualisation types are integrated it is possible to have more recommended visualisation per dataset and message type. In these scenarios we did not present how to create a new visualisation type as a plug-in in the Visualisation Picker. However, we do provide the user with a template plug-in in our implementation and already provided some information about creating plug-ins in Section 6.5.

8 Conclusions and Future Work

The exploration of larger data sets is an important but also difficult problem in presentation tools. Recent research in information visualisation resulted in various techniques to address this issue. Visualising data has a high potential and many applications, such as presentation tools, can use these information visualisation techniques for improved visualisations of data. In this final chapter, we would like to recapitulate a number of contributions we made in information visualisation techniques for presentation tools. We will also discuss some future directions of our Visualisation Picker and improvements that can be made.

8.1 Contribution

Within this thesis, we have provide a set of principles and guidelines that can help solve active common problems in the domain of information visualisation, as well as data visualisations within presentation tools. We have illustrated how visualising data is handled in common presentation tools and how they partake in solving the active problems of information visualisation in their own tool. We also demonstrated in our implementation how data visualisation can be adapted in presentation tools to tackle such problems by picking the best visualisation type, utilising efficient visualisation aesthetics and avoiding techniques that make the visualisation misleading.

Recapitulating, this thesis has made the following contributions:

- We conducted a literature study in order to identify problematic areas of information visualisation in common presentation tools. We discussed different types of visualisation types that are commonly used to present data and studied the challenges that are present in information visualisation. This gave us an extensive look in how we create visualisation nowadays and how all these challenges lead to three problematic topics.

- Using the insights gained from the literature study, we broadened our view in order to explore guidelines, principles and tricks that might help improve the quality of data visualisations in presentation tools. By combining our research in guidelines, principles and tricks together with our investigation in the quality of the data visualisation provided by presentation tools, we were able to identify the limitations of information visualisation in these presentation tools.

- By implementing a new media type in the presentation tool MindXpres that makes it possible to visualise data visualisation, we provided the creator of a MindXpres presentation with the possibility to create efficient, effective and interactive visualisations.

- To help the user create a good visualisation that present the data in the best possible way, we also created an extensible functional prototype called the Visualisation Picker. This prototype will guide the user through the process of picking the best visualisation type that suits the data and create visualisations that follows the guidelines and principles defined in this thesis. The Visualisation Picker can be used as basis for future research of information visualisation in presentation tools.

- As already stated, Visualisation Picker is an extensible prototype. This means that it is possible to extend our prototype with new types of visualisations and design aesthetics. Many presentation tools only provide a limited set of visualisation types and also limit their number of different design. With Visualisation Picker and MindXpres we give the user the possibility to add new visualisation types or create new design for the visualisations. No more need for workarounds to visualise the data in a visualisation type that is not implemented in the presentation tool by the developers, now the user can create or add visualisation types and design aesthetics to our tool.

8.2 Future Work

In this thesis we have presented the initial version of the Visualisation Picker. The limited time available for this thesis did not suffice to bring the Visualisation Picker to its full potential. Still, with the limited time we had at our hands, we provide a solid foundation, making sure that future research can build on our extensible core and plug-in architecture. A lot of time was spent on these architectures to offer a high degree in usability and extensibility. In this section we suggest some possible expansions and improvements in the functionalities for the Visualisation Picker.

8.2.1 Analysing Data Files

Our implementation can read JSON files and displays the information available in these files to the user with a form. However, the actual data in the data sets is not shown to the user. It could be interesting to display this data to the user in another format than that is defined in the Schema for JSON files. A possibility is to show the data in a format used by many spreadsheet application such as Microsoft Excel. This could give the user a better idea about the data in the file without having to open the file within another application. The prototype also supports only JSON files, it could be interesting to support other formats, due to the fact that not all the data in the world is stored in JSON files.

It would also be interesting to extend the analysis of the data with artificial intelligence to get more information from the data. Jock Mackinlay's work [32] already provides a lot of research about how to use AI for creating effective graphical visualisations for a wide variety of information. Combining our work could further improve the process of picking and creating the best visualisations for any data set.

8.2.2 Recommended Visualisation Type

At the moment, when the Visualisation Picker proposes multiple visualisation types to the users they are displayed in random order. It would be interesting to order the visualisation types in function of efficiency and effectiveness in displaying the data. Weighted rules and requirements could for instance return a total score which could be used to establish an order in the recommended visualisation types.

