

# TAG<sup>2</sup>S<sup>2</sup>: A Tool for Automatic Generation of Good viSualizations using Scoring

Master thesis from

**Slobodan Kocevski**

1. PRÜFER

2. PRÜFER

JProf. Dr. Kai Lawonn Dr. Ute Masermann

---

September 23, 2019

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Thesis structure . . . . .	6
<b>2</b>	<b>Fundamentals</b>	<b>7</b>
2.1	What is data? . . . . .	7
2.1.1	Data types . . . . .	7
2.1.2	Data analysis . . . . .	10
2.2	What is Data Visualization? . . . . .	17
2.2.1	Visually encoding data . . . . .	20
2.2.2	Purpose for visualizing data . . . . .	22
2.3	What is good data visualization? . . . . .	25
2.3.1	Graphical perception . . . . .	26
2.3.2	Visualization clarity . . . . .	27
2.3.3	Data attributes . . . . .	29
2.4	Summary . . . . .	33
<b>3</b>	<b>Related work</b>	<b>35</b>
3.1	Tools and Research Approaches . . . . .	35
3.1.1	Commercial tools . . . . .	36
3.1.2	Research Approaches . . . . .	41
3.2	Analysis of Tools and Research Approaches . . . . .	45
3.2.1	Comparing Criteria . . . . .	46
3.2.2	Tool Comparison . . . . .	48
3.2.3	Strengths and Limitations . . . . .	50
3.3	Discussion . . . . .	53
3.4	Summary and Conclusion . . . . .	54

<b>4</b>	<b>Utility Metric for Generating Good Visualizations</b>	<b>56</b>
4.1	Visualization detection . . . . .	56
4.2	Visualization Ranking . . . . .	57
4.2.1	Defining Good Visualization . . . . .	57
4.2.2	Criteria for Ranking Visualizations . . . . .	58
4.2.3	Ranking Principle . . . . .	61
<b>5</b>	<b>Implementation</b>	<b>65</b>
5.1	Architecture and Technology . . . . .	69
5.2	Front-end . . . . .	70
5.3	Back-end . . . . .	73
<b>6</b>	<b>Results</b>	<b>83</b>
6.1	Synthetic Data-set . . . . .	83
6.2	Anonymized Real World Data-set . . . . .	96
6.3	Performance Measure . . . . .	109
6.4	Validating the Utility Metric . . . . .	110
<b>7</b>	<b>Conclusion and Future Work</b>	<b>116</b>
	<b>Appendix A Anonymized telecommunication data-set</b>	<b>121</b>
	<b>Appendix B reports2go Visualizations</b>	<b>122</b>
	<b>Appendix C TAG<sup>2</sup>S<sup>2</sup> visualizations</b>	<b>125</b>
	<b>Appendix D Chart properties object</b>	<b>127</b>
	<b>Appendix E Synthetic data-set</b>	<b>132</b>
	<b>Appendix F Data-set</b>	<b>133</b>
	<b>Appendix G Chart suggestions</b>	<b>134</b>
	<b>Appendix H date-time formats</b>	<b>135</b>
	<b>Bibliography</b>	<b>136</b>
	<b>Eidesstattliche Erklärung</b>	<b>140</b>

# Abstract

Data visualization is an effective way to explore data. It helps people to get a valuable insight of the data by placing it in a visual context. However, choosing a good chart without prior knowledge in the area is not a trivial job. Users have to manually explore all possible visualizations and decide upon ones that reflect relevant and desired trend in the data, are insightful and easy to decode, have a clear focus and appealing appearance. To address these challenges we developed a **T**ool for **A**utomatic **G**eneration of **G**ood **v**isualizations using **S**coring (TAG<sup>2</sup>S<sup>2</sup>). The approach tackles the problem of identifying an appropriate metric for judging visualizations as good or bad. It consists of two modules: *visualization detection*: given a data-set it creates a list of combination of data attributes for scoring and *visualization ranking*: scores each chart and decides which ones are good or bad. For the later, an utility metric of ten criteria was developed and each visualization detected in the first module, is evaluated on these criteria. Only those visualizations that received enough scores are then presented to the user. Additionally to these data parameters, the tool considers user perception regarding the choice of visual encoding when selecting a visualization. To evaluate the utility of the metric and the importance of each criteria, test cases were developed, executed and the results presented.

# Acknowledgments

First and foremost, I thank to my thesis advisor JProf. Dr. Kai Lawonn for the support and the opportunity to write a thesis on a very interesting topic for me.

Many thanks to Dr. Ute Masermann and Alexandr Yessipovskiy. I highly appreciate the valuable feedback and all guidance which helped me to keep moving and improve my work. My sincere thanks also goes to DECADIS AG for the permission to use company's infrastructure throughout the period of writing the thesis.

Finally, I must express my gratitude to my parents for providing me with support and continuous encouragement throughout my years of study and through the process of writing this thesis. This accomplishment would not have been possible without them. Thank you.

# List of Figures

1.1	Good line chart presenting company's revenue over a year . . . . .	2
1.2	Cluttered line chart presenting company's cost over a year . . . . .	3
1.3	A good multi-graph series showing company's costs over a year . . . . .	3
1.4	Example of bad visualization showing part-whole relation with a pie chart over a time dimension . . . . .	4
1.5	Example of good visualization showing part-whole relation over a time interval . . . . .	5
2.1	Example of correlation coefficient in a plot . . . . .	12
2.2	Visual presentation of not binned numerical data . . . . .	16
2.3	Visual presentation of binned numerical data . . . . .	17
2.4	Visual presentation of patterns from a productivity data table . . . . .	19
2.5	Examples of using preattentive properties . . . . .	20
2.6	Hue wheel presented in a circular form . . . . .	21
2.7	Examples of saturation of hue red . . . . .	21
2.8	Examples of lightness degree of hue red . . . . .	21
2.9	Example of exploratory explanation data visualization . . . . .	23
2.10	Example of explanatory data visualization . . . . .	24
2.11	Example of hybrid data visualization . . . . .	25
2.12	Hierarchy of chart types ordered from most to least accurate by Cleveland and McGill . . . . .	26
2.13	Example of scatter plot having all visual elements on same level . . . . .	28
2.14	Adjusted scatter-plot with focus on the fitted line . . . . .	28
2.15	Good line chart showing trend of <i>Films</i> and <i>Games</i> over a year . . . . .	29
2.16	Grouped bar displaying trend of <i>Films</i> and <i>Games</i> over a year . . . . .	30
2.17	Example of good side-by-side bar chart . . . . .	30
2.18	Example of bad side-by-side bar chart . . . . .	31
2.19	Example of a good multi-graph . . . . .	31

*List of Figures*

2.20	Showing correlation between numerical values . . . . .	32
2.21	Good and bad example of charting part-whole relation with pie and stacked bar charts . . . . .	33
3.1	Gartner’s Magic Quadrant 2019 for Analytics and Business Intelligence Platforms . . . . .	36
3.2	Elicit construction [KW18] . . . . .	41
3.3	Visualization assessment [KW18] . . . . .	41
3.4	Analysis of visualization clusters [KW18] . . . . .	42
3.5	Overview of DeepEye [Luo+18] . . . . .	43
3.6	SeeDB Front-end [Var+15] . . . . .	45
3.7	SeeDB architecture [Var+15] . . . . .	45
5.1	reports2go upload data-set view . . . . .	65
5.2	Uploading data-set in reports2go . . . . .	66
5.3	A configuration page of reports2go . . . . .	67
5.4	A configuration of data-attributes of reports2go . . . . .	68
5.5	Architecture of TAG <sup>2</sup> S <sup>2</sup> . . . . .	69
5.6	Example of line chart for calculating intersection points . . . . .	78
6.1	<i>Units sold per Regions</i> . . . . .	86
6.2	Example of a good pie chart showing <i>Total costs per Sales channel</i> . . . . .	86
6.3	Good slope chart showing <i>Total costs of Order priorities over two Sales channels</i> . . . . .	87
6.4	Example of a good grouped-bar chart displaying <i>Units sold</i> on different channels with different priority . . . . .	88
6.5	Scatter-plot showing <i>Total revenue</i> and <i>Total cost</i> . . . . .	88
6.6	Example of bubble chart presenting <i>Total revenue</i> and <i>Total cost</i> with <i>Units sold</i> as size of dots . . . . .	89
6.7	Examples of bad lines charts due to big clutter or data overload . . . . .	91
6.8	Multi-graph showing <i>Units sold by Item type</i> . . . . .	92
6.9	Examples of bad charts showing part-whole relation for <i>Country sales</i> data-set . . . . .	93
6.10	Overloaded slope chart containing few slopes and too many data points . . . . .	94
6.11	Examples of bad stacked-bar charts . . . . .	95
6.12	Too many groups of temporal data presented with grouped-bar chart . . . . .	96

*List of Figures*

6.13	Two numerical attributes having no correlation displayed with scatter-plot	96
6.14	Scores for each criteria summed for all 1452 visualizations . . . . .	97
6.15	The change of <i>KPI1</i> for <i>Size</i> over <i>Date</i> . . . . .	100
6.16	The change of <i>KPI2</i> for <i>Products</i> over <i>Date</i> . . . . .	100
6.17	Values of <i>KPI1</i> for each <i>Size</i> . . . . .	101
6.18	<i>KPI1</i> values for each <i>Size</i> . . . . .	101
6.19	Examples of good grouped-bar charts . . . . .	102
6.20	Slope charts . . . . .	103
6.21	The correlation between <i>KPI2</i> and <i>KPI3</i> . . . . .	104
6.22	Bad line charts . . . . .	106
6.23	Change of <i>KPI2</i> for <i>Area</i> given by <i>Size</i> . . . . .	107
6.24	Examples of bad charts when too many categories need to be visualized .	107
6.25	Size of <i>KPI1</i> for each <i>Area</i> per <i>calendarWeek</i> . . . . .	108
6.26	Number of <i>KPI1</i> per <i>calendarWeek</i> goruped by <i>Area</i> . . . . .	108
6.27	Good visualization automatically generated by TAG <sup>2</sup> S <sup>2</sup> . . . . .	112
6.28	Visualizations generated after modifying the utility metric . . . . .	113
6.29	Good visualization automatically generated by TAG <sup>2</sup> S <sup>2</sup> . . . . .	114
6.30	Visualizations generated after re-modifying the utility metric . . . . .	115
B.1	Size of <i>KPI1</i> for each <i>Area</i> distributed by <i>Date</i> . . . . .	122
B.2	<i>KPI1</i> values for each <i>Customer Group</i> per <i>Date</i> . . . . .	123
B.3	<i>calendarWeek</i> per <i>Date</i> given for the size of <i>KPI1</i> . . . . .	123
B.4	<i>KPI1</i> values for <i>Product</i> given per <i>Date</i> . . . . .	124
B.5	<i>KPI1 Size</i> per <i>Date</i> . . . . .	124
C.1	<i>KPI1</i> value by <i>Area</i> over a <i>Date</i> . . . . .	125
C.2	<i>KPI1 Product</i> size per <i>Date</i> . . . . .	126
C.3	<i>Size</i> categories shown with measure <i>KPI1</i> per <i>Date</i> . . . . .	126
G.1	Chart selector developed by Dr. Abel [KNA13] . . . . .	134



# List of Tables

1.1	A snippet of a data form a publishing company . . . . .	2
2.1	Example of quantitative data . . . . .	8
2.2	Example of nominal category . . . . .	9
2.3	Example of ordinal category . . . . .	9
2.4	Example of interval categorical data . . . . .	10
2.5	Example of time series data . . . . .	10
2.6	Overview of test scores with hours of preparation . . . . .	12
2.7	Example of calculating average(mean) salary . . . . .	13
2.8	Example of using mean when it provides misleading information . . . . .	14
2.9	Students scores from reading and writing examination . . . . .	15
2.10	Data about people’s age from a small group . . . . .	16
2.11	A binned data about people’s ages . . . . .	17
2.12	Company costs and revenue for two quarters . . . . .	18
3.1	SAS’ criteria for chart type selection . . . . .	38
3.2	Rules for creating automated visualization in Tableau . . . . .	40
3.3	Criteria evaluating the visual representation . . . . .	46
3.4	Data-related evaluation criteria . . . . .	48
3.5	Color coding scheme used for performing comparative analyse . . . . .	48
3.6	Evaluation criteria observed for tools’ analyse . . . . .	49
3.7	Comparative analyse of tools for automated visualizations . . . . .	53
4.1	Weights of each criterion per visualization type . . . . .	64
5.1	Overview of a calculating intersection for two lines . . . . .	79
6.1	Good charts with chart type and their score . . . . .	85
6.2	Bad charts for the <i>Country sales</i> data-set and their issues . . . . .	90

*List of Tables*

6.3	Description of the good charts and their score for Customer data-set . . .	99
6.4	Bad charts with chart type and their score . . . . .	105
6.5	Run-time comparison matrix for each process of TAG <sup>2</sup> S <sup>2</sup> . . . . .	110
6.6	Altered weights of the utility metric used for validation . . . . .	111
6.7	Altered weights of the utility metric used for second validation . . . . .	114
7.1	Evaluation criteria observed for TAG <sup>2</sup> S <sup>2</sup> . . . . .	118
A.1	Subset of the Customer data-set . . . . .	121
E.1	Subset of the Country sales data-set . . . . .	132
F.1	Overview of number of males and females in years . . . . .	133

# 1 Introduction

## 1.1 Motivation

Data visualization is crucial in today's data-focused world as it is a common and effective way to explore unknown data. Using graphical objects to present and comprehend large amounts of data is easier because we tend to process visual information far more easily than written information. With the help of data visualizations now business leaders are able to respond faster to the market changes and identify new opportunities, behaviors of various kind.

Tools exist that give the possibility to create visualizations in a few seconds. The main idea behind those tools is to allow the user to select data and pick a chart type to create the visualization. However, to identify a good visualization, users must have a good knowledge about the data-set, and the current visualization tools provide good charts only to those users who know the data well.

Ideally, users need a tool that automatically will recommend visualizations, so they can pick the right one. This is not easy to accomplish as there exist no agreed on definition of what good visualization mean [Luo+18]. Yet recommending data visualizations that provide new and valuable insights is a challenging problem. These recommending systems automatically recommend top-k charts that are interesting, where interestingness is quantified by an utility function [ESC18]. These systems work well with a specific data attribute such as measures or dimensions (e.g., [Var+15]) or focus on providing single type of good chart (e.g., [Luo+18]).

As there exists no common definition on good visualization, many researchers define charts as good based on different points of view. Good visualizations provide insight, are clear and aesthetically pleasing [Cai16]. They are easy-to-interpret and include only the necessary elements (like legends, axis values, and grid) for a single user to

## 1 Introduction

have an easy-to-understand look of the data. Those visualizations that do not follow any distribution and are hard to interpret are bad visualizations [Luo+18]. For example, line charts could be complex and difficult to interpret when many values and big changes in values have to be presented, thus making the chart cluttered and chaotic. While pie charts are good when presenting part-to-whole share, they are mostly misunderstood and difficult to interpret, as people are forced to compare angles which is hard [Eve17]. To illustrate the difference between good and bad visuals, let us consider the following examples.

**Example 1.** Table 1.1 contains data regarding publishing and selling printed materials. The visualizations presented in Figures 1.1 - 1.3 consider the entire data from this table.

Product	Month	Units sold	Revenue	Cost
Books	01.2018	80	13000	4680
Newspaper	01.2018	913	9800	360
Journals	01.2018	60	3600	1830
...	...	...	...	...

Table 1.1: A snippet of a data form a publishing company

The chart in Figure 1.1 is an example of a good chart showing the time dimension *Month* and the measure *Revenue*. It gives an insight of the total revenue gained per month by a single product. The lines for the products are easy to follow and compare the revenue gained from each product over the year.

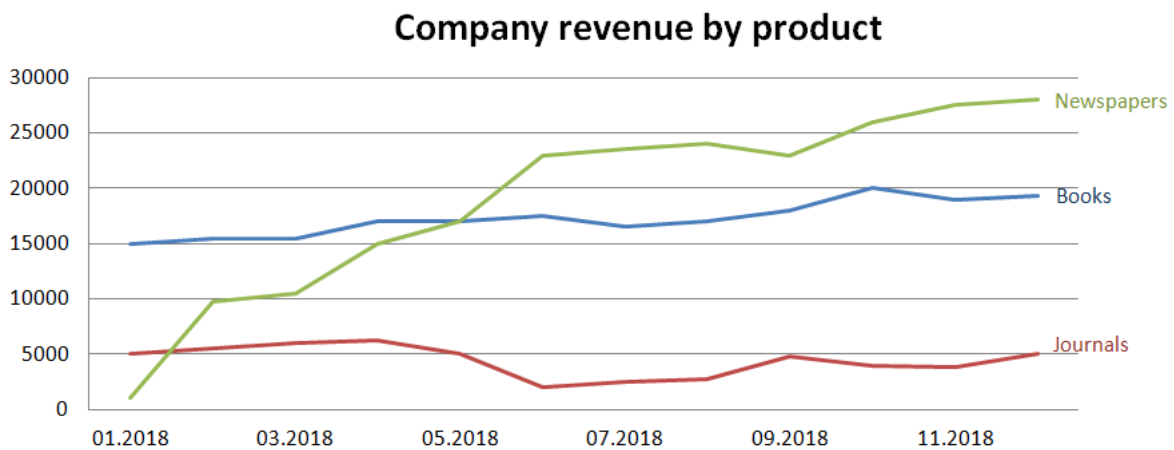


Figure 1.1: Good line chart presenting company's revenue over a year

## 1 Introduction

We can easily conclude that *Journals* brought least *Revenue* to the company. In contrast, throughout the whole year *Newspapers* have increased sale. These conclusions would have not been easy to make by looking at the row data.

Figure 1.2 displays the same time dimension *Months* and the measure *Costs*. Here, it is very difficult to make any conclusion as this chart does not follow any distribution and cannot tell anything. As the values for the measure fluctuate from one point to another, the resulting visual is cluttered and hard to interpret. The multi-graph displayed in Figure 1.3 presents a good alternative focusing the user on the data by dimming the grid lines, eliminating the clutter by splitting each category in a separate chart allowing comparison.

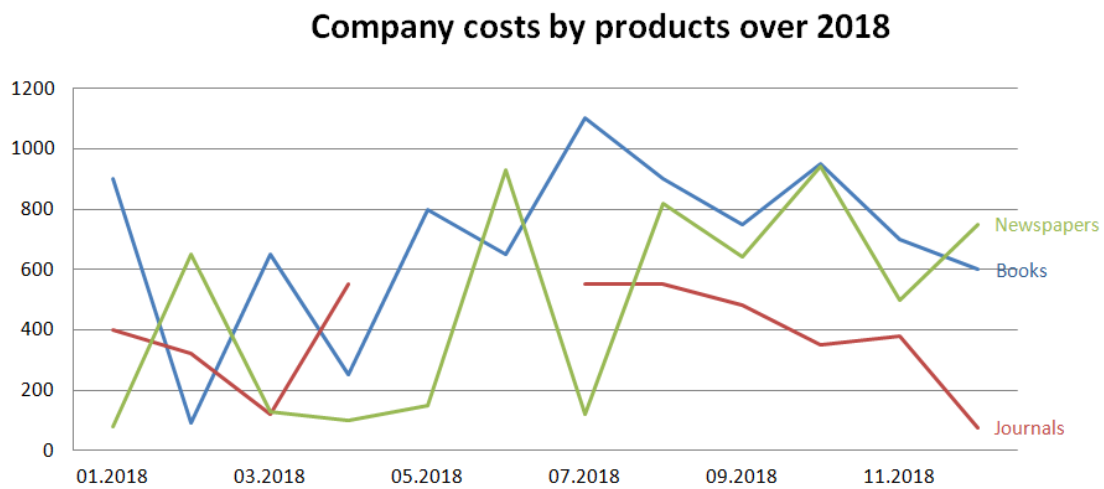


Figure 1.2: Cluttered line chart presenting company's cost over a year

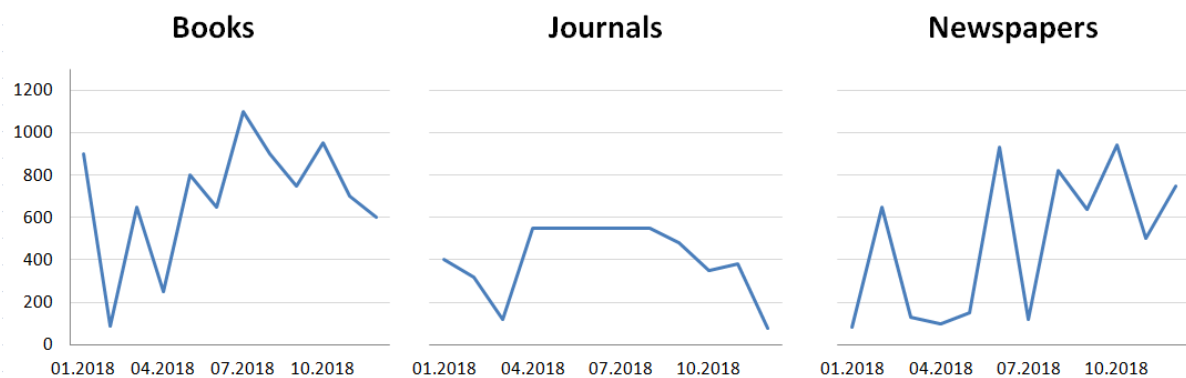


Figure 1.3: A good multi-graph series showing company's costs over a year

Now, we are able see the trend of each line. The costs for printing *Books* and *Newspapers* have been increased in the second half of the year, while those for *Journals* have a steep decline during the last period.

**Example 2.** Let us consider the pie charts in Figure 1.4. The percentage on the pie chart on the right are left on purpose. The goal of the visuals is to provide change in movie genre popularity over a decade. What is obvious to conclude is that the preference for comedy movies has been increased, but what about horror movies? Also, western and documentary movies were equally popular in 2004, is the same case in 2014?

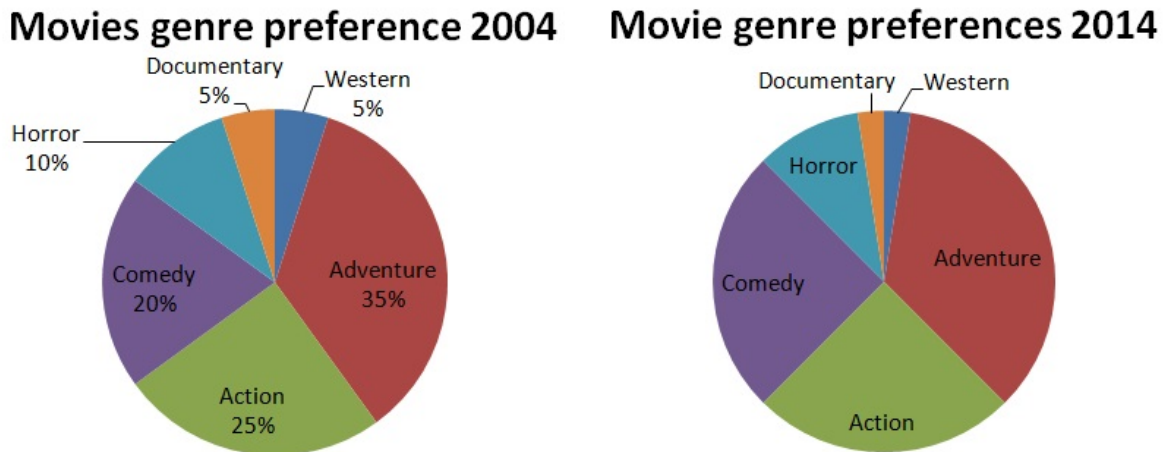


Figure 1.4: Example of bad visualization showing part-whole relation with a pie chart over a time dimension

Comparing slices within a pie chart is difficult, but comparing two or more pie charts is more difficult. With pie charts, people are forced to focus on the number in the graph, rather than to focus on the graphical object presented and its size. When they have to read the numbers to get an insight, the utility of this chart becomes questionable [Cai16]. Another problem with these pie charts is the number of dimensions. Good pie charts work best when there are maximum four categories and the differences among them are distinct [Eve17]. A good alternative to the data shown in the pie chart would be an area chart as given in Figure 1.5.

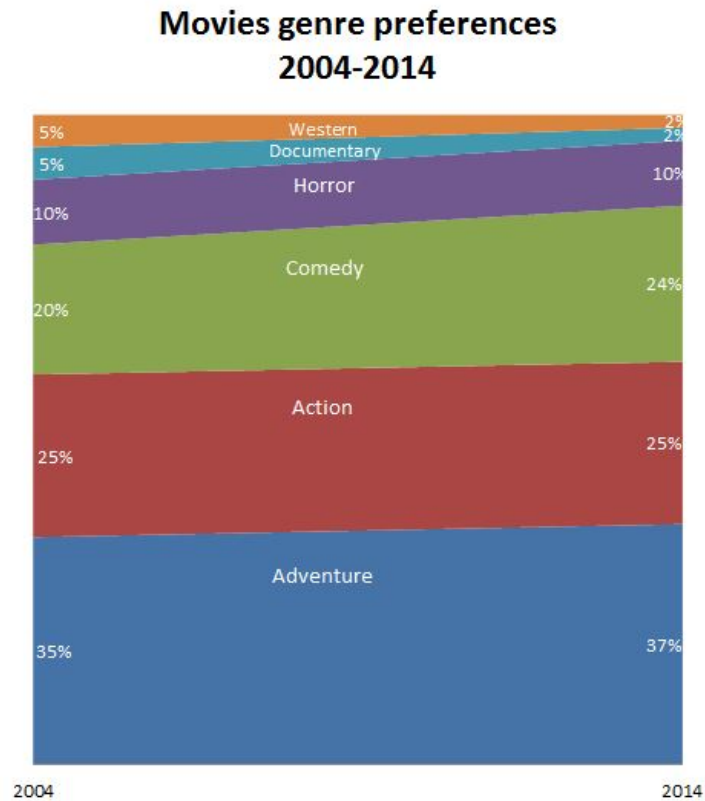


Figure 1.5: Example of good visualization showing part-whole relation over a time interval

Area charts are an alternative that show trend over time and indicate that the pieces make up an entire unit [Eve17]. On the one hand they imply that the data has part-whole relation, and on the other hand they follow human perception rules for making the chart easy-to-interpret and compare.

The main goal of this thesis is to elaborate what a good visualization is and how to create one automatically out of a given data-set. For this purpose, the following research objectives have been defined:

- What is a good visualization?
- How to generate a good visualization automatically?
- Implementation of a tool that generates such visualizations.

## 1.2 Thesis structure

To answer the research questions, we take the following approach:

As a first step, we define what we mean by *data* and *visualization*. We provide as well definitions of good visualization from three point of views: data attributes, visualization clarity and human perception. The findings are described in Chapter 2.

The next step is to study the tools that automatically provide visualizations. The motivation here is to identify gaps and topics for improvement by studying their features. Chapter 3 presents commercial tools for automated visualization and research approaches for defining good or interesting visualization. An evaluation matrix is provided to give an overview of the features these tools and approaches have.

In the following step we provide our definition for good visualization, based on the knowledge acquired in the previous steps. For this purpose a set of ten data related criteria is defined that comprise a utility metric responsible to differentiate between good and bad visualization. For each visualization type a list of ten weighted criteria is defined. The weights vary depending on the importance of the criteria for the specified chart type. The sum of these weights gives a score which is used later for ranking. The utility metric together with the weighted criteria is described in Chapter 4.

Afterwards, the implementation of TAG<sup>2</sup>S<sup>2</sup> (a **T**ool for **A**utomatic **G**eneration of **G**ood **v**iSualizations using **S**coring) is described. It is a tool that automatically generates good visualizations using scoring. The utility metric defined in the previous step is employed within the algorithm. Chapter 5 gives the technical background for developing the tool.

Results from the implementation of TAG<sup>2</sup>S<sup>2</sup> are given in Chapter 6. Examples of good and bad visualization are provided as a result from the employed utility metric. This Chapter as well gives the results from evaluation of the metric. For this step, the weights for each criteria were manipulated to compare the results given with not manipulated weight. The goal is to assess whether a change in weight for a single criteria will result in producing similar (good or bad) visualizations.

Concluding thoughts on the topic and the implementation are given in the Chapter 7.

The core Chapters in this thesis are Chapter 4 and Chapter 5 as we provide our definition of good visualization and describe the implementation of TAG<sup>2</sup>S<sup>2</sup>.



## 2 Fundamentals

In order to comprehend the approach taken to tackle the problem of defining good visualizations, we provide background knowledge and definitions of key topics, methods and concepts.

### 2.1 What is data?

Data comes in different formats and sizes. Every second, big amounts of data is produced by cell phones, fitness trackers or web applications in form of numbers, pictures, audio or text. The easy availability of data contributes to developing various tools for data storage and data analysis. When properly handled, data provides accurate and meaningful information allowing us to see patterns and connections that matter.

However, data as it is generated sometimes is not useful for end-users. They are interested into acquiring knowledge, finding patterns, outliers, doing analyses, therefore the need for tools that can support such analysis is increasing proportionally. In the following sub-chapter different data types are elaborated, while the second describes statistical measures which can be used on a specific data types. Later these measures are used for developing data-based utility measure for scoring visualizations.

#### 2.1.1 Data types

Before starting to describe the visualization process, it is necessary first to talk about data, different types and statistical measures that can be applied on such data types. As there exist many data types, for the purpose of this thesis we define only those commonly used for visualizations [Tom19]. These data types can be classified into three groups: numerical (or quantitative), categorical and time series [Few05].

## Quantitative

**Definition 1.** *Quantitative data is defined as the value of data in the form of counts or numbers where each data-set has an unique numerical value associated with it [Bha19b].*

This data is any quantifiable information, describes how much there is of something that can be used for mathematical calculations and statistical analysis. It can be verified and can also be conveniently evaluated using mathematical techniques. The range of one measure is different and can vary. Examples of quantitative data are given in the columns *Age* and *Grade* from Table 2.1.

Student ID	Age	Subject	Grade
S23956	20	English	3.0
S56254	18	Math	2.7
S63568	21	History	2.0
S23565	19	Sport	1.3
S23556	21	Geography	1.0

Table 2.1: Example of quantitative data

## Categorical

Categorical data used in tables or charts to name certain measure can come in one of the following three types: nominal, ordinal, and interval [Few05]:

- Nominal category.

**Definition 2.** *A nominal category or a nominal group is a group of objects or ideas that can be collectively grouped on the basis of a particular characteristic - a qualitative property [RP06].*

It consists of discrete values in a single category that do not relate to one another in any particular way. The items have no particular order and do not represent quantitative values. The column *Department* from Table 2.2 gives an example of nominal category.

ID	Department	Hours
1003	Sales	5

1004	Finance	6
1005	Marketing	8
1006	Administration	10
1007	Shipping	15

Table 2.2: Example of nominal category

- Ordinal category

**Definition 3.** *Ordinal data is a categorical, statistical data type where the variables have natural, ordered categories and the distances between the categories is not known [Agr13].*

It has a prescribed order, but like the nominal category, the items do not represent quantitative values. This kind of data is often found in questionnaires. A well-known example is the Likert scale given in Table 2.3 [CSP96].

Order	Scale
1	Strongly agree
2	Agree
3	Neutral
4	Disagree
5	Strongly disagree

Table 2.3: Example of ordinal category

- Interval category

**Definition 4.** *Interval data is defined as a data type which is measured along a scale, in which each point is placed at equal distance from one another. Interval data always appears in the form of numbers or numerical values where the distance between the two points is standardized and equal [Bha19a].*

It consists of items that have a prescribed order, but here they represent quantitative values. Usually the items from this category subdivide a larger range of quantitative values into smaller ranges. These ranges (intervals) have a specific order from smallest to largest. The column *Percentage* in Table 2.4 gives an example of interval categorical data.

Percentage	Grade points	Interpretation
0-59	0.0	Failure
60-69	1.0	Lowest acceptable
70-79	2.0	Average
80-89	3.0	Above average
90-100	4.0	Outstanding

Table 2.4: Example of interval categorical data

## Time series

**Definition 5.** *Time series data are a collection of ordered observations recorded at a specific time, for instance, hours, months, or years [Str19].*

It presents a sequence of data points. Time is a continuous variable and can be divided into intervals of varying duration [Few05]. The time series data given in Table 2.5 is stored in monthly intervals.

Month	Passengers
01.2019	150
02.2019	180
03.2019	185
03.2019	200

Table 2.5: Example of time series data

### 2.1.2 Data analysis

Data analysis is the process of systematically applying statistical and/or logical techniques to describe and illustrate, condense and recap, and evaluate data. As one of the thesis goals is to develop a data-based metric that automatically generates good visualizations, in this sub-chapter we present some of the statistical measure we use in our metric (e.g., for detecting relationship between two numerical data attributes or understand how spread out from the average the numbers in one data-set are), and provide

background information for measures that related tools use.

**Correlation**

This measure compares two sets of quantitative values in order to detect if increase in one value results in either increase or decrease in another value.

**Definition 6.** *The correlation coefficient is a statistical measure that calculates the strength of the relationship between the relative movements of two variables [GAN19].*

The correlation coefficient is calculated by using the Formula 2.1. First we determine the covariance of the two variables in question. Next, we calculate each variable’s standard deviation. The correlation coefficient is determined by dividing the covariance with the product of the two variables’ standard deviations.

$$r = \frac{n(\sum XY) - (\sum X)(\sum Y)}{\sqrt{(n \sum X^2 - (\sum X)^2)(n \sum Y^2 - (\sum Y)^2)}} \tag{2.1}$$

The result is a single value between -1 and 1 called linear correlation coefficient. This value indicates if there is a correlation between the values, and if so, is it positive or negative. When the correlation coefficient has a value of 0, it means that there exists no correlation. In contrast, values of -1 or 1 gives negative or positive correlation respectively. The greater the value, either positive or negative, the stronger the linear correlation. This measure is also used to predict values (e.g., sales revenue) by knowing or controlling other values (e.g., marketing emails) [Few12].

One example of positive correlation is given in Table 2.6. It shows data about time spent to prepare for a test and the test scores.

Hours of preparation	Test scores
8	81
6	80
6	75
5	65
7	91
6	80
3	40

## 2 Fundamentals

4	44
3	8
3	32
8	85
6	75
6	77
5	62
7	78
6	70
3	39
4	48
3	35
3	36
Correlation r	0.9544

Table 2.6: Overview of test scores with hours of preparation

The coefficient 0.9544 means that there is a strong correlation between the two variables. When the number of hours to prepare for the test increases so does the test score. This can as well be seen from Figure 2.1. The trendline is given to make the correlation clear in the scatter-plot.

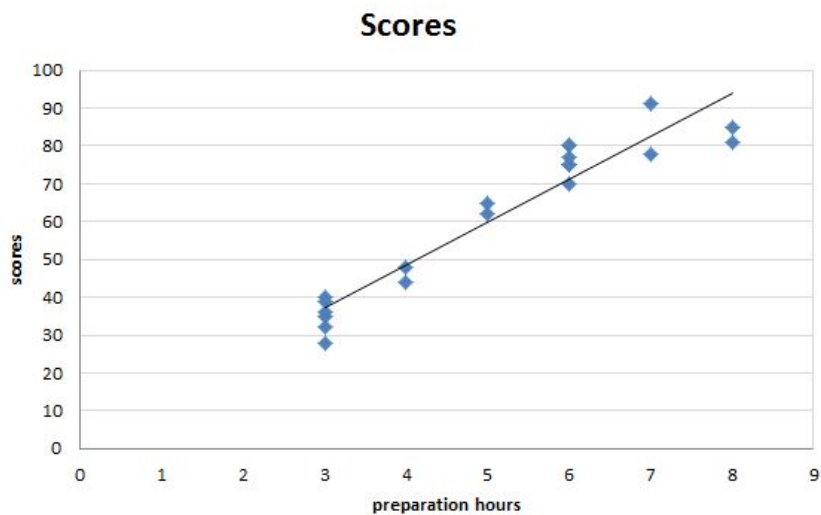


Figure 2.1: Example of correlation coefficient in a plot

**Mean**

**Definition 7.** *In mathematics and statistics, the mean or average is the sum of a collection of numbers divided by the count of numbers in the collection [Jac94].*

For a data-set consisting of the values  $\{a_1, a_2, \dots, a_n\}$ , then the arithmetic mean  $A$  is defined by the following formula:

$$A = \frac{1}{n} \sum_{i=1}^n a_i \quad (2.2)$$

Its value represents the center of an entire set. Let us consider the data in Table 2.7

Employee ID	Salary
EMP165	2500
EMP175	2700
EMP185	2400
EMP195	2300
EMP205	2550
EMP215	2650
EMP225	2750
EMP235	2450
EMP245	2600
EMP255	2400
Average salary	2530

Table 2.7: Example of calculating average(mean) salary

By calculating the mean we can understand that the average salary in the company is 2530. However, when the data consists of few outliers (extreme high or low values), then the mean could be a misleading measure to use. Let us consider the data values from column *Income* in Table 2.8.

Employee	Department	Income
A	D1	1200
B	D1	1100
C	D2	5500

## 2 Fundamentals

D	D2	6300
E	D3	900
F	D3	750
Average salary		2625

Table 2.8: Example of using mean when it provides misleading information

In this data-set, the mean is much higher than most of the salaries, thus giving the impression that employees are suited better than they really are. Alternatively, calculating a median gives better results. First, we arrange all elements from smallest to greatest, then take the middle value. In a case as ours when we have even numbers of values, then we calculate the mean of the two values in the middle and give us 1150. Now it is clear that most of the values in the table are closer to the median than the mean, thus giving more realistic insight.

### Standard Deviation

**Definition 8.** *In statistics, the standard deviation is a measure that is used to quantify the amount of variation or dispersion of a set of data values [BA96].*

It represents a single number providing how spread out the numbers are. If the data points are far from the mean (spread out), there is a higher deviation within the data-set. The formula for standard deviation is:

$$s = \sqrt{\sum_{i=1}^N (x_i - \bar{x})^2} \quad (2.3)$$

To illustrate this measure, let us consider the data in Table 2.9.

Student	Reading score	Writing score
A	176	166
B	125	163
C	155	164
D	115	155



E	195	175
F	180	140
G	203	155
H	105	168
I	145	166
J	155	163
ST	33,44	9,55

Table 2.9: Students scores from reading and writing examination

It shows reading and writing score for individual student. Standard deviation is calculated for both measures. For the reading score, standard deviation is 33.44, whereas for writing scores it is 9.55. Comparing these two values can be concluded that the scores for the reading test vary much more than for the writing test.

### Binned Aggregation

**Definition 9.** *Binned aggregation is a process of grouping numerical values along a dimension into adjacent intervals over the range of values covered by that dimension [ESC18].*

It is a pre-processing technique for data smoothing. For example, in a data-set about a group of people, we can arrange their ages into a smaller number of age intervals (e.g., grouping every five years). Table 2.10 contains information regarding the age of group of people. The visualization for this raw data is given in Figure 2.2.