Also, the visualisation types that are not suitable for the data for one or multiple reasons are hidden for the user. Providing an explanation to the user why these visualisation types are not suitable for the data could be helpful. With this information they could learn more about picking the best visualisation types and principles in design aesthetics, which could actually improve their skills in information visualisation.

8.2.3 Plug-ins

Due to time restrictions implied by this thesis, we focused on creating plug-ins for the more common visualisation types. However, not all the data can be displayed by these common visualisation types. Creating and extending plug-ins for the Visualisation Picker and MindXpres will improve the process of picking the best visualisation type and improve the overall aesthetic quality of data visualisation.

8.2.4 Evaluation

Time restrictions also prevented us to evaluate the tool with real users. One potential conclusion that might come out of such an evaluation is that dialogue specifications could be adapted to the different skill level in information visualisation of the users. However, the techniques for picking visualisation types and the guidelines and principles in design aesthetics are based on research in the field of information visualisation, which makes us confident that the Visualisation Picker provides a high level of quality in visualisations of data.

Bibliography

[1] H. G. Begley and D. Shere. A History Of Dishonest Fox Charts. `http://mediamatters.org/research/2012/10/01/a-history-of-dishonest-fox-charts/190225`, Oktober 2012. Accessed on: 28.5.2014.

[2] J. Bertin. *Semiology of Graphics: Diagrams, Networks, Maps*. Madison, WI, University of Wisconsin Press,, 1st edition, November 1983. Translated by William J. Berg.

[3] S. Card, J. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Sagebrush Education Resources, January 1999.

[4] W. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American statistical Association*, 79(387):531–554, September 1984.

[5] W. S. Cleveland. *Visualizing Data*. Hobart Press, 1st edition, March 1993.

[6] W. S. Cleveland. *The Elements of Graphing Data*. Hobart Press, 2nd edition, Oktober 1994.

[7] N. Cowan. The magical mystery four how is working memory capacity limited, and why? *Current Directions in Psychological Science*, 19(1):51–57, 2010.

[8] A. Edmunds and A. Morris. The problem of information overload in business organisations: a review of the literature. *International Journal of Information Management*, 20(1):17–28, 2000.

[9] W. Eells. The relative merits of circles and bars for representing component parts. *Journal of the American Statistical Association*, 21(154):119–132, 1926.

[10] B. Ennehar, O. Brahim, and T. Hicham. An Appropriate Color Space to Improve Human Skin Detection. *INFOCOMP Journal of Computer Science*, 10(1):2–11, 2010.

[11] S. Few. Tapping the power of visual perception. *Visual Business Intelligence Newsletter*, pages 1–8, 2004.

[12] S. Few. Effectively communicating numbers: Selecting the best means and manner of display. *Perceptual edge*, November 2005.

[13] S. Few. *Now You See It: Simple Visualization Techniques for Quantitative Analysis*. Analytics Press, 1st edition, April 2009.

[14] S. Few. Sometimes We Must Raise Our Voices. *Perceptual Edge*, pages 1–9, February 2009.

[15] S. Few. *Show Me the Numbers: Designing Tables and Graphs to Enlighten*. Analytics Press, USA, 2nd edition, 2012.

[16] J. Flannery. The relative effectiveness of some common graduated point symbols in the presentation of quantitative data. *The Canadian Cartographer*, 8(2):96–109, 1971.

[17] J. D. Foley, S. K. Feiner, J. F. Hughes, and A. V. Dam. *Computer Graphics: Principles and Practice*. Addison-Wesley, 2nd edition, June 1990.

[18] M. Friendly. *Visualizing Categorical Data*. SAS Institute, September 2000.

[19] M. Friendly and D. J. Denis. A Quick Illustrated History of Visualisation. `http://data-art.net/resources/history_of_vis.php`, May 2014. Accessed on: 28.4.2014.

[20] R. Gaskins. Sample product proposal: Presentation graphics for overhead projection. August 1984. Retrieved on 16 July 2014 from http://www.gbuwizards.com/files/gaskins-original-powerpoint-proposal.pdf.

[21] M. Handford. *Where's Waldo?* Candlewick, act ina re edition, April 2007.

[22] J. A. Harrell and V. M. Brown. The World's Oldest Surviving Geological Map: The 1150 BC Turin Papyrus from Egypt. *The Journal of Geology*, (100):3–18, 1992.

[23] M. Harrower and C. Brewer. Colorbrewer. org: an online tool for selecting colour schemes for maps. *The Cartographic Journal*, 40(1):27–37, 2003.