Age	Number of people
13	1
15	2
20	1
22	4
23	2
25	1
29	3

## 2 Fundamentals

30	3
32	1
33	2
34	1
35	2
39	1
40	1
44	2
50	1

Table 2.10: Data about people's age from a small group

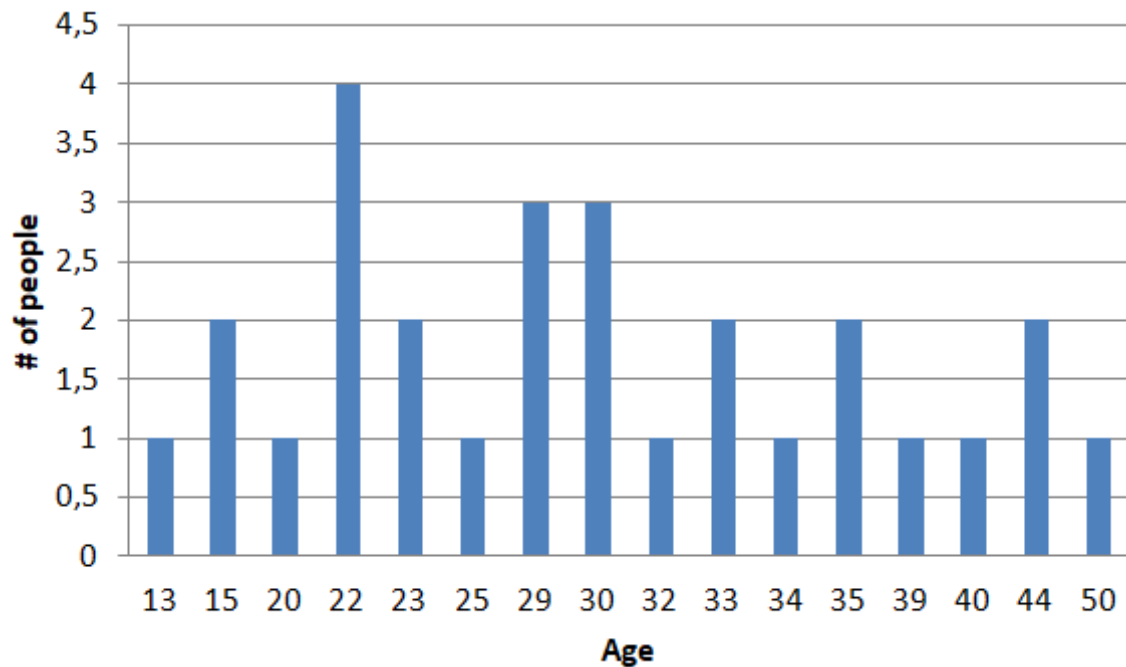


Figure 2.2: Visual presentation of not binned numerical data

By binning the age of the people in bins each representing 10 years the data can be summed for different age groups instead of for each age. Table 2.11 gives the binned aggregated data for people ages. Figure 2.3 visualizes this data as a bar chart.

Age groups	Number of people
$x \leq 20$	4

$20 < x \leq 30$	13
$30 < x \leq 40$	8
$40 < x \leq 50$	3

Table 2.11: A binned data about people's ages

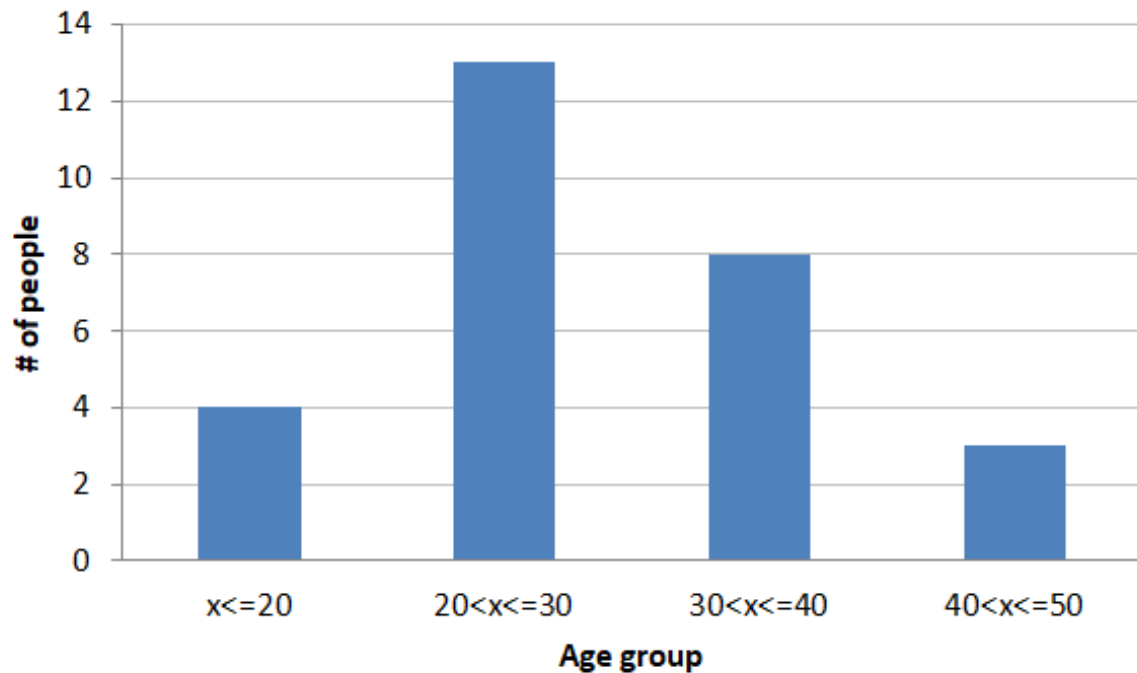


Figure 2.3: Visual presentation of binned numerical data

In this way we are able to group a number of more or less continuous values into a smaller number of bins.

## 2.2 What is Data Visualization?

Visualizations have a potentially enormous influence on how data are used to make decisions across all areas of human endeavor [Cor19]. Nowadays, it is hard to find an area that does not use tools to make the data more understandable, varying from the public sector, to social science. Whether to predict sales volumes or see trends in society, the need and use cases of data visualization is prolific.

## 2 Fundamentals

Data visualization is a common way of graphical representation of information. With the use of graphical elements like bars, circles or slices, it provides an accessible way to examine massive high dimensional data-sets.

Most commonly used visualization types are [Rue19]:

- Line
- Area chart
- Scatter-plot
- Bubble chart
- Bar chart
- Stacked-bar chart
- Pie chart

As there exist big number of visualization types, Knaflic has made a classification in the following four categories [Kna15]:

1. Points
2. Lines
3. Bars
4. Area

People perceive more accurately graphs, rather than long columns of numbers in a table. In a data-set with thousands of records regarding the productivity of employees and the use of two competing software, one could immediately see the nature of the relationship [Few12]. However, data does not always have to be large and complex, for a visualization to be useful. Lets take a look at Table 2.12.

	Department 1		Department 2	
	Quarter 1	Quarter 2	Quarter 1	Quarter 2
Revenue	533	683	480	614
Costs	620	810	750	550

Table 2.12: Company costs and revenue for two quarters

## 2 Fundamentals

By looking only at the numbers it is quite difficult to assess the productivity of the two departments for the first half of the year. Is there any unusual pattern that occurs in one of the departments? Do both departments make profit for the company?

Now, let's take a look at the line chart shown in Picture 2.4.

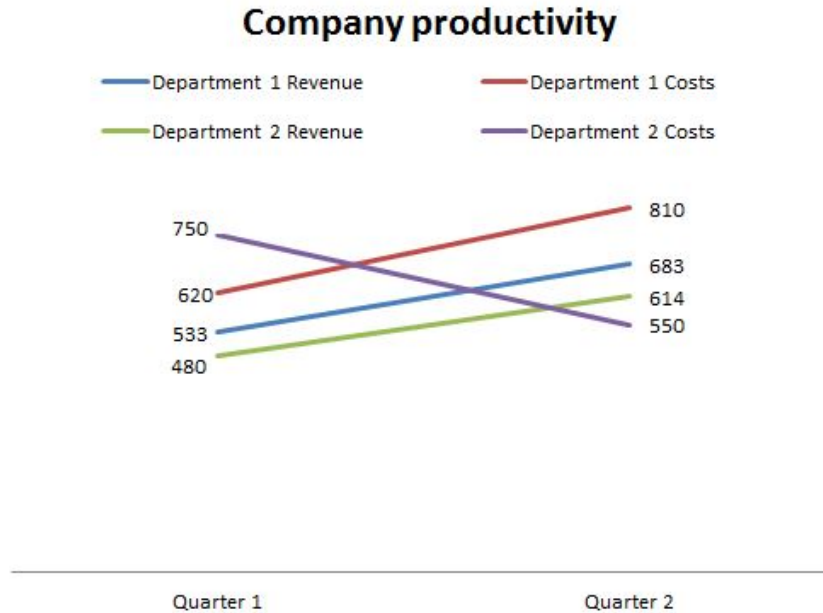


Figure 2.4: Visual presentation of patterns from a productivity data table

Here, clearly we can conclude that despite the reduced costs in the second half of the year, Department 2 still increased the revenue. The result is obvious after seconds of looking at the graph. What was easier to spot in the graph, required more time to understand by looking at the numbers.

Visually presenting information in a graph makes patterns to be obvious, provides valuable knowledge including changes, differences among data attributes which would be hard to spot when looking at the raw data.

Throughout the thesis, we use the words *chart* and *graph* interchangeably. Typically, chart is the broader category, with graphs being one of the subtypes (other chart types include maps and diagrams) [Kna15]. We do not point to any distinction and use both terms.

### 2.2.1 Visually encoding data

One of the great strengths of data visualization is people’s ability to process visual information faster than verbal information. Preattentive visual processing occurs in the brain for a very short time prior to conscious awareness. Some basic attributes, such as differences in length, size, hue, color intensity, angle, texture and shape, could be used as building blocks of data visualization [Few16]. Ware defines four types of preattentive attributes: form, color, spatial position and motion [War13]. Humans can identify differences among graphical objects when any of the preattentive attributes are present. Figure 2.5 illustrates this with three examples. In the first example, length is taken as a form attribute to make a certain object stand out from the rest, in the second example color is used to distinguish between objects, whereas the last example uses position as preattentive attribute to show difference. With the help of these attributes, we can design graphs that visually focus the user on the important information. Each of these four attributes can be quantified and expressed as a value.



Figure 2.5: Examples of using preattentive properties

#### Form

Form applies to one of the following attributes: length, width, orientation, shape, size, enclosure. The first example in Figure 2.5 illustrates variation of line length. When these lines present values from a certain dimension, it is important that all lines share a common baseline for easy comparison [Few12].

#### Color

This attribute is considered as the most common property to call attention. The reason being so is that intensities and hues are subjected to preattentive processing. The primary system used to describe color is known as HSL (Hue, Saturation and Lightness)

## 2 Fundamentals

scale.

Hue is another term for color (red, green, yellow). Numerically can be presented as percentage (0-360%). Figure 2.6 shows a color wheel in a circular form with percentage for a certain hue.

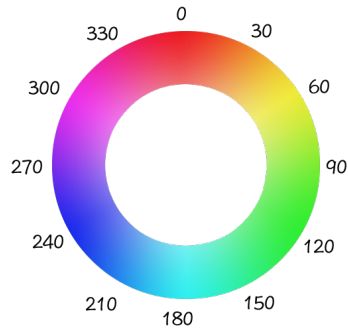


Figure 2.6: Hue wheel presented in a circular form

Saturation measures the degree to which a hue exhibits its essence. It is expressed as a percentage starting from 0 to 100. Figure 2.7 provides a saturation scale for the hue red, with 0% on the left and 100% on the right.



Figure 2.7: Examples of saturation of hue red

Lightness measures in percentage the degree to which a color appears dark or bright. Figure 2.8 shows the scale of lightness for the hue red. Any color can be described numerically using the three measures: hue: 0-360, saturation 0-100 and lightness 0-100.



Figure 2.8: Examples of lightness degree of hue red

### **Spatial position**

Spatial position is the ability to perceive two or more objects position in two dimensions: vertical and horizontal positions. The third dimension - depth is not perceived as well as the first two [Few12].

### **Motion**

Movement has the two sub-attributes: flicker and motion. However, they can cause a distraction from the rest of the information that is being presented. Mostly employed technique for web advertising, and not for visualizing data-sets.

### **2.2.2 Purpose for visualizing data**

The idea of using graphical objects to understand data has been known for a long time, from the period when people were using maps and graphs in the 17th century to the invention of the first chart types. It kept evolving over time, and with the rapid growth of technologies, data visualization increased its popularity and usability.

Due to the brain capabilities, humans are able to faster understand large amount of complex data using charts, that might be impossible to see in a text-based data. Furthermore, the big computation power has also contributed in increasing the popularity of the visualization. Computers are able to process and visualize large amount of data at fast speed.

Data visualization is also powerful when it comes to remembering information. At a very young age, people start storing visual memories and can remember and recall some of them throughout the entire life. This is because our brain can commit visuals to long-term memory a lot easier than text. According to Dr. Lynell Burmark, Ph.D. Associate at the Thornburg Center for Professional Development [Par16]:

”Words are processed by our short-term memory where we can only retain about 7 bits of information (plus or minus 2). Images, on the other hand, go directly into long-term memory where they are indelibly etched.”

Whether to solve a certain problem or to get a new insight, the application of data visualization is broad, today it is hard to think of a professional industry that does



not benefit from it. In general, data visualization has two main goals: to explore or to explain [SI11].

### Exploratory data visualizations

Exploratory data visualizations help to understanding the data by identifying its features, relations outliers or trends. It is part of the data analysis phase, and is used to find the story the data has to tell. The focus here is not on a single story but on discovering many small stories in the visuals. The purpose is to present the data in a way that the viewer is able to notice the obvious, and discover surprising insights [Tay14]. One example is the interactive visualization that updates in real time: Google Hot Trends<sup>1</sup>. This visualization does not highlight any single search query, but rather allows the user to explore any part of the visual to find out what users are searching for at the moment.

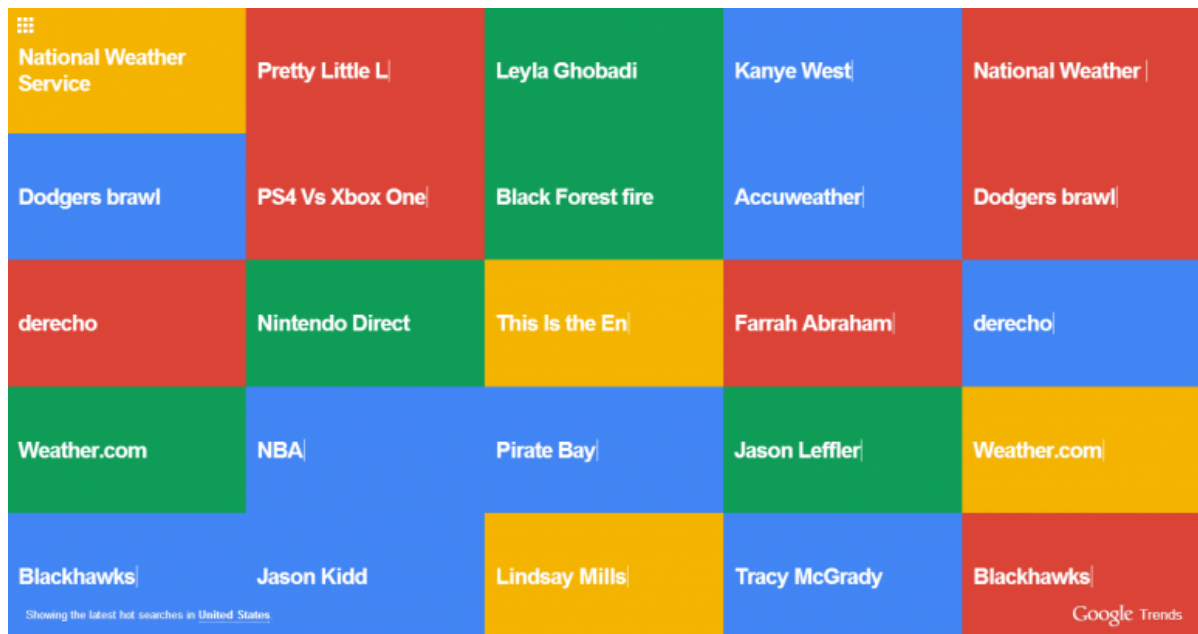


Figure 2.9: Example of exploratory explanation data visualization

### Explanatory data visualizations

This type of visualization is useful to support a story people want to tell. In contrast to exploratory data visualization, this type of visualization is part of the presentation phase,

<sup>1</sup><https://trends.google.com/trends/hottrends/visualize?nrow=5&ncol=5&pn=p15>

that is why the design is important here. As these visuals tend to remove the noise and distraction from the main narrative, they also tend to be static and not interactive. The purpose of this type of visualization is mainly to answer a question. However they are also useful when supporting a decision or communicating information. In the business sector, they can be easily recognized in dashboards, business presentations, training materials, and marketing content. They are also used in the media for advertising, print and television journalism, and political campaigning [Tay14]. For example, Figure 2.10 clearly answers the question: In which country the most sales were made [Pea16]?

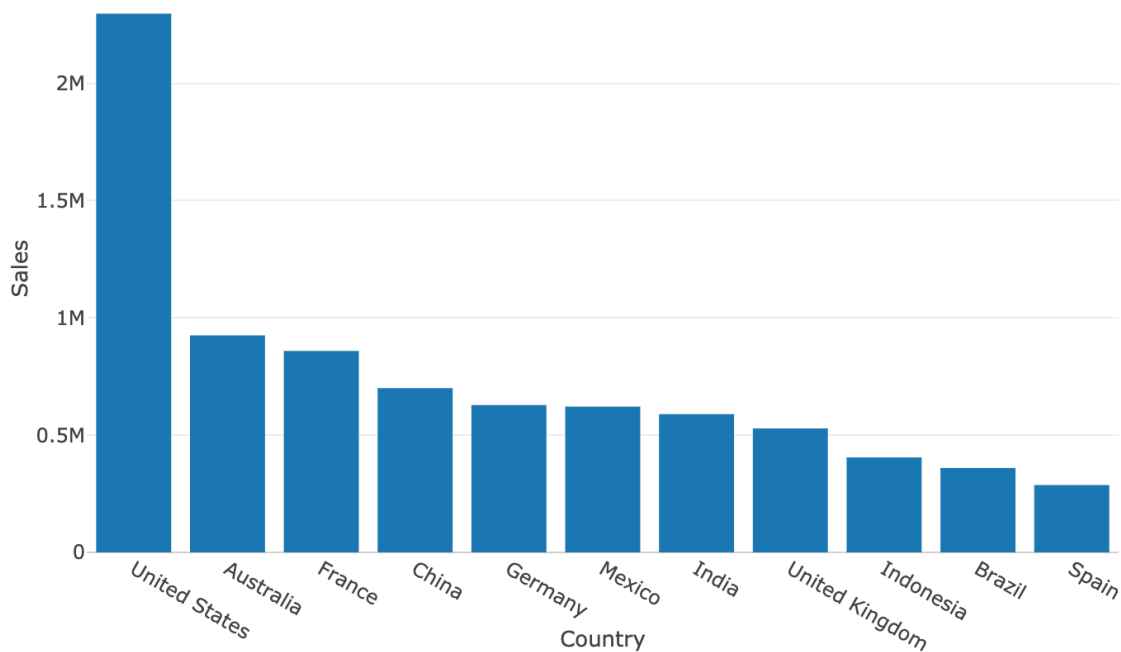


Figure 2.10: Example of explanatory data visualization

### Hybrids: Exploratory Explanation

These type of visualization includes both explanatory and exploratory visualization. Usually involves interactive interface allowing user to choose and constrain certain parameters, thus discovering insights the data-set may have to offer [SI11]. One example of this model would be interactive maps (e.g., Google maps). They provide driving directions from point A to B (Explanatory) and by zooming, and panning to discover the surrounding areas (exploratory) [Tay14]. This type of designs provide a certain degree

of freedom to discover from the information presented. The most commonly used visualization on blogs is the tag clouds. They highlight the most-used tags (exploration) and also show many less frequent tags.

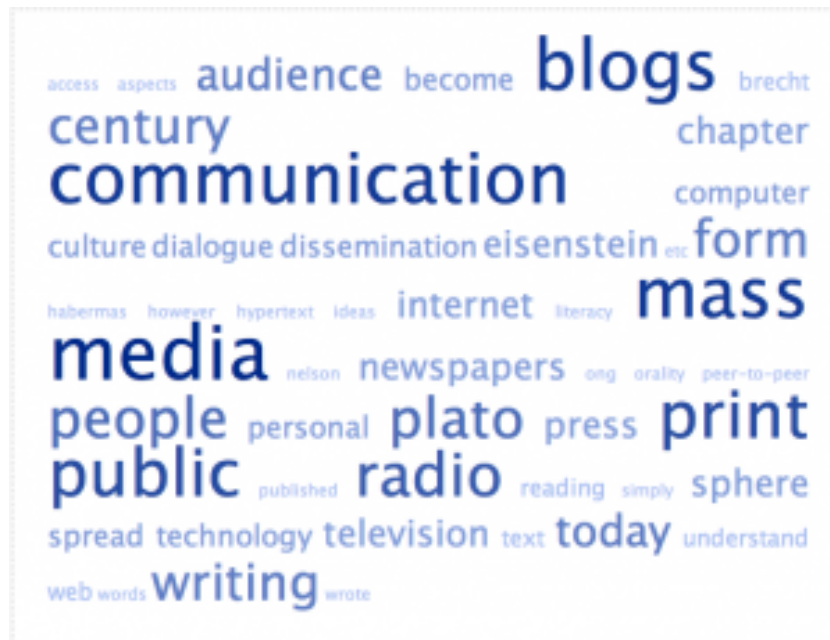


Figure 2.11: Example of hybrid data visualization

## 2.3 What is good data visualization?

Data visualization is not about making something visible to people. The main goal is to successfully communicate a point eliminating misleading information or errors. With a large number of chart types available, each performing good for a specific purpose, it is crucial to choose the right type. Picking a wrong form is one of the most common and critical visualization mistakes [Ste18].

However, selecting the right chart is not a trivial task and depends on many factors. It can depend on the nature of the data, its type, the visualization purpose or the human perception. One way to prevent distortion and misinterpretation is to consider the human perception, how accurate people perceive certain graphical objects, their size, color or position [CM84]. The following sub-chapters provide definitions and examples of good charts from different perspectives.

### 2.3.1 Graphical perception

Cleveland and McGill are one of the first scientists who focused their research on graphical perception. In many studies they conducted, the researchers were testing different chart types in order to understand which type people can understand easy and accurate. The result of these studies is shown in Figure 2.12. It presents a hierarchy of graphical objects ordered by most to least accurate and easy to understand for humans.

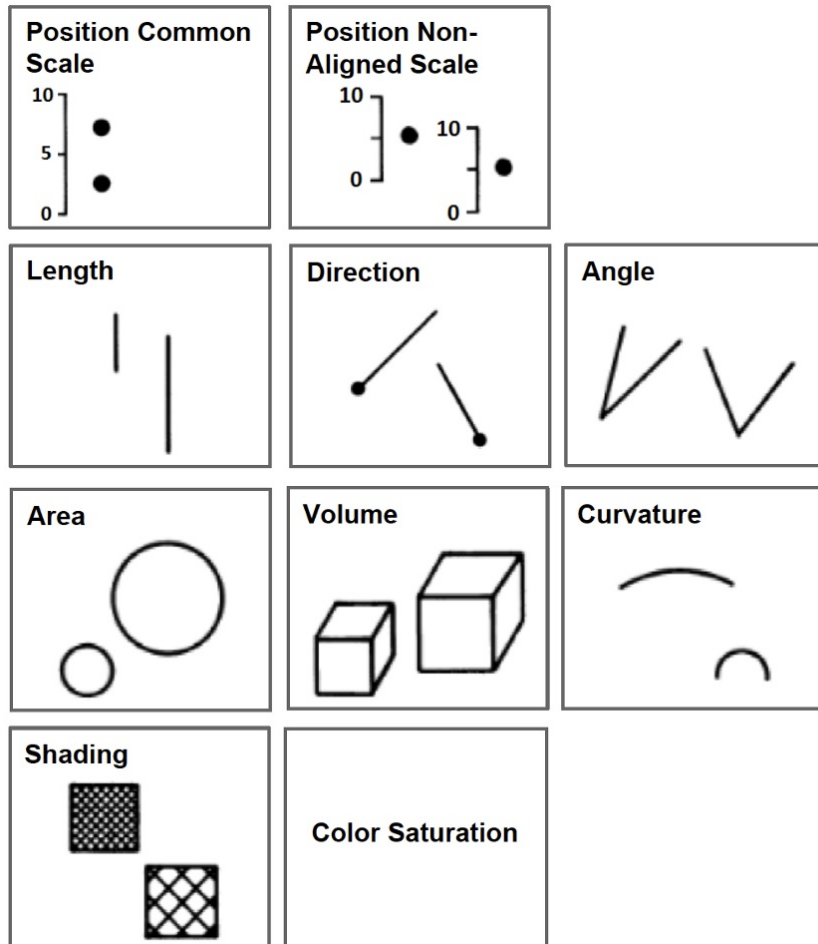


Figure 2.12: Hierarchy of chart types ordered from most to least accurate by Cleveland and McGill

At the top of Figure 2.12, Cleveland and McGill explain that position on a common scale is the easiest visualization type for people to understand with high accuracy. Next best is position on non-aligned scale, meaning people can understand more than one chart with dots as long as all charts have same scales.

On the next level are the length, direction and angle. With the first, bar charts are presented, next are the line and slope charts and finally, pie charts use angle to show the data. A further research has been done focusing mainly on these three graphical objects, in order to ensure their order in this hierarchy. The findings show that the angle charts produce most errors.

According to the study, people interpret area, volume and curvature not very accurate. Bubble charts use area to present data, all 3D chart types consider the volume of the objects and visualizations like the donut charts are shown with curvature. The last place on scale of this hierarchy is the color saturation. The adapted version of this hierarchy by Evergreen places this graphical form aside. She notes that the pattern fill used to shade the graph back in 1984 caused optical illusion, thus placing this form at the last place. However the advanced technology today allow shading with different colors. A new research found out that people can accurately distinguish between four shades of one color [War13].

The point of this hierarchy is choosing a chart that belongs higher in the hierarchy, so it can be easily and accurately interpreted. However, not every data can be presented with the same chart type. In addition to this hierarchy, other factors need to be considered [Eve17] like data-related criteria.

### 2.3.2 Visualization clarity

Good charts are clear, insightful, visually encoded so relevant patterns are easily noticeable and well organized [Cai16]. Their clarity makes them simple to read by focusing the reader on relevant elements. To reach clarity in chart, only important feature are highlighted and irrelevant elements are part of the background [Yau13].

An example from Yau regarding clarity in charts is shown in Figure 2.13 [Yau13]. The scatter-plot presents NBA players' usage percentage versus points per game. It is important to note that all visual elements are on the same level, meaning the dots, fitted line, grid lines and the border are same color and thickness, thus making this chart difficult to focus on a single element.

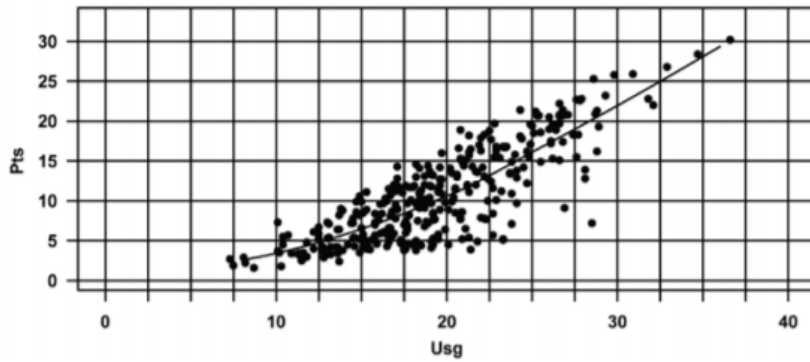


Figure 2.13: Example of scatter plot having all visual elements on same level

In contrast, Figure 2.14 have an accent on the main point - the correlation between two data attributes. The importance of visualization clarity is shown here, by focusing the reader on the fitted line with highlighted color and width. It can be noticed that all other elements are dimmed such as the width of the grid lines, their color and value labels adjusted.

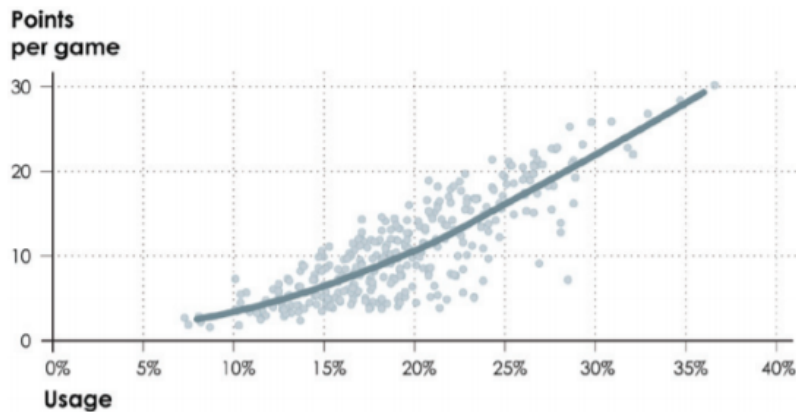


Figure 2.14: Adjusted scatter-plot with focus on the fitted line

Another aspect of reaching chart clarity is by allowing the user easily to compare values. This point is crucial for visualizing data. One visualization is not helpful if it does not fill this basic requirement.

A good visualization is not confusing and unclear. Having a clutter in a chart contributes to confusion and difficult readability. Enough space to divide the visual elements in one chart or to divide many charts result in decreasing the clutter.

### 2.3.3 Data attributes

Data is another source for influence in designing a visualization. Depending on the type, cardinality or visualization purpose, a decision for right chart type can be made [SI11].

#### Types of data

Different types of data may require different visualization types to reveal its aspects. Time series data present change over time, therefore relevant charts are: line, bar, area or slope charts, whereas numerical attributes can be presented with line, scatter-plot or bubble charts [Pow18]. The charts in Figures 2.15 and 2.16 show an altered example from Knaflic showing differences between line and bar charts when visualizing time series data [Kna15]. Same data is presented. Here the line chart has advantage over bar chart as it allows us to see even small changes easily (e.g. in the third month films have a small increase) for a long period of time.

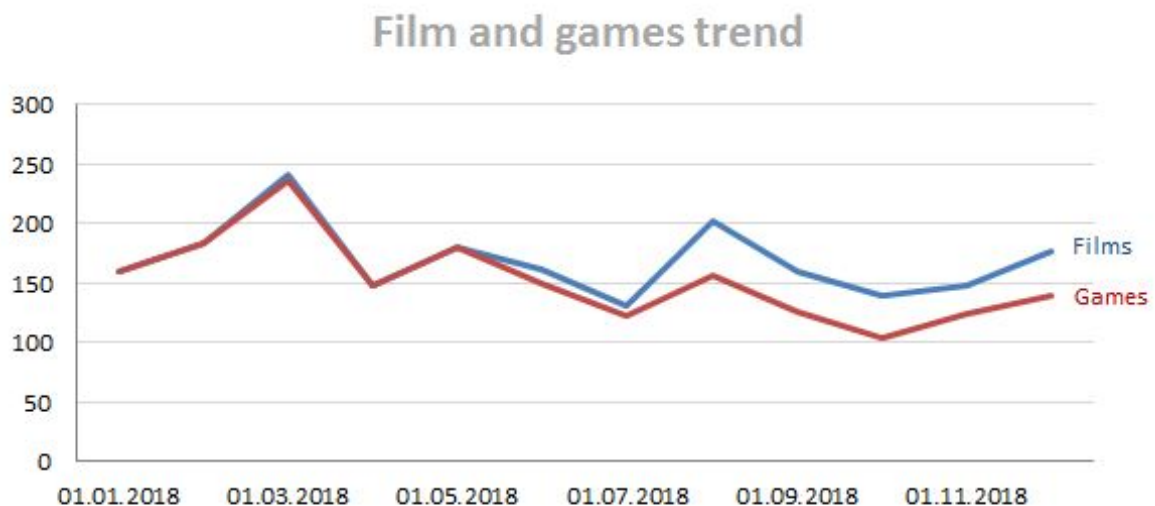


Figure 2.15: Good line chart showing trend of *Films* and *Games* over a year

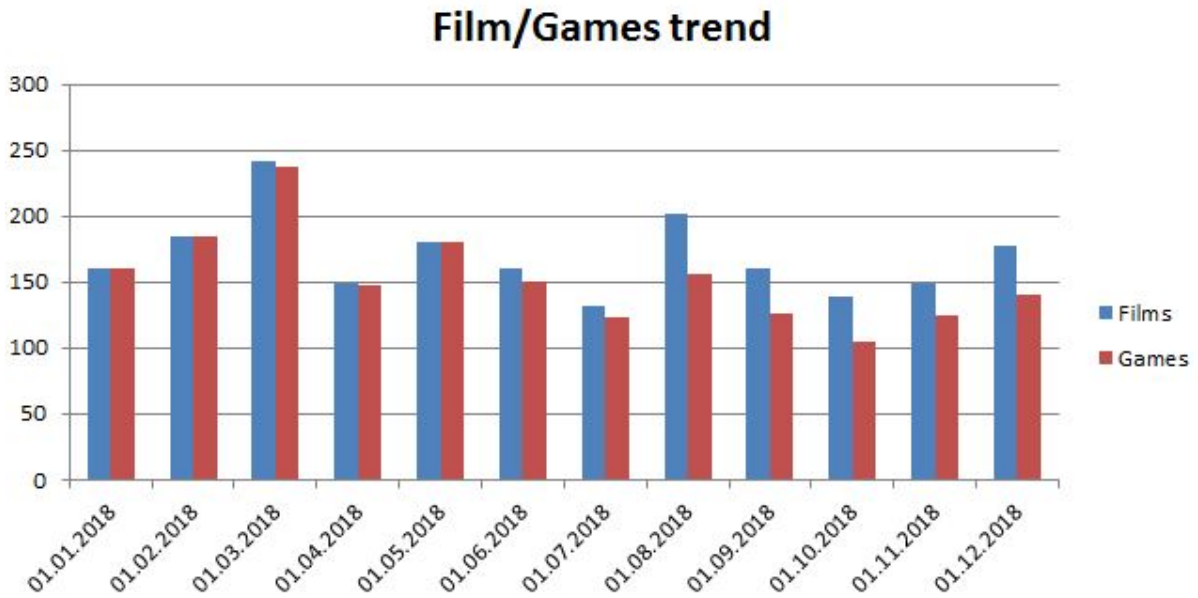


Figure 2.16: Grouped bar displaying trend of *Films* and *Games* over a year

### Dimensions

Data dimensions are as well good indicator for selecting appropriate chart type. Different chart types are well suited to show different number of categories or dimensions. Evergreen suggests that two categories could be best presented with side-by-side charts, slope-graphs or multi-graph series as given in Figure 2.17 [Eve17].

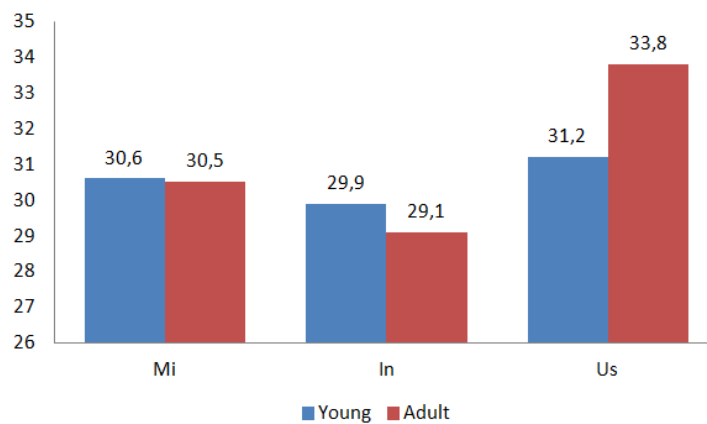


Figure 2.17: Example of good side-by-side bar chart

Figure 2.18 shows an example of chart where too many dimensions are charted on side-



by-side graph, thus making it difficult for the user to make comparison across multiple columns. A better solution for this data is a multi-graph series as given in Figure 2.19 as it can be easily comprehended. This way we prevent to have over complicated chart holding too much information [Eve17].

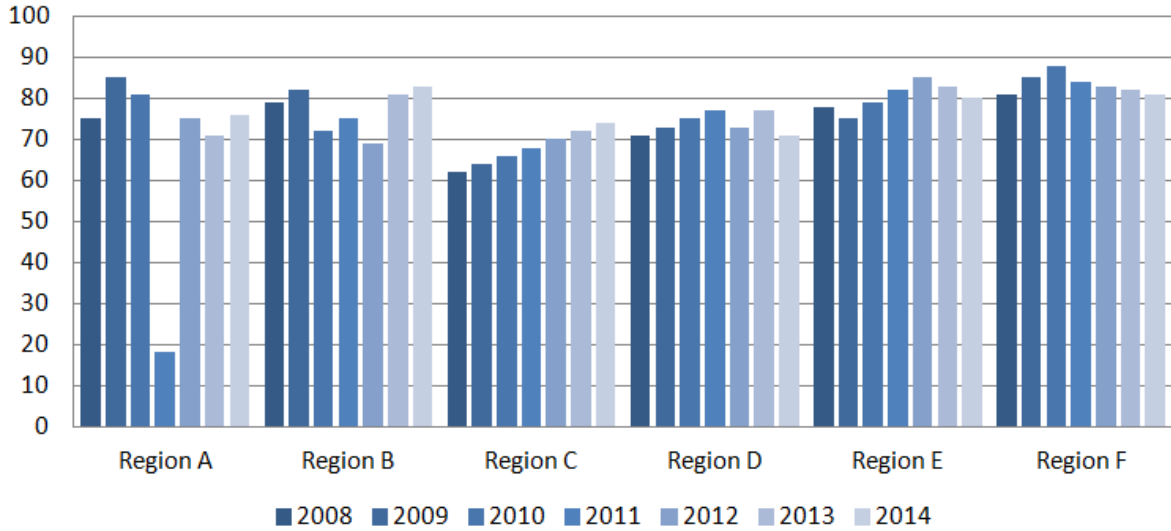


Figure 2.18: Example of bad side-by-side bar chart

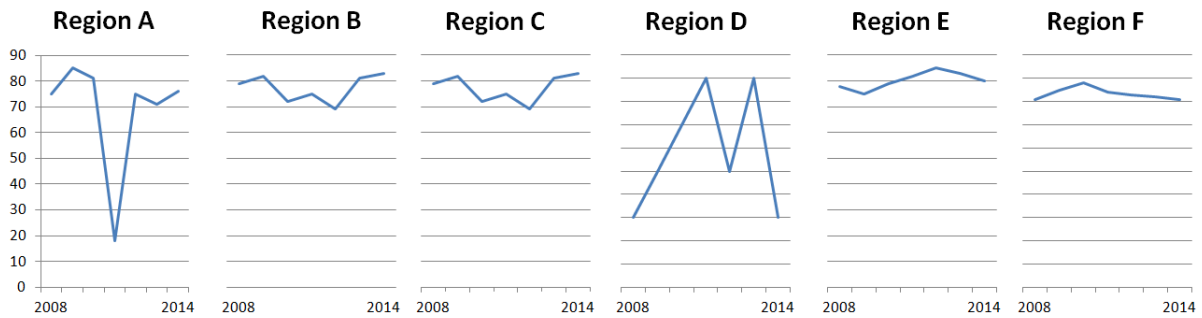


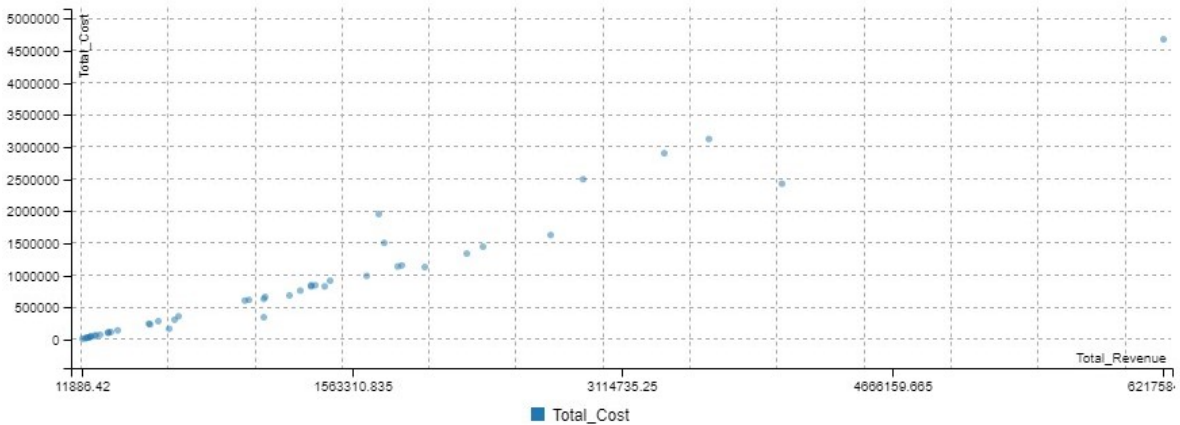
Figure 2.19: Example of a good multi-graph

### Relationship between data attributes

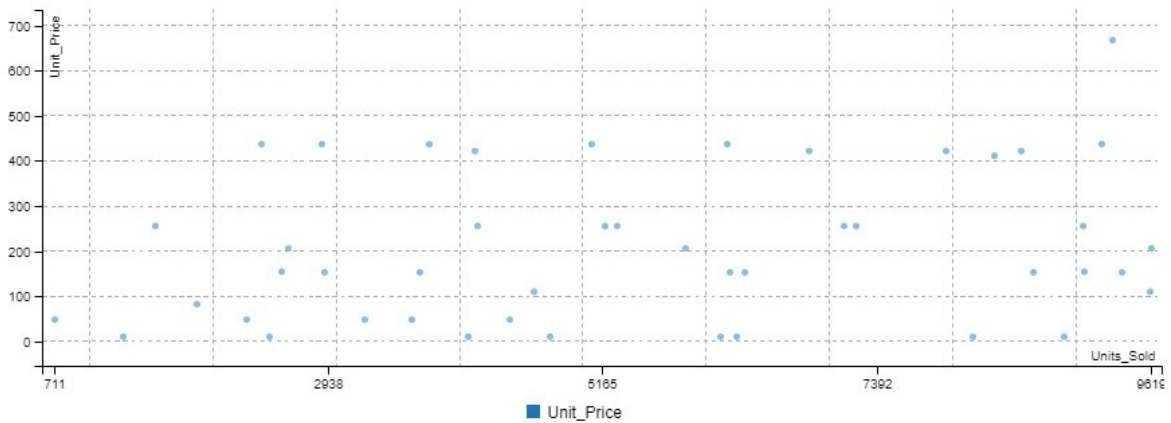
In the chart chooser (given in Appendix G) developed by Dr. Abel, a relationship between two or more numerical attributes can be shown with scatter-plot or bubble chart, depending on the number of numerical attributes. Third kpi would indicate selecting a bubble chart. Examples of charts when such relationship exist is given in

## 2 Fundamentals

Figure 2.20a and when no correlation exist between the two measures is given in Figure 2.20b.