[24] J. Heymans. A WYSIWYG Template Authoring Solution for the MindXpres Presentation Tool. Master's thesis, Vrije Universiteit Brussel (VUB), 2013.

[25] I. G. D. Huff. *How to Lie with Statistics*. Norton, W. W. & Company, Inc., twenty-second printing edition, January 1954.

[26] W. Jansen. Neurath, Arntz and ISOTYPE: The Legacy in Art, Design and Statistics. *Journal of Design History*, 22(3):227–242, 2009.

[27] Joey. Data Looks Better Naked. `http://darkhorseanalytics.com/blog/data-looks-better-naked/`, August 2013. Accessed on: 2.6.2014.

[28] D. Kelly, J. Jasperse, I. Westbrooke, and New Zealand Department of Conservation. *Designing Science Graphs for Data Analysis and Presentation: The Bad, the Good and the Better*. DOC Technical Series. New Zealand Government - Department of Conservation, 2005.

[29] T. Koch. The Map as Intent: Variations on the Theme of John Snow. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 39(4):1–14, 2004.

[30] J. G. Koomey. *Turning Numbers into Knowledge: Mastering the Art of Problem Solving*. Analytics Press, 2nd edition, April 2008.

[31] B. Lindgren. *Statistical Theory*. Chapman and Hall/CRC, 4 edition, Oktober 1993.

[32] J. Mackinlay. Automating the design of graphical presentations of relational information. *Transactions on Graphics (TOG)*, 5(2):110–141, 1986.

[33] M. Nyman. *Four Colors/One Image*. Peachpit, September 1993.

[34] R. Parikh. How to Lie with Data Visualization. `http://data.heapanalytics.com/how-to-lie-with-data-visualization/`, 2014. Accessed on: 16.8.2014.

[35] K. Pearson. *The Life, Letters and Labours of Francis Galton*. HardPress Publishing, January 2013.

[36] R. Roels and B. Signer. *MindXpres: An Extensible Content-driven Cross-Media Presentation Tool.* PhD thesis, Vrije Universiteit Brussel, 2012.

[37] R. Roels and B. Signer. MindXpres - An Extensible Content-driven Cross-Media Presentation Tool. In *Proceedings of the 27th BCS Conference on Human Computer Interaction (HCI 2013)*, London, United Kingdom, 2013.

[38] R. Roels and B. Signer. MindXpres: An Extensible Content-driven Cross-Media Presentation Platform. In *Proceedings of WISE 2014, 15th International Conference on Web Information System Engineering*, Thessaloniki, Greece, October, 2014.

[39] G. Scagnetti. The Diagram of Information Visualization. *The Parson Institute for Information Mapping*, 4(4):1–8, 2012.

[40] C. J. Schwarz. A Short Tour of Bad Graphs. Technical report, Department of Statistics and Actuarial Science, Simon Fraser University, 2006.

[41] B. Shneiderman. Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99, 1992.

[42] D. J. Simons and D. T. Levin. Change Blindness. *Trends in Cognitive Sciences*, 1(7):261–267, 1997.

[43] I. Spence and S. Lewandowsky. Graphical Perception. Chapter 1. In J. Fox & J. S. Long (Eds.). *Modern methods of data analysis*, pages 13–57, 1990.

[44] R. Spence. *Information Visualization: Design for Interaction.* Pearson Education Limited, 2nd edition, 2007.

[45] B. C. Stahl. On the Difference or Equality of Information, Misinformation, and Disinformation : A Critical Research Perspective. *Informing Science: The International Journal of an Emerging Transdiscipline*, 9:84–96, 2006.

[46] S. S. Stevens. On the Theory of Scales of Measurement. *Science*, 103(2684):677–680, 1946.

[47] E. R. Tufte. *Envisioning Information.* Graphics Pr, May 1990.

[48] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Pr, 2nd edition, May 2001.

[49] E. R. Tufte. *Beautiful Evidence*. Graphics Pr, first edition, July 2006.

[50] S. Ullman. *High-Level Vision: Object Recognition and Visual Cognition*. A Bradford Book, 1st edition, July 2000.

[51] A. Unwin. Requirements for Interactive Graphics Software for Exploratory Data Analysis. Technical report, Mathematics Institute, University of Augsburg, 2011.

[52] J. Wagemans, J. H. Elder, M. Kubovy, S. E. Palmer, M. a. Peterson, M. Singh, and R. von der Heydt. A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure-ground organization. *Psychological bulletin*, 138(6):1172–217, 2012.

[53] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann, 2nd edition, April 2004.