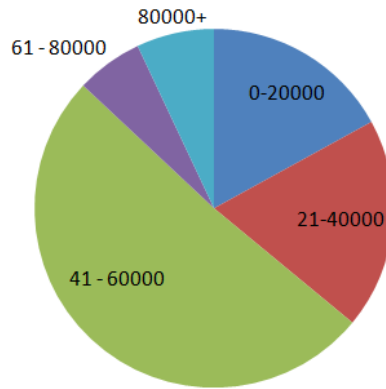
(a) Strong correlation between measures visualized with scatter-plot



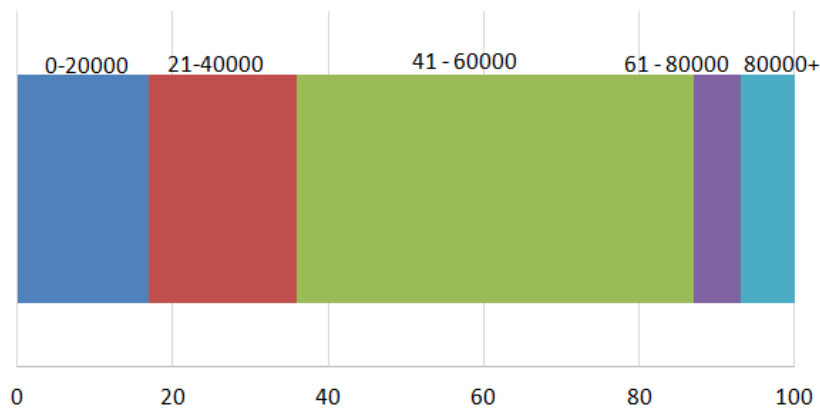
(b) No correlation between measures visualized with scatter-plot

Figure 2.20: Showing correlation between numerical values

Stacked column bars or pie charts are considered to be relevant when the data attributes have part-whole relationship [KNA13]. As in the previous point (*Dimensions*), the number of category should decide whether to choose pie or stacked-bar. Data with more than four categories requires choosing stacked-bar [Eve17]. Therefore one good alternative to the pie chart from Figure 2.21a is the stacked-bar given in Figure 2.21b because of two reasons: a) clearly states the part-whole relation and b) it is easy to compare the categories. For visualizing part-whole data over a period of time, area charts are good choice as they combine both functions of such data: to show change over time and the part-whole relation [Eve17].



(a) Too many categories shown on pie chart



(b) Alternative to pie chart for charting more categories

Figure 2.21: Good and bad example of charting part-whole relation with pie and stacked bar charts

## 2.4 Summary

In this chapter, we defined what is data and data visualization. We described ways of encoding data by using each of the following preattentive attribute: form, color, spatial position and motion. The most common visualization types use form and spatial position to visualize the data and enable comparison between categories. Three purposes for data visualisations were described: 1) to understand the data, 2) to answer questions or 3) combination of both.

Based on the human perception good visualizations consider the amount of data presented and the perception of graphical objects used to encode values. Most accurately

people understand charts that use position on common and non-aligned scales (e.g. bars, lines, dots). Least accurate are charts which use colors and shading to present the data (e.g. heat-maps).

Good charts highlight the data and keep in the background other graphical objects like grid lines or color, additional lines, border and background color.

The third aspect of good visualization is based on the data attributes. We referenced three data-related aspects relevant when choosing a good chart type:

- Data type. Time series data is charted with line or area charts while categorical data with a type of bar charts (stacked, grouped), multi-graph or slope.
- Number of dimensions. Two dimensions can be presented good with grouped bar charts, slope charts. The number of categories per dimensions as well needs to be considered. Multi-graph series are good for displaying many categories. Pie charts are limited to four categories, while stacked-bar chart more than four.
- Relationship between data attributes. Scatter plots and bubble charts are used when the data variables have strong correlation. Part-whole relation to be shown with pie, stacked-bar or area chart

The definitions given in this chapter we use further:

1. To create criteria for analysing related visualization tools
2. To define a utility metric for good visualization

## 3 Related work

### 3.1 Tools and Research Approaches

As data is produced at a very fast pace, the need to analyze this data has been increased as well. The result is the emerging of many tools mainly providing customized data visualization that require user interaction in configuration of data. However, the research done in this area has resulted in developing new frameworks which provide automation as well as recommendation of good or interesting charts. They take the data-set, do a certain transformation over the data (e.g., grouping, sorting) and recommend what could be considered as good or interesting.

This chapter focuses particularly on those tools that support automation and recommendation of good charts. For this purpose, the research study on Business Intelligence (BI) tools done by Gartner<sup>1</sup> is taken for objective selection of automated visualization tools. This study evaluates 15 criteria varying from data management to visual appalling. As a result, a Magic Quadrant is produced where tools are divided in four groups: Niche players, Visionaries, Challengers and Leaders [How+19].

For the purpose of this Chapter, a list has been created consisting of commercial tools reported in the 2019 Gartner Magic Quadrant given in Figure 3.1 and some of the popular scientific tools and is presented as follows:

- IBM: Cognos Analytics;
- SAS: Visual Analytics;
- MS: Microsoft Office Excel;
- Tableau

---

<sup>1</sup><https://www.gartner.com/en>

### 3 Related work

- RepGrids. Exploring the Visualization Design Space with Repertory Grids;
- ERAD. Efficient Recommendation of Aggregate Data visualizations;
- DeepEye: Towards Automatic Data Visualization;
- SeeDB: Efficient Data-Driven Visualization Recommendations to Support Visual Analytics



Figure 3.1: Gartner’s Magic Quadrant 2019 for Analytics and Business Intelligence Platforms

Additionally a matrix with evaluation criteria has been produced and each tool has been further analyzed to understand which features related to automatic detection are present and what is missing. Later this matrix will help when defining criteria for our metric for good visualizations.

#### 3.1.1 Commercial tools

In the following sub-chapter, four commercial tools chosen from each of the four groups from Gartner’s Magic Quadrant 2019 will be introduced and analyzed by providing background information.

#### **IBM Cognos Analytics**

One of the Niche players on the market according to Gartner Magic Quadrant for 2019 is the IBM's BI platform - Cognos Analytics. Primarily focusing the platform only on Cognos installed base, the vendor has focused only on building augmented capabilities into Cognos Analytics, resulting in slowing down the process of innovation [How+19].

IBM Cognos Analytics is a modern analytic and BI platform supported with augmented analytic capabilities. Cognos Analytics offers enterprise reporting, governed and self-service visual exploration and augmented analytics in a single platform. The vendor of Cognos analytics, IBM is one of the first to release augmented analytics capabilities. Its latest version includes an artificial intelligence (AI) assistant interface and native natural language generation (NLG). The product allows analysts, data scientists, data engineers and others to discover and identify the lineage of enterprise data assets [Vol08].

#### **SAS Visual Analytics**

SAS Visual Analytics provides integrated environment for governed data discovery and exploration. According to the vendors, any user can examine and understand patterns, trends and relationships in data<sup>2</sup>. Its easy-to-use analytic and visualizations help to get insights from data to better solve complex business problems. That is as well one of the reasons why Gartner positions this product as Visionary, together with the product's robustness, migration and global presence.

The product combines data preparation, reporting and visual exploration. With the augmented analytic capabilities, the platform provides explanation regarding the importance of variable analyses (which variables contribute to an outcome). Voice integration with personal assistants is supported, and additional chat-bot integration. Geo-spatial configured data is charted by using map charts.

The tool performs some of the mathematical operations like data calculation, aggregation, custom binning (moving into a small number of groups) to allow better interpretation and presentation of results. Custom sort allows to order category data items by

---

<sup>2</sup>SAS Visual analytics fact sheet. Available at: [https://www.sas.com/content/dam/SAS/en\\_us/doc/factsheet/sas-visual-analytics-on-sas-viya-108779.pdf](https://www.sas.com/content/dam/SAS/en_us/doc/factsheet/sas-visual-analytics-on-sas-viya-108779.pdf)  
Accessed on: 23.03.2019

characteristics (e.g., products, customers) most relevant for the user. One way filtering allows to link content (e.g., visualizations, reports).

The user interaction is necessary for data configuration: pre-selection of measures and categories. The auto-charting feature visualizes the configured data in a compelling way by implementing advanced data visualizations and automated guides analyses that highlights relationships in data with comprehensive explanations. This automated feature discovers key relationships, outliers, clusters, trends and more revealing insights. It automatically chooses the graph best-suited to display selected data. The auto-charting in SAS Visual Analytics takes into account the cardinality of the data and adjusts the visuals accordingly<sup>3</sup>. These rules are given in Table 3.1. The visualizations include histograms, donuts, bars, heat maps, scatter, correlation matrices, line charts with forecasting, etc.

Data items	Chart type
One measure	Histogram
One category	Bar chart
One date-time category and any number of other categories or measures	Line chart
One geography and up to two measures	Geo map
One geography and three or more measures	Bar chart
Two measures	Scatter plot or heat map
Three or more measures	Scatter plot matrix or correlation matrix
One or more categories and any number of measures and geographies	Bar chart

Table 3.1: SAS' criteria for chart type selection

### Microsoft Office Excel

The version of Microsoft Office 2013 supports chart recommendation for categorical and numerical data. The actual algorithm of how Microsoft recommends charts is not

<sup>3</sup>Working with Automatic Charts. Available at: <http://support.sas.com/documentation/cdl/en/vaug/65747/HTML/default/viewer.htm#n1xa25dv4fiyz6n1etsfkbz75ai0.htm>  
Accessed on: 23.03.2019



public. To use the function it is enough to select the desired columns for visualization and choose 'Recommend charts', users get a recommended list of good charts. The automation algorithm does data analysis based on heuristics, resulting in a list of chart types with live preview feature. Most commonly recommended charts are: bar, line, heat-map and a multilevel pie chart - sunburst.

The new feature intelligently recognizes geo-related data, generates a map chart, displaying specific facts or dimensions. For numerical values, the program is providing automatically mathematical operations like sum, average, distribution metrics and understands when the data has part-whole relation. It also successfully detects number formats (percentage, currency) and provide respective charts. It performs assessment whether the data has repeating values, such as categories of sales transactions. In such cases, a Pivot-Chart (with the appropriate grouping applied) is recommended instead of a regular chart. This Pivot-Chart as in the previous versions of the product can be further configured by the user, selecting features for specific axis.

## **Tableau**

Tableau offers an interactive, visual-based exploration that enables users to access, prepare, analyze their data without technical skills or coding. The platform is available as a stand-alone desktop application or integrated with a server for sharing content; Tableau Online is the cloud-based offering. Due to its popularity, high customer satisfaction and strong road-map, this product is places as a Leader in the Gartner Magic Quadrant.

The data manipulations provide wide range of sources for data to be uploaded, blended and visualized with consideration of visual perception. In addition to the worksheets for charts generation and analyse, users can create stories and customized dashboards.

A story represents a sequence of visualizations that work together to convey information<sup>4</sup>. This information can tell a data narrative, provide context, demonstrate how decisions relate to outcomes, or to simply make a compelling case. As for regular worksheets, it is possible to create, name, and manage stories.

A dashboard is a collection of several views. It help users to display many views at once, rather than navigate to separate worksheets. It is interesting to note that, the

---

<sup>4</sup>Stories. Availble at: <https://onlinehelp.tableau.com/current/pro/desktop/en-us/stories.htm>

Accessed on: 29.03.2019

### 3 Related work

data in sheets and dashboards is connected, change in one sheet will result in change in a dashboard and vice versa.

Tableau Desktop is data analysis and data visualization tool that comes with the *Show Me* feature. It is there to help users who are starting out with the program. Once desired data for visualization is properly drag'n dropped (selected rows and columns), this feature highlights available charts and fades those which are unavailable. The first type of charts are determined by the number of measures, dimensions, bins, etc. Previously defined quantified rules for each data property is assigned for each chart. Table 3.2 describes some of the rules used in creating automatic views<sup>5</sup>.

Chart type	Rule
Text table	Adding a dimension first produces a text table (or cross-tab). All subsequent clicks on fields result in refinement of the text table.
Bars	Adding a measure first and then a dimension produces a bar view. All subsequent clicks result in refinement of the bar view, unless a date dimension is added, at which time the view is changed to a line.
Line	Adding a measure and then a date dimension produces a line view. All subsequent clicks result in refinement of the line view.
Continuous Line	Adding a continuous dimension and then a measure produces a continuous line view. Subsequent dimensions result in refinement of the continuous line view. Subsequent measures add quantitative axes to the view.
Scatter	Adding a measure and then another measure produces a scatter view. Subsequent dimensions result in refinement to the scatter view. Subsequent measures will create a scatter matrix.
Maps	Adding a geographic field produces a map view with latitude and longitude as axes and the geographic field on the Level of Detail shelf. Subsequent dimensions add rows to the view while subsequent measures further refine the map by adding size and color encoding.

Table 3.2: Rules for creating automated visualization in Tableau

---

<sup>5</sup>Start Building a Visualization by Dragging Fields to the View. Available at: [https://onlinehelp.tableau.com/current/pro/desktop/en-us/buildmanual\\_dragging.htm](https://onlinehelp.tableau.com/current/pro/desktop/en-us/buildmanual_dragging.htm)  
 Accessed on: 29.03.2019

### 3.1.2 Research Approaches

#### Exploring the Visualization Design Space with Repertory Grids

This research paper proposes a user-centric technique called "Repertory Grid technique" in order to explore the visualization design space. This technique is based on the personal construct theory, in which every person creates own ways of seeing the world [KW18]. Kurzahls and Weiskopf researched the differences between users and how they interpret various visualizations. Two groups of users were involved: experts and non-experts. They were asked to elicit constructs for visualization of their choice (Figure 3.2). Based on the answers given, a repertory grid was developed for expert and non expert users. The expert constructs were objective, describing visual mapping (like visual primitives ,line based, area based), color mapping (random colors heat map) , the use of text (legend no legend)) and composition aspects (number of views, alignment), whereas the non-expert construct focus mainly on subjective, visual experience. Figure 3.3 shows a repertory grid which displays all elicited constructs from the users for the rating of elements.

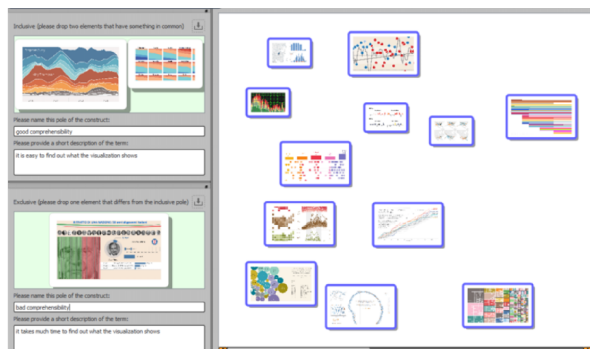


Figure 3.2: Elicit construction [KW18]



Figure 3.3: Visualization assessment [KW18]

Another objective of the research paper is to define similar visualizations and the reasons for being similar. Based on the elicited construct from the involved participants, 6 clusters were identified (Figure 3.4). Bar charts were clustered together because these

### 3 Related work

charts were seen as clutter free, single techniques with no geo-related layout and that is why they were grouped together.

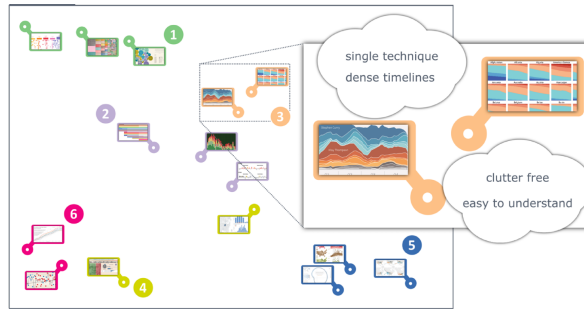


Figure 3.4: Analysis of visualization clusters [KW18]

### Efficient Recommendation of Aggregate Data visualizations

Ehsan, Sharaf and Chrysanthis wrote a research paper for the IEEE Transaction on Knowledge and Data Engineering journal published in February 2018<sup>6</sup> describing the problem of recommending interesting bar visualization for numerical dimensions by choosing the right binning parameter [ESC18].

The authors argue that the deviation-based metric has been shown to be effective when visualizing categorical data, but when it comes to presenting numerical value, the utility of such a metric is lower. Therefore, binned aggregation is required to group numerical dimension values into intervals. This parameter allows to reduce the clutter and sparsity in the generated visualization as well as to group similar data together (group player according to their minutes played on the field).

According to the authors, the resulting visualizations from the framework are:

- Interesting, when a certain view reveals new insights about the data. It is measured using deviation metric.
- Usable, if a visualization is able to provide understandable uncluttered representation. It is measured with the width of the bin and,
- Accurate, to capture the characteristics of the analyzed data, measured with accuracy metric.

<sup>6</sup><https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=69>

Together these metrics comprise the multi-objective utility metric used to assess the utility of a certain view. As combining them together requires more computational time, the authors propose a search schemes which reduce the processing time. The first searching scheme evaluates the multi-objective utility function where different objectives are computed progressively. The second scheme is able to detect the high utility views earlier. The third and last searching scheme is memory aware, meaning that it provides the same pruning power as the second but has memory usage constraints in order to prevent multiple views to be considered for recommendation.

### DeepEye: Towards Automatic Data Visualization

DeepEye<sup>7</sup> is an automatic data visualization system. The research work is published as part of the International Conference on Data Engineering in France in 2018 [Luo+18]. The framework works in a way that for a given data-set (or selected from already defined data-sets) efficiently to discover interesting visualizations to tell compelling stories.

Figure 3.5 provides an overview of the framework. It consist of two components.

- The first one, named offline component is responsible to rank visualizations based on examples trained on ML models: binary classifier and a learning-to-rank model. As well experts' knowledge is included when specifying rules as partial orders (e.g., attribute importance, attribute correlation).
- The second, online component uses the trained classifier to determine whether an identified visualization is good or not.

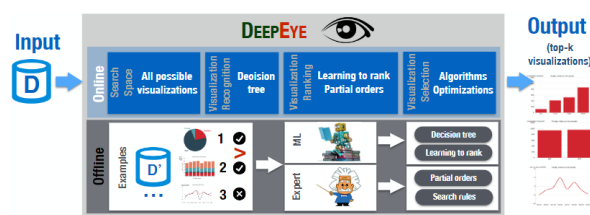


Figure 3.5: Overview of DeepEye [Luo+18]

With the help of machine learning techniques, the creators tackle three problems for automatic data visualization: visualization recognition, ranking and selection.

<sup>7</sup><http://www.deepeye.tech/>

**Visualization Recognition** uses the binary classifier from the offline component to assess whether a combination of columns and an identified visualization type will generate good or bad visualization.

**Visualization Ranking** is picking a better visualisation out of two. Therefore, the machine learning model learning-to-rank gives two feature vectors to a learning function  $F(\cdot)$  (previously trained on examples) to make a decision. The feature vector includes: the number of distinct values in a column, their ratio, number of tuples, max and min values in a column, data type of a column, the correlation of two columns and the visualization type.

**Visualization Selection** outputs a ranked list, from the inputted visualization nodes with their feature vectors.

#### **SeeDB: Efficient Data-Driven Visualization Recommendations to Support Visual Analytics**

The framework adopts a deviation-based metric as a visualization utility in order to recommend interesting data visualization. In the research paper published in 2015 as "Proceedings VLDB Endowment", the authors propose the SeeDB, a visualization engine to facilitate fast visual analyses [Var+15].

The focus is on two challenges: scalability and utility. In order to evaluate the utility of big number of visualisation in a timely manner, the framework introduces pruning optimization and sharing optimization. Deviation from a reference data-set or data column is taken as a utility metric for judging the interestingness of a visualization. The larger the deviation between the data-sets, the more interesting visualization is.

The framework has a mixed-initiative front-end which allows the user to navigate easily. Figure 3.6 provides an overview of the interface containing four parts. (A) database connector and query builder, (B) visualization builder, (C) visualization display and (D) place for displaying recommended visualizations.

Developed as a web based solution, it takes the user input and provides visualisation generated by a server. The server has a view generator, responsible for parsing the input, querying the system metadata and to generate list of visualisation, and execution engine, responsible for query evaluation using optimizations (Figure 3.7). Interesting visualization (with high deviation) are then recommended to the user.

### 3 Related work



Figure 3.6: SeeDB Front-end [Var+15]

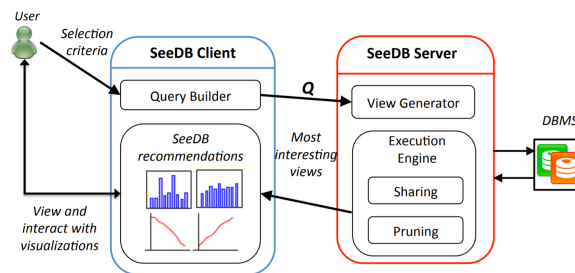


Figure 3.7: SeeDB architecture [Var+15]

## 3.2 Analysis of Tools and Research Approaches

After describing the tools' features, next step is to perform a comparative analysis. Therefore, we use the three aspects of good visualizations described in sub-chapter 2.3. These definitions consider good visualization based on: the user perception, visualization clarity and the data.

The result of the analysis will identify the gaps which are not filled by the existing approaches in terms of features, data types supported or data operations not carried out. It will also help to point out which criteria are supported by most of the tools. For this purpose a matrix of the main criteria is defined and presented in the following sub-chapters.

### 3.2.1 Comparing Criteria

In order to compare the identified tools, two groups of criteria have been developed: visualization representation criteria and data-related.

Table 3.3 defines the first group of criteria for assessing the visual representation. For defining the criteria in this group we reference the findings regarding good visualization based on: the graphical perception, given in sub-chapter 2.3.1 and the visualization clarity, given in sub-chapter 2.3.2.

The second group of criteria includes data-related measures. For defining this list we consider the statistics that the related tools use in order to provide automation (described in sub-chapter 2.1.2) and the data-aspect for defining good visualization provided in sub-chapter 2.3.3. The definition of the criteria from this list is presented in Table 3.4.

<b>Criteria</b>	<b>Definition</b>
Automatic visualizations	Provide automated data configuration and automated generation of visualizations.
No chart types restriction	No restriction in working with chart types.
Clutter-free	Focus is on the data and dim other chart elements (density, width and color of grid lines, border color and width, axes values).
Interactivity	User is able to interact with the chart (e.g., click on the bars, lines and request additional information).
Graphical perception	Accurate decoding graphical forms. Follows the Cleveland and McGill's hierarchy when selecting chart types.
Multi-graph series	The tool generates a small set of charts, each representing a single category to prevent the clutter or cognitive overload when same data is presented in a single chart.
Design principles for multi-graph series	For generating multi-graph series, the tool uses: same axis values for all charts, all charts are ordered, distant, only horizontal grid lines and uses line chart type

Table 3.3: Criteria evaluating the visual representation



### 3 Related work

<b>Criteria</b>	<b>Definition</b>
Time series data	Detects temporal data in a data-set and generates time series visualizations.
Categorical data	Detects categorical data and generate categorical chart with two and more dimensions.
Numerical data	Works with numerical values and generates charts with two and more facts (e.g., scatter-plot, bubble chart).
Data type	Considers the nature of the data when selecting charts. Defined in sub-chapter 2.3.3, for each data type is defined a set of chart types (time series data with line charts or mutli-graph, numerical with dot charts and categorical with types of bar, pie slope or multi charts)
Cardinality	Number of dimensions and categories is indicator for selecting chart.
Deviation	Calculates deviation and its values is taken as a utility metric for automatic generation of charts.
Correlation coefficient	Measures the correlation between numerical values and suggests a chart based on the value. Valid only for quantitative data.
Part-whole relation	Part-whole relation is detected and appropriately charted, as described in sub-chapter 2.3.3.
Calculation of intersection	Calculates the intersection of lines in line charts. The number of intersection points in a chart is indicator for selecting a chart type. Relevant for time series data.
Aggregation	Performs aggregation of numerical values.
Reference data-sets	Additional data-set or data row is necessary to provide automatically charts.
Binning parameter	Groups the data and presents the groups as bars in the chart. Considers the number of bins and their width as criteria for automatically generating charts. Relevant for numerical data.



Data sorting	Categories and axis values in the chart are sorted (alphabetically on categorical chart or ascending order on time series charts).
Allow missing values	Missing date-time value is not generating interrupted line chart. Valid only for time series data.

Table 3.4: Data-related evaluation criteria

### 3.2.2 Tool Comparison

Table 3.6 presents a matrix as a result from analyzing the tools and research approaches with the criteria discussed in the previous sub-chapter. Here, the rows are the criteria, and the columns are the tools. Their intersection shows the existence or absence of a certain criteria in the tool. Color coding scheme is introduced in Table 3.5 for easy notation.

Table 3.5: Color coding scheme used for performing comparative analyse

	A feature does not exist in the tool
	A feature exists in the tool

Evaluation criteria	IBM	SAS	MS	Tableau	Rep Grids	ERAD	Deep Eye	SeeDB
Automatic visualization								
No chart types restriction								
Clutter free								
Interactivity								
Graphical perception								
Multi-graph series								

### 3 Related work

Evaluation criteria	IBM	SAS	MS	Tableau	Rep Grids	ERAD	Deep Eye	SeeDB
Design princ. multi-series	Red	Red	Red	Green	Red	Red	Red	Red
Time series data	Red	Green	Green	Green	Red	Red	Green	Red
Categorical data	Red	Green	Green	Green	Red	Green	Green	Green
Numerical data	Red	Green	Green	Green	Red	Red	Green	Red
Data type	Red	Green	Green	Green	Red	Green	Green	Green
Cardinality	Red	Green	Green	Green	Red	Red	Green	Red
Deviation	Red	Red	Red	Red	Red	Green	Green	Green
Correlation coefficient	Red	Red	Red	Red	Red	Red	Green	Red
Part-Whole relation	Red	Red	Green	Red	Red	Red	Green	Red
Calculation of intersection	Red	Red	Red	Red	Red	Red	Red	Red
Aggregation	Green	Green	Green	Green	Red	Green	Green	Green
Reference data-sets	Red	Red	Red	Red	Red	Green	Red	Green
Binning parameter	Red	Green	Red	Green	Red	Green	Green	Red
Data sorting	Green	Green	Red	Green	Red	Green	Green	Green
Allow missing values	Red	Red	Red	Green	Red	Green	Green	Green

Table 3.6: Evaluation criteria observed for tools' analyse

### 3.2.3 Strengths and Limitations

Each of the presented tool and research approach has strengths in specific points but also some limitations in other aspects. To have better overview over these different aspects, 3.7 shows a summary of the analysis.

Tool	Strengths	Limitations
IBM	<ul style="list-style-type: none"> <li>• Allows creation, running and managing different styles of reports (list, chart, map, financial).</li> <li>• Provides adding calculation to reports</li> <li>• Data filtering and sorting</li> <li>• Dynamically adding titles to reports and apply formatting</li> </ul>	<ul style="list-style-type: none"> <li>• Does not provide automation</li> <li>• Requires creating packages to run reports. Configuration needed</li> <li>• Users must run queries and reports to generate visualizations.</li> <li>• Has a default chart type. Remembers last selection of chart type for further use.</li> <li>• Complex interface for non-expert user. Require different web applications to create visualizations (Cognos Connect and report Studio)</li> </ul>
SAS	<ul style="list-style-type: none"> <li>• Provides automated visualization with additional guided visual analytics</li> <li>• Generates charts with many facts or dimensions</li> <li>• Works with both, time series and categorical data</li> <li>• Text visualization</li> </ul>	<ul style="list-style-type: none"> <li>• User has to select/define measures and dimensions (time dimensions)</li> <li>• It has strict rules for chart selection. Follows a single metric for automation.</li> <li>• Does not consider how cluttered a chart could be.</li> <li>• It neither includes human perception of graphical objects nor follows the elementary perceptual task when deciding upon charts.</li> </ul>

### 3 Related work

Tool	Strengths	Limitations
MS	<ul style="list-style-type: none"> <li>• Works with many different chart types.</li> <li>• Automatically provides set of recommended good charts.</li> <li>• Works well with number formatting (percentage and currency) thus presenting the right format in the charts.</li> </ul>	<ul style="list-style-type: none"> <li>• Supports charting for only two columns.</li> <li>• Charts mostly already aggregated data.</li> <li>• Time series data is not detected successfully, charting this attribute as a category.</li> <li>• Human perception is not considered.</li> <li>• Data is not always aggregated by dimension charted, and duplicate values are presented on the axis.</li> </ul>
Tableau	<ul style="list-style-type: none"> <li>• Works with many multiple file formats.</li> <li>• Automatically recognizes data features.</li> <li>• Possibility to connect columns from different data-sets.</li> <li>• Allows configuration in regards to the size, colors, description and selection of objects in one chart.</li> <li>• Automatically generates chart for selected data features.</li> <li>• Only tool that generates multi-graph series.</li> </ul>	<ul style="list-style-type: none"> <li>• User configuration required for charts generation.</li> <li>• Does not consider the clutter when numerical values have big fluctuation.</li> <li>• By default categorical data is sorted alphabetically.</li> <li>• Single metric for automated visualization (data cardinality and data types)</li> </ul>

### 3 Related work

Tool	Strengths	Limitations
RepGrids	<ul style="list-style-type: none"> <li>• Explores the visualization space to define visualization features relevant to a certain group of users.</li> <li>• Studies which charts are more relevant to which group of users (experts, non-experts).</li> <li>• Research which visualizations are seen as similar.</li> </ul>	<ul style="list-style-type: none"> <li>• Does not generate charts automatically.</li> </ul>
ERAD	<ul style="list-style-type: none"> <li>• Produces automatically clutter-free bar charts.</li> <li>• Considers multi-metric for recommending good bar charts.</li> <li>• Provides chart without noise.</li> </ul>	<ul style="list-style-type: none"> <li>• Limited in working with different data types. Works only with categorical data.</li> <li>• Considers only one chart type - bars.</li> <li>• Does not generate numerical charts.</li> </ul>
DeepEye	<ul style="list-style-type: none"> <li>• Recognizes different data types.</li> <li>• The generated charts are good and clutter-free.</li> <li>• Uses machine learning technique for detection of good charts.</li> <li>• Includes experts knowledge for chart selection.</li> <li>• Combines data attributes for recommending a good set of charts.</li> </ul>	<ul style="list-style-type: none"> <li>• Considers only four chart types.</li> <li>• Missing consideration of human perception of graphical objects.</li> <li>• Visualizes only single fact or dimension.</li> <li>• It does not calculate the number of intersection points in line charts, to prevent overload and hard interpretation.</li> <li>• Works with categorical and numerical data</li> </ul>

Tool	Strengths	Limitations
SeeDB	<ul style="list-style-type: none"> <li>• Generates interesting charts, referencing additional column or data-set.</li> </ul>	<ul style="list-style-type: none"> <li>• A single metric used as utility metric for automated creation of interesting charts.</li> <li>• Human perception is not taken into consideration.</li> <li>• Works only with categorical data.</li> <li>• Reference data-set is necessary to generate charts automatically.</li> </ul>

Table 3.7: Comparative analyse of tools for automated visualizations

### 3.3 Discussion

In this sub-chapter, specific points resulting from the tools' analysis will be discussed. These points are taken into consideration when defining our utility metric for good visualizations.

Most of the commercial tools do not provide automated visualization. All scientific prototypes use only a certain set of graphical presentation, thus focusing on certain data types ensuring clutter-free charts which is typical for the research prototypes. Users can upload time series or categorical data and the tools generate list of recommended charts showing combination of data attributes from one or many dimensions or one or many facts. The criteria for generating automated charts differs based on the tool and the goal they want to achieve. Most of the tools use similar techniques for data preparations, transformation and chart selection. Also, similar metrics for chart selection are used among the described tools. Data aggregation, binning and sorting can be found in all of the tools.

Not all of the tools provide good charts. Few of them generate charts based on a single metric (data type, cardinality of simple mathematical or statistical metrics) with the purpose to ensure the automation or generation of the interesting charts. Almost all of the tools neither consider the principles for human perception nor the elementary

perceptual metric. Therefore, it is not a surprise that none of the tools generate multi-graph series, because this type of chart is commonly used to prevent chart overload or clear the clutter.

User interaction is necessary in all tools (except in DeepEye) for defining data attributes in the data. The tool is not able to provide visualizations without data configuration. In DeepEye where automation is provided, time series data is not correctly detected and charted as a dimension, thus functionality of the chart is affected. Pie charts are used for presenting time series data showing multiple dimensions, therefore producing charts which are not functional but confusing.

Line charts are generated with lots of noise. The number of lines in the chart and the fluctuation of the data is not considered when generating such chart type. Stacked bars are produced with multiple groups or dimensions per groups, thus causing chart overload requiring more cognitive effort and time to perceive its message.

Additionally to the described features under consideration for the analysis, it was observed the user interaction with the tool. The following aspect were concluded:

- Every tool has different interface, but all of them provide clear and easy-to-navigate views.
- There exist limitations in the data file uploaded and its size.
- Only the commercial tools allow the user to upload data from different sources (connecting different databases).
- All of the commercial tools require user interaction while configuring the data in terms of selecting/defining measures and dimensions.

## 3.4 Summary and Conclusion

Eight tools and prototypes were described and analyzed in this chapter. After careful analysis, it was found out that all tools are either data-type specific or chart-type specific. In most cases the generated visualizations are of a single type or from a small limited set of chart types, and consider only a single criteria to provide the user with charts. The analysis have shown that there is only one tool that provides automated



visualization. All other tools provide rather half-automation. Although the tools generated visualization without the need of the user to specify chart types, this process would have not been possible without configuring the data. No single tool considers all of the perspectives of good visualizations as already discussed in Chapter 2.3. To conclude, the performed analysis made in this chapter supports the need for developing a tool that fully automatically generates good visualizations.

# 4 Utility Metric for Generating Good Visualizations

Based on the analysis performed in the previous chapter, it has been discovered that the existing tools have some limitations in automatically generating good visualizations. The next step is to develop a utility metric that will solve this problem. For this reason, in this chapter, a new utility metric for good visualization will be developed. This metric combines the findings from the tools' analysis and the three aspects of good visualization discussed in sub-chapter 2.3.

Here, we define an approach which fully automatically suggests good visualizations to the user. To be able to do so, first we describe how we detect all possible visualizations from a data-set. Secondly we describe how we rank the detected visualization. Therefore we give our definition of good visualization and define the criteria for our utility metric for good visualization. Both, the definition and the utility metric consider the findings made in Chapter 2 about good visualization. and the result from the analyses in previous Chapter.

## 4.1 Visualization detection

This module is responsible for finding all possible combinations of data attributes required to generate a visualization from a given data-set. The result of this module is taken as an input in the second module. We recognize three types of data attributes:

- Time dimensions,
- Dimensions
- Facts

A time dimension is any column that has time series points as values(days, months, years, week days, or combination of all these). For a detected time series column its format should also be known for the purpose of proper chart formatting. Facts are distinguished from dimensions if they present numerical values.

We define the following combinations of data attributes to detect visualizations:

- Time series chart: one time dimension, one dimension and one fact.
- Categorical chart: one dimension and one fact or two dimensions and one fact.
- Numerical chart: Two or three facts.

For each of the three visualization types we require a data-attribute for the x and y axis and third attribute to show additional dimension.

## 4.2 Visualization Ranking

The goal of this module is to find all good visualizations in a set of possible visualizations. To be able to achieve this, a utility metric responsible for deciding upon good and bad visualization is developed. Firstly, our definition for good visualization is provided.

### 4.2.1 Defining Good Visualization

Before developing the utility metric, first we provide our definition of good visualization. On one side our definition combines the findings made about good visualization in sub-chapter 2.3 and on the other side it follows the results from the tools' analysis.

**Definition 10.** *A good visualization is functional, has no clutter and uses graphical objects understandable by humans.*

In order to be functional, a chart needs to describe the data accurately, e.g. generating meaningful objects so people can interpret its connotation right. As each chart type has a different function, we define the following functions of chart types:

- Line charts show change in time. Relevant for time series data.
- Area charts show part-whole relation over a time period. Relevant for time series data.

- Pie and stacked-bar chart display distribution of a category when part-whole relation exists. Relevant for categorical data.
- Scatter-plot and Bubble chart display positive or negative high correlation coefficient.
- Bar charts (horizontal and vertical) compare categories of a single dimensions.
- Stacked-bars have purpose to show multiple dimensions.
- Grouped-bar represent and compare different categories of two or more groups.

These chart types were chosen based on the Cleveland and McGill's suggestion regarding accurate interpretation of graphical objects [CM84] and Evergreen's recommendation in terms of data load and human perception [Eve17].

Graphical clutter, overusing of special effects or elements bring confusion and disorientation. Good charts draw user attention on the data, therefore chart background, borders, shading, patterns, dark grid lines (also known as chartjunk<sup>1</sup>) are dismissed in our charts. Noise free charts use the least amount of ink to communicate the data, and remove anything that is distracting, causing the data and the graph to stand out.

The last point from our definition concerns the user perception of visualization. Accordingly, we consider the *elementary perceptual task* described in sub-chapter 2.3.1. It presents a hierarchy of graphical objects ordered by most to least accurate for humans to understand. Furthermore, we consider the chart chooser developed by Dr. Abel described in 2.3.3 in which strict rules are defined concerning the user perception for each chart type.

### 4.2.2 Criteria for Ranking Visualizations

The task of the visualization ranking module is to score each detected visualization based on a specific set of criteria. This set includes statistical metrics and follow the

---

<sup>1</sup>The American statistician Edward Tufte defines "chartjunk" as "The interior decoration of graphics generates a lot of ink that does not tell the viewer anything new. The purpose of decoration varies to make the graphic appear more scientific and precise, to enliven the display, to give the designer an opportunity to exercise artistic skills. Regardless of its cause, it is all non-data-ink or redundant data-ink, and it is often chartjunk." [Tuf83]

rules for human perception. We developed a utility metric for assessing visualizations which includes the following ten criteria:

1. **Number of dimensions.** It is a relevant indicator of a chart type. The number of dimensions defines whether a pie, bar or numerical types of charts to favorize. Should a second dimension be charted then pie chart or horizontal bars should be avoided. If there exists no dimension in the chart combination then it is an indication of numerical chart type e.g, scatter or bubble chart,
2. **Number of facts.** This criterion points out to certain subset of chart types the same way as the number of dimension. Combination of data attributes with one fact points out to a larger list of good visualisations, e.g., line, vertical bars, slope, multi-graph or area charts, whereas two or more numerical dimensions recommends numerical chart types,
3. **Number of dimension categories,** indicates how many lines (when line chart to be visualized), pie slices (for pie chart) or bars need to be visualized. This value points out to how "loaded" a chart would be. Slope charts are good when maximum six categories are shown, bigger number results into more cognitive load and generates cluttered chart [Eve17], which by our definition given in Sub-chapter 4.2.1 should be avoided. Pie charts are only good for dimensions with up to four categories, line and area charts with six categories, vertical and horizontal stacked bar - five categories. According to the human perception of grouped items, two or three categories are indicator for generating good grouped bar chart. Multi-graph series on x-axis can accept up to five categories, when charts are positioned in three rows. Horizontal bars are good when dimension has more then one category, otherwise it would generate a single horizontal bar, which is a pointer that other criteria needs to be considered (e.g., number of facts, dimensions or data tuples),
4. **Types of the data attributes.** To ensure presence of the first point from our definition (functionality), different data types require specific chart type. Therefore we define the following rules for this criterion. Numerical charts are shown with scatter, bubble or line charts. Categorical charts are presented with pie, bar, slope multi-graph, horizontal and vertical stacked bar and grouped bar, for distinguishing of the categorical share. Time series charts are shown with area, line, grouped, vertical stacked bar, multi-graph and slope graph, for showing change in time or trend of a certain value.

5. **Number of data tuples**, presents the number of points in the charts. As we have defined good charts as clear and noise-free, it is important to set an upper boundary in terms of data points for each chart. The research and practice have shown that: slope charts can visualize only two data tuples, if more date-time values should be shown then areas and line charts are better, grouped bars are limited to five data points due to the limitation in human perception of grouped bars. A single data point indicates to horizontal bar or pie chart.
6. **Deviation** calculates how spread out the data points in one chart are from the mean. High deviation (more then 20% of the average) indicates big change in category values and is good to be charted as slope or grouped bars. Stacked bars are good when deviation is low, as categories need to start at the same or similar baseline to be easily compared. Area and line charts need to have lower deviation, in order to prevent generation cluttered and chaotic chart.
7. **Correlation Coefficient**, a value between -1 and 1, relevant only for the numerical charts and combination of data attributes having only numerical values. The closer value to -1 or to 1 ( $\leq -0.6$  or  $\geq 0.6$ ) indicates correlation (negative or positive respectively) meaning scatter or bubble charts should be chosen. Coefficient close to 0 means there is no correlation between data attributes, meaning line chart to be selected.
8. **Intersection of two lines** is relevant for time series or categorical data charted with line, slope or area charts. This criteria is important to prevent producing line charts which have many crisscross lines. Examples from practice have shown that the number of intersection points should be equal to the number of unique dimension values (categories). To prevent the clutter in one chart a multi-graph chart should be visualized.
9. **Part-whole relation** is relevant for pie or vertical stacked chart types. Pie charts are famous for showing data which is whole but distributed among many categories. The decision between pie and vertical stacked bar is made based on other criteria (e.g., number of categories)
10. **Null values**, or missing values for a certain fact or dimension is relevant for line area, stacked, grouped charts. Practices have shown that trend between two or more lines can be easily compared when both lines have same length, stacked bars have similar baseline and the categories are shown with bars.

### 4.2.3 Ranking Principle

For all combinations of data attributes from *Visualization recognition* module, a utility score is calculated for each chart types. As different visualization types are suitable for different purposes or data relations, each criteria has different importance for certain chart type. Therefore, individual weights have been assigned for each criterion depending on the chart type. The weights are given in Table 4.1

Visualization type	Weight - 10	Weight - 5
Pie	<ul style="list-style-type: none"> <li>• Number of dimensions,</li> <li>• Number of data tuples,</li> <li>• Number of dimension categories,</li> <li>• Part-whole relation,</li> <li>• Type of the data attributes</li> </ul>	<ul style="list-style-type: none"> <li>• Number of facts,</li> <li>• Intersection of two lines,</li> <li>• Deviation,</li> <li>• Correlation,</li> <li>• Null values</li> </ul>
Bar	<ul style="list-style-type: none"> <li>• Number of dimensions,</li> <li>• Number of dimension categories,</li> <li>• Number of data tuples,</li> <li>• Part-whole relation</li> </ul>	<ul style="list-style-type: none"> <li>• Number of facts,</li> <li>• Type of data attributes,</li> <li>• Deviation,</li> <li>• Correlation,</li> <li>• Intersection of two lines,</li> <li>• Null values</li> </ul>
Slope	<ul style="list-style-type: none"> <li>• Number of dimensions,</li> <li>• Number of dimension categories,</li> <li>• Number of data tuples,</li> <li>• Intersection of two lines,</li> <li>• Null values</li> </ul>	<ul style="list-style-type: none"> <li>• Number of facts,</li> <li>• Type of data attributes,</li> <li>• Deviation,</li> <li>• Correlation,</li> <li>• Part-whole relation</li> </ul>

4 Utility Metric for Generating Good Visualizations

Visualization type	Weight - 10	Weight - 5
Multi-graph series	<ul style="list-style-type: none"> <li>• Number of dimension categories,</li> <li>• Type of data attributes,</li> <li>• Number of data tuples,</li> <li>• Intersection of two lines,</li> <li>• Null values</li> </ul>	<ul style="list-style-type: none"> <li>• Number of dimensions,</li> <li>• Number of facts,</li> <li>• Deviation,</li> <li>• Correlation,</li> <li>• Part-whole relation</li> </ul>
Horizontal stacked-bar	<ul style="list-style-type: none"> <li>• Number of dimension categories,</li> <li>• Type of the data attributes,</li> <li>• Part-whole relation,</li> <li>• Null values</li> </ul>	<ul style="list-style-type: none"> <li>• Number of dimensions,</li> <li>• Number of facts,</li> <li>• Number of data tuples,</li> <li>• Deviation,</li> <li>• Correlation,</li> <li>• Intersection of two lines</li> </ul>
Vertical stacked-bar	<ul style="list-style-type: none"> <li>• Number of dimension categories,</li> <li>• Deviation,</li> <li>• Intersection of two lines,</li> <li>• Null values</li> </ul>	<ul style="list-style-type: none"> <li>• Number of dimensions,</li> <li>• Number of facts,</li> <li>• Type of the data attributes,</li> <li>• Number of data tuples,</li> <li>• Correlation,</li> <li>• Part-whole relation</li> </ul>
Grouped-bar	<ul style="list-style-type: none"> <li>• Number of dimension categories,</li> <li>• Number of data tuples,</li> <li>• Part-whole relation,</li> <li>• Null values</li> </ul>	<ul style="list-style-type: none"> <li>• Number of dimensions,</li> <li>• Number of facts,</li> <li>• Type of the data attributes,</li> <li>• Deviation,</li> <li>• Correlation,</li> <li>• Intersection of two lines</li> </ul>



4 Utility Metric for Generating Good Visualizations

Visualization type	Weight - 10	Weight - 5
Line	<ul style="list-style-type: none"> <li>• Number of dimension categories,</li> <li>• Type of the data attributes,</li> <li>• Number of data tuples,</li> <li>• Intersection of two lines,</li> <li>• Null values</li> </ul>	<ul style="list-style-type: none"> <li>• Number of dimensions,</li> <li>• Number of facts,</li> <li>• Deviation,</li> <li>• Correlation,</li> <li>• Part-whole relation,</li> </ul>
Area	<ul style="list-style-type: none"> <li>• Number of dimension categories,</li> <li>• Type of the data attributes,</li> <li>• Number of data tuples,</li> <li>• Intersection of two lines,</li> <li>• Null values</li> </ul>	<ul style="list-style-type: none"> <li>• Number of dimensions,</li> <li>• Number of facts,</li> <li>• Deviation,</li> <li>• Correlation,</li> <li>• Part-whole relation,</li> </ul>
Scatter	<ul style="list-style-type: none"> <li>• Number of facts,</li> <li>• Type of the data attributes,</li> <li>• Number of data tuples,</li> <li>• Correlation</li> </ul>	<ul style="list-style-type: none"> <li>• Number of dimensions,</li> <li>• Number of dimension categories,</li> <li>• Deviation,</li> <li>• Intersection of two lines,</li> <li>• Part-whole relation,</li> <li>• Null values</li> </ul>
Bubble	<ul style="list-style-type: none"> <li>• Number of facts,</li> <li>• Type of the data attributes,</li> <li>• Number of data tuples,</li> <li>• Correlation,</li> <li>• Null values</li> </ul>	<ul style="list-style-type: none"> <li>• Number of dimensions,</li> <li>• Number of dimension categories,</li> <li>• Deviation,</li> <li>• Intersection of two lines,</li> <li>• Part-whole relation</li> </ul>

Visualization type	Weight - 10	Weight - 5
--------------------	-------------	------------

Table 4.1: Weights of each criterion per visualization type

We calculate a utility score for each visualization type with the criteria weights defined for this particular type by using the weighted formula given in (4.1),

$$W_{B_i} = \sum_{1 \leq i \leq 10} Criteria_{iChartType_j}, \quad (4.1)$$

where *ChartType* is defined as:

*ChartType*={*pie, bar, slope, multi-graph, horizontal stacked-bar, vertical stacked-bar, grouped-bar, line, area scatter, bubble*}

and *Criteria* is the set of all 10 criteria from our utility metric. Its value depends on the visualization type for which we calculate the score. It is defined as:

*Criteria*={ '*number\_dimensions*', '*number\_facts*', '*number\_categories*', '*data\_type*', '*number\_tuples*', '*deviation*', '*correlation*', '*intersection*', '*part-whole*', '*nulls*' }

From Table 4.1, we can see that even though all chart types have different number of high weighted criteria, some of the chart types have similar criteria for generating good charts. Bar chart, horizontal and vertical stacked-bar, grouped-bar and scatter chart have all a list of 4 high weighted criteria, thus giving score of 40 when these criteria are met. Additionally the number of lower weighted criteria for these charts is 6, thus giving score of 30 when met. In total for these charts, a score of 70 points is minimum when all criteria are met. Therefore we define:

**Definition 11.** *All visualizations having a utility score of more or equal 70 are good.*

Only those charts with this score tend to satisfy all three aspects from our definition (functionality, clutterness and graphical perception).

In the next Chapter we describe the implementation of the utility metric in a web-based application.

## 5 Implementation

In order to provide automation for generating good charts we have developed TAG<sup>2</sup>S<sup>2</sup> - a Tool for Automatic Generation of Good viSualization using Scoring. It presents a web-based solution which fully automatically creates or recommends good visualizations. It is build on an existing reporting tool - *reports2go*, which provides users with access to visualizations from an uploaded data-set. *reports2go* has an intuitive and easy-to-navigate interface which allows users to examine the data, get insights or support explanation. It works without server thus making it possible the creation of visualizations even when there is no connection to the internet. In Figure 5.1 a screenshot of the initial view of *reports2go* is given, where user can upload data-set and follow the next steps.

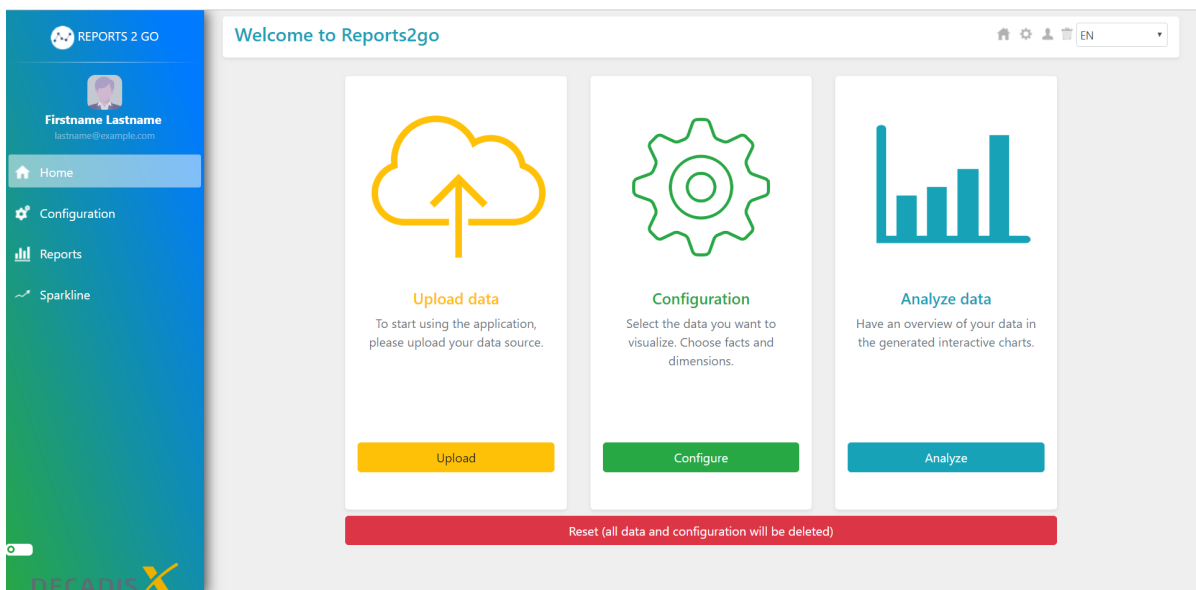


Figure 5.1: reports2go upload data-set view

In order to present the tool and to draw a line between functionality of TAG<sup>2</sup>S<sup>2</sup> and *reports2go* we show a use case. For this purpose we click on *Upload* and a pop-up view

## 5 Implementation

opens as given in Figure 5.2. We upload a json file by choosing *Upload data* and selecting the correct data format. A snippet of the uploaded data-set is given in Appendix A.

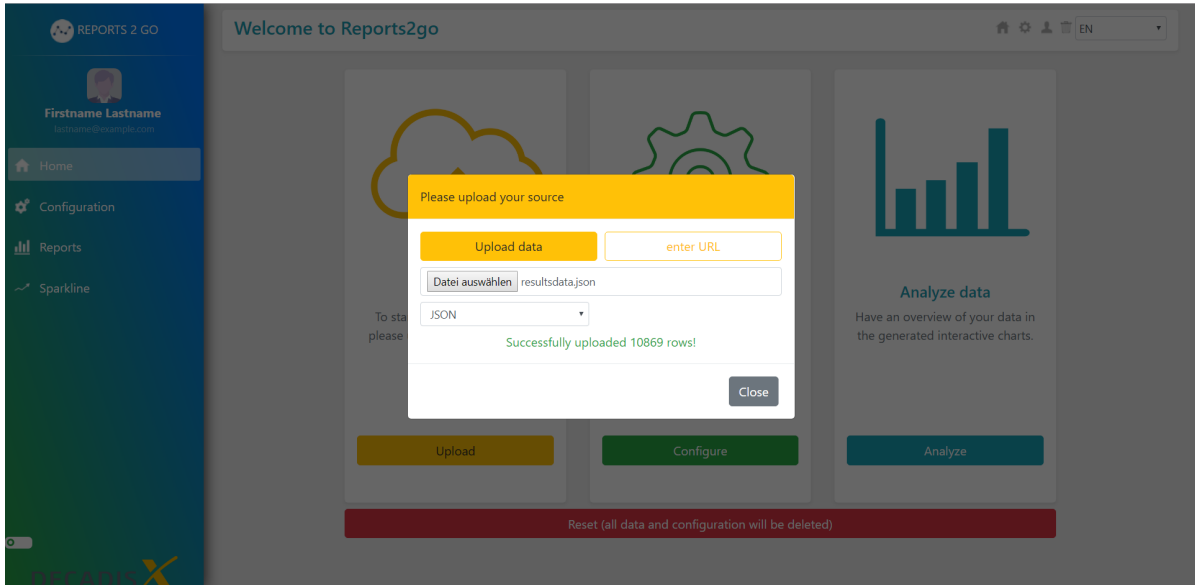


Figure 5.2: Uploading data-set in reports2go

Next step is the configuration. We select *Configure* in which a new view is opened as given in Figure 5.3. The configuration options include: defining data attributes as time dimensions, dimensions and kpis, adding additional dimension columns or calculated kpis (calculated by selecting two columns and a certain mathematical operation). Here, we have the possibility to select whether we want to have time series or categorical visualizations. After the mandatory selection of data-attribute as value for x-axis, we are ready to configure the other columns. However, the user is not obligated to continue the configuration. The tool has already marked data-attributes as dimensions or kpis by analyzing its values.

## 5 Implementation

The screenshot shows the 'Object configuration' page in the Reports 2 Go application. The interface includes a sidebar on the left with navigation options: Home, Configuration, Reports, and Sparkline. The main content area is titled 'Object configuration' and features a header with 'Upload json configuration!' and 'Select from saved configurations!'. Below this, there are radio buttons for 'Time series' (selected) and 'Categorical'. A date selection dropdown is set to 'Date', and a format input field shows '%d.%m.%Y'. An 'Export config' button is visible. A 'Configure input' button and a 'Go further' button are also present. The main part of the page displays a table with the following data:

Date	BL	Size	kw	Weekday	KPI_2	KPI_3	Customer_Group	Product
02.04.2019	BW	S	2019KW14	Di	0.033311126	0.023405439	Group 4	Product 1
13.04.2019	HE	Sonst	2019KW15	Sa	0.00708115	0.005077849	Group 9	Product 3
11.04.2019	RP	M	2019KW15	Do	0.019516003	0.031047491	Group 9	Product 1
01.04.2019	RP	Sonst	2019KW14	Mo	0.01556178	0.048042719	Group 1	Product 1
09.04.2019	MV	Sonst	2019KW15	Di	0.020328155	0.035664909	Group 5	Product 3
01.04.2019	BB	S	2019KW14	Mo	0.104166667	0.10220522	Group 7	Product 2
05.04.2019	SN	M	2019KW14	Fr	0	0	Group 3	Product 2
05.04.2019	SN	Sonst	2019KW14	Fr	0.017059606	0.040023078	Group 9	Product 3
04.04.2019	HE	Sonst	2019KW14	Do	0	0	Group 3	Product 3
11.04.2019	NI	Sonst	2019KW15	Do	0.005581292	0.035752103	Group 2	Product 3

Figure 5.3: A configuration page of reports2go

We select *Time series* and choose the column *Date* as x-axis. With the help of the date-time pop-up window we define the appropriate date format for the selected time dimension. In this case it is *%d.%m.%Y* and configure further the data. Now the user is able to select/unselect certain columns. The purpose of this step is to mark the column as data-attribute or to select/unselect columns he wants to see in the visualizations. Our configuration is given in Figure 5.4. This view as well allows naming and storing different configurations for a single data-set. All stored configurations are ready for export as a csv file but also are accessible after the user has closed the browser. The exported files can later be uploaded to omit the configuration step of the same data.

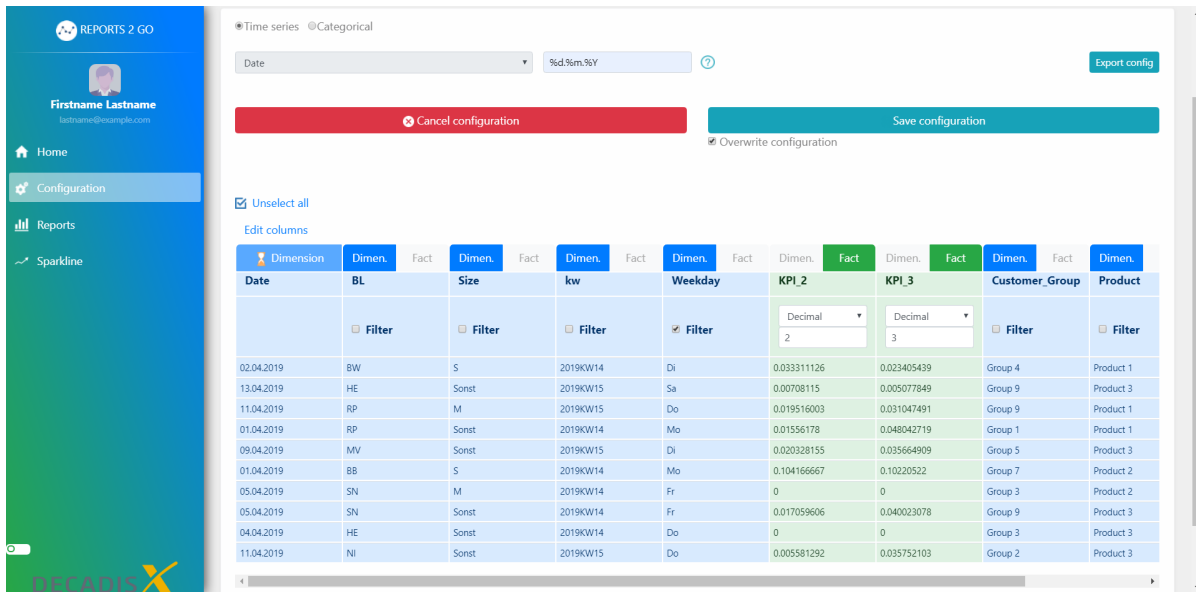


Figure 5.4: A configuration of data-attributes of reports2go

After saving the configuration and navigating to *Reports*, the user is able to see the configured visualizations. Each visualization is grouped together with a table view and a set of options, displayed as separate blocks. For the configuration made *reports2go* generated five visualizations given in Appendix B. All of them are stacked-bars and show values of the first kpi from the configuration array (in this case KPI.1). For each chart, the user can change the chart type, add a filter or switch to another kpi. Additionally a running average can be added to the chart.

In a case when the user has not performed the configuration step and did not select the type of charts and column for x-axis, in this view, the tool shows a warning message and no visualization is displayed on the screen.

### TAG<sup>2</sup>S<sup>2</sup>'s contribution

Our tool TAG<sup>2</sup>S<sup>2</sup> updates the algorithm in the visualization view, mainly the selection of a chart type. As in the current implementation of *reports2go*, for all generated charts the same chart type is selected, our contribution improves the process of chart selection by employing the utility metric described in sub-chapter 4.2.2. It is done in two ways:

- By getting the user configuration. The goal is to get the configuration made by the user and provide only good visualizations. For the current configuration and data-set,

TAG<sup>2</sup>S<sup>2</sup> has generated three visualizations given in Appendix C.

- By automatically generating configuration. When no configuration is made by the user, instead of a warning message, the tool generates a set of good visualizations. TAG<sup>2</sup>S<sup>2</sup> creates configuration automatically by looping through the table columns and marking each based on its values throughout the whole data-set. When a column is marked as a time dimension, its date-time format is automatically detected and stored. The results produced by our tool when no user configuration is provided are given in Chapter 6.

## 5.1 Architecture and Technology

In this sub-chapter we describe the architecture as given in Figure 5.5 and the technology we have used to develop TAG<sup>2</sup>S<sup>2</sup>.

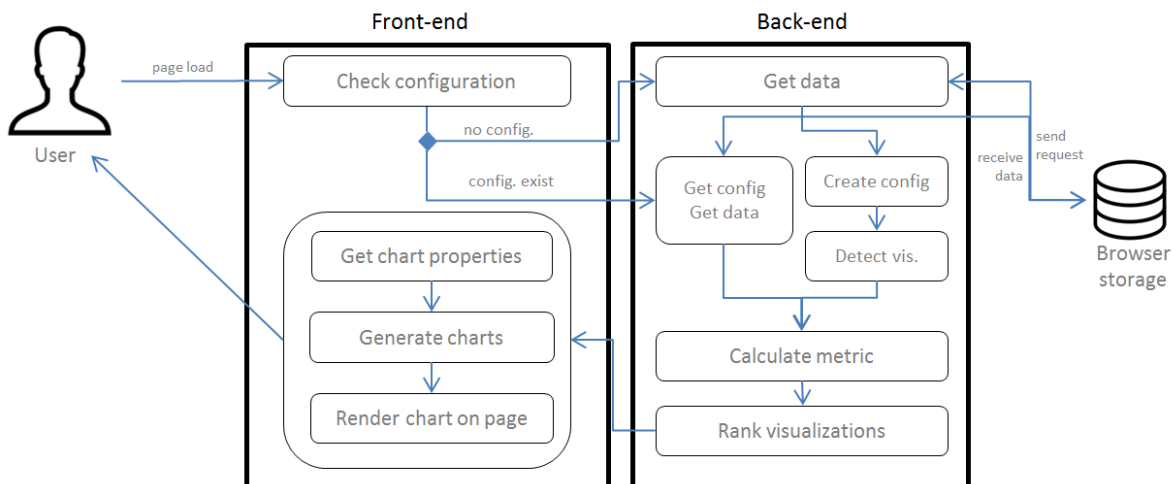


Figure 5.5: Architecture of TAG<sup>2</sup>S<sup>2</sup>

The TAG<sup>2</sup>S<sup>2</sup>'s front-end is responsible for generating good visualizations and displaying them on the screen. For this purpose we use two free-source charting libraries: C3.js<sup>1</sup> and D3.js<sup>2</sup> due to the big possibilities for customization of the charts and formatting after chart rendering on the page. We store a JSON object for styling properties for all charts. Later we pass the properties of a specific chart to the charting library.

<sup>1</sup><https://c3js.org/>

<sup>2</sup><https://d3js.org/>

The back-end is responsible for automated tasks. As it is a more complex part we have divided it in three components: *detect data attributes*, *creation of visualizations* and *metric calculation*. For the first and second component we use standard functions from JavaScript 6.0 for detecting numerical or textual values and external libraries (e.g., *moment.js*<sup>3</sup>) for detecting date-time formats. In order to calculate the utility of a chart in the last component we use Node.js libraries such as *line-intersection*<sup>4</sup> and *stats-lite*<sup>5</sup> for calculating statistical operations (intersection between two lines, deviation and correlation). More details about the libraries and how they are mapped with the back-end functions are provided in the sub-chapter 5.3.

## 5.2 Front-end

The front-end is responsible for generating and rendering charts on the page. It makes sure that the charting libraries are called with the right data and chart-parameters in order to generate the good visualizations and display them on the screen.

It consists of:

- Reading configuration and calling proper back-end function.
- Reading chart properties for a single type (e.g., number of x-axis values, axis names, show/display legend, position of legend)
- Call to *c3.js* function to generate a chart.

The function given in Algorithm 1 loops over the array of objects (good charts) received from the back-end and creates valid JSON objects as required by *c3.js*. These JSON objects already consist of data parameters (calculated and prepared by the back-end) and we append respective styling properties (loaded from an external file). Each object comes with the following data parameters:

- Id of the chart
- Name of the x-axis dimension
- Array of unique categories (for time series or categorical data)
- The format of the x-axis dimension (if time series chart to be charted)
- Name of the kpi

---

<sup>3</sup><https://momentjs.com/>

<sup>4</sup><https://www.npmjs.com/package/line-intersection>

<sup>5</sup><https://www.npmjs.com/package/stats-lite>



- Data
- Type of data (categorical, numerical or time series)
- Chart type

Within the loop for each object (chart) we call two functions: for getting the chart properties and generating the chart.

---

**Algorithm 1** prepareAndGenerateChart
 

---

**Require:** **Input:** chartsObject []

- 1: **for**  $i \leftarrow \text{range}(0 - \text{len}(\text{chartsObject}))$  **do**
  - 2:  $\text{prop} \leftarrow \text{getChartProperties}(\text{chartsObject}[i].\text{dataType}, \text{chartsObject}[i].\text{chartType})$
  - 3:  $\text{generateChart}(\text{chartsObject}[i], \text{prop})$
  - 4: **end for**
- 

The first function being called in the loop within Algorithm 1 is to get the properties of a chart. Therefore, we have created a JSON object where for each chart type we have defined properties based on the utility metric for good visualizations. Algorithm 2 includes this properties' file and based on the arguments (chart type) returns the requested properties of chart. The content of this JSON chart properties object is given in Appendix D.

---

**Algorithm 2** getChartProperties
 

---

**Require:** chartProperties.json

**Input:** dataType, chartType **Output:** properties

- 1: return  $\text{chartProperties}.[\text{dataType}][\text{chartType}]$
- 

The second function called by the front-end accepts an object with data and chart properties. Algorithm 3 represents a pseudo code for this function responsible for generating a chart and rendering it on the page. It requires the libraries c3.js and d3.js. The goal of this algorithm is to map the parameters of the object received as an input to a keys which are known for the libraries when calling their api for chart generation. In this object the data parameters are stored from the back-end and the chart-properties. At the beginning we map the respective chart to a html container with an id='chartN' where N is the id of the chart. Then we map the data-parameters: the data itself, the keys for x-axis and y axis, order in which the data to be shown. After this, we give the chart-properties: visibility of data-points, number of x-axis values, number of ticks on

the x-axis, name for the axis, grid lines visibility and legend visibility and position. At the end we return this chart and store it in an array for manipulating it later.

---

**Algorithm 3** generateCharts
 

---

**Require:** c3.js, d3.js

**Input:** chartObj **Output:** chart

```

1: Obj ← {},
2: Obj.bindto ← "#chart -" + chartObj.chartId
3: Obj.data ← {json : chartObj.data,
4:   keys : {
5:     x : chartObj.xaxis,
6:     value : chartObj.dataKeys}
7:   order : chartObj.dataOrder}
8: Obj.point ← chartObj.showDataPoint
9: Obj.axis ← {
10:  x : {
11:    type : chartObj.dataType,
12:    tick : {
13:      format : chartObj.xaxisFormat,
14:      culling : {max : chartObj.xCulling},
15:      count : chartObj.xTickCount}
16:    label : chartObj.xAxisLabel}
17:  y : {
18:    show : chartObj.showY Axis,
19:    tick : {format : chartObj.yaxisFormat},
20:    label : chartObj.yAxisLabel}
21:  rotated : chartObj.axesRotate}
22: Obj.grid ← {
23:  x : {show : chartObj.xGridShow,
24:    lines : chartObj.optionalGridLines}
25:  y : {show : chartObj.yGridShow,
26:    lines : chartObj.optionalYGridLines}}
27: Obj.legend ← {show : chartObj.showLegend, position : chartObj.legendPosition}
28: chart ← c3.generate(Obj)

```

---

### 5.3 Back-end

The TAG<sup>2</sup>S<sup>2</sup>'s back-end is more complex and is separated in three algorithms performing different tasks.

The first part *detect data attributes* is responsible for deciding upon dimensions and kpis. The Algorithm 4 shows the implementation for this part. It accepts the data-set uploaded by the user and loops over each row until all possible keys are placed in one of the three arrays: time dimension, dimension or kpis. For this purpose, we use the free-source library `moment.js` to assess if a string is a date and if so, to detect its format. For the numerical values we use respective JavaScript function.

---

**Algorithm 4** Detect data attributes

---

**Require:** `moment.js`

**Input:** `dataSet`

```

1: for  $i \leftarrow \text{range}(0 - \text{len}(\text{dataSet}))$  do
2:   if moment(i).isValid() then
3:      $\text{timeDimensions} \leftarrow i$ 
4:   else if !(isNaN(i)) then
5:      $\text{kpis} \leftarrow i$ 
6:   else
7:      $\text{dimensions} \leftarrow i$ 
8:   end if
9: end for

```

---

The next component *creation of visualizations* combines the detected data attributes so each possible combination is included. This function loops over the populated arrays generated from Algorithm 4 and makes sure that all combinations are created. The output is an array of objects (visualizations) where we store the data attributes. Algorithm 5 provides this process. At the beginning, we create all possible time series charts, then categorical and at the end we loop over all kpis in order to create numerical charts for evaluations. The output array is then passed to the last part of the back-end.

---

**Algorithm 5** Creation of visualization

---

**Input:** timeDimensions[], dimensions[], kpis[]  
**Output** visualizations []

```

1: for  $i \leftarrow \text{range}(0 - \text{len}(\text{timeDimensions}))$  do
2:   for  $j \leftarrow \text{range}(0 - \text{len}(\text{dimensions}))$  do
3:     for  $k \leftarrow \text{range}(0 - \text{len}(\text{kpis}))$  do
4:        $\text{visualizations} \leftarrow \text{timeDimensions}[i], \text{dimensions}[j], \text{kpis}[k]$ 
5:     end for
6:   end for
7: end for
8: for  $i \leftarrow \text{range}(0 - \text{len}(\text{dimensions}))$  do
9:   for  $j \leftarrow \text{range}(0 - \text{len}(\text{dimensions}))$  do
10:    for  $k \leftarrow \text{range}(0 - \text{len}(\text{kpis}))$  do
11:       $\text{visualizations} \leftarrow \text{dimensions}[i], \text{dimensions}[j], \text{kpis}[k]$ 
12:    end for
13:  end for
14: end for
15: for  $i \leftarrow \text{range}(0 - \text{len}(\text{kpis}))$  do
16:   for  $j \leftarrow \text{range}(0 - \text{len}(\text{kpis}))$  do
17:    for  $k \leftarrow \text{range}(0 - \text{len}(\text{kpis}))$  do
18:       $\text{visualizations} \leftarrow \text{kpis}[i], \text{kpis}[j], \text{kpis}[k]$ 
19:    end for
20:  end for
21: end for

```

---

The last component (*metric calculation*), calculates a utility score for each visualization. The array given as an argument contains the necessary data properties for calculation of the metric (data, x and y axis and their unique values). The Algorithm 6 gives an overview of calculation a utility score for pie chart. For better readability, we use hard coded values to check the values of the object.

---

**Algorithm 6** Calculation of utility metric

---

**Input:** visualizations []    **Output:** scores []

```

1: for  $i \leftarrow \text{range}(0 - \text{len}(\text{visualizations}))$  do
2:   if  $\text{visualizations}[i].\text{numberDimensions} == 1$  then
3:      $\text{scores}['\text{pie}'] \leftarrow 10$ 
4:   end if
5:   if  $\text{len}(\text{visualizations}[i].\text{data}) == 1$  then
6:      $\text{scores}['\text{pie}'] \leftarrow 10$ 
7:   end if
8:   if  $\text{len}(\text{visualizations}[i].\text{categories}) \leq 4$  then
9:      $\text{scores}['\text{pie}'] \leftarrow 10$ 
10:  end if
11:  if  $\text{partWhole}(\text{data}, \text{xaxis}, \text{kpi}) == \text{true}$  then
12:     $\text{scores}['\text{pie}'] \leftarrow 10$ 
13:  end if
14:  if  $\text{visualizations}[i].\text{dataType} == \text{categories}$  then
15:     $\text{scores}['\text{pie}'] \leftarrow 10$ 
16:  end if
17:  if  $\text{visualizations}[i].\text{intersectionPoints} == 0$  then
18:     $\text{scores}['\text{pie}'] \leftarrow 5$ 
19:  end if
20:  if  $\text{visualizations}[i].\text{deviation} == \text{null}$  then
21:     $\text{scores}['\text{pie}'] \leftarrow 5$ 
22:  end if
23:  if  $\text{visualizations}[i].\text{nulls}$  then
24:     $\text{scores}['\text{pie}'] \leftarrow 5$ 
25:  end if
26:  if  $\text{visualizations}[i].\text{correlation} == 0$  then
27:     $\text{scores}['\text{pie}'] \leftarrow 5$ 
28:  end if
29:  if  $\text{len}(\text{visualizations}[i].\text{facts}) == 1$  then
30:     $\text{scores}['\text{pie}'] \leftarrow 5$ 
31:  end if
32: end for

```

---

## 5 Implementation

As our utility metric described in Chapter 4 has ten criteria, for each criterion in Algorithm 6 we calculate a score. For a single chart to receive this score, it needs to have the predefined value for each criteria.

Following we describe how we calculate the values for all criteria of the utility metric.

*Number of dimensions* is calculated by checking if a certain dimension is part of the time dimension array or dimensions of the data object. We perform this for each key given in the visualization object and count the dimensions.

*The data length* is simple calculated by using the JavaScript *length()* function on the data. It gives the number of rows(object in one array).

*Number of categories* is determined by counting the unique values of the x-axis keys in the data. Algorithm 7 receives the data as an input together with the name of the x-axis for which we want to find the unique values. It returns an array of the unique values for that key. We simply count the values in the array to get its number.

---

**Algorithm 7** Number of categories

---

**Input:** data [], xaxis    **Output:** uniqueVals

```
1: uniqueVals ← []
2: for i ← range(0 – len(data)) do
3:   if !uniqueVals.include(data[i][xaxis]) then
4:     uniqueVals.push(data[i][xaxis])
5:   end if
6: end for
```

---

For calculating *part-whole relation* in a data-set we take a sum of 100 to be an indicator for such a relationship. First we group the data by values of x-axis and then for each value of x-axis we sum its value. Algorithm 8 responsible for detecting this relation accepts two arguments: the data and x-axis name for which we sum the values. In order to get the unique values of the x-axis dimension we call the function from Algorithm 7. Then we loop over each key in that array and sum their values. If their sum is differs from 100 we stop the process and return false.

---

**Algorithm 8** Part-whole relation

---

**Input:** data [], xaxis    **Output:** hasPartWhole

```

1: categories ← getCategory(data, xaxis)
2: sum ← 0
3: hasPartWhole ← false
4: for i ← range(0 – len(data)) do
5:   sum ← 0
6:   for j ← range(0 – len(categories)) do
7:     sum+ = data[i][categories[j]]
8:   end for
9:   if sum! = 100 then
10:    return hasPartWhole
11:  end if
12: end for
13: hasPartWhole ← true
14: return hasPartWhole

```

---

The *data type* can have one of the three values: *categories*, *time series* or *numerical*. When the *dimension* array in one chart object has two values we say it is categorical chart, when *kpis* array contains two values then it is numerical chart. A single value in the *timeDimension* array in the chart object indicates time series chart.

The *intersection* points are calculated only for time series charts having one dimension and one fact. We do not calculate the intersection points for other data-types because this criteria is related with lines charts which we relate with time series data. It is done by using the *lines-intersection*<sup>6</sup> Node.js library. For calculating intersection of two lines, the function takes an array of coordinates of four points. In order to find the coordinate of a certain point in the chart, the algorithm takes the index of an element in the array of dimensions or data. For example, for the data given in Appendix F, the following values from the algorithm are calculated:

```

data=[{Year: 2008, female: 80, male: 420}
      {Year: 2009, female: 70, male: 430}
      {Year: 2010, female: 60, male: 440}
      {Year: 2011, female: 90, male: 410}

```

---

<sup>6</sup><https://www.npmjs.com/package/lines-intersection>

```
{Year: 2012, female: 75, male: 425}
{Year: 2013, female: 68, male: 432}
{Year: 2014, female: 85, male: 415}]
```

```
dimension = 'Gender'
timeDimension = 'Year'
categories=['female', 'male']
xAxisValues=[2008, 2009, 2010, 2011, 2012, 2013, 2014]
yAxisValues=[0, 1, 2, 3, 4 ... 441]
```

The line chart for this data is given in Figure 5.6. Our algorithm goes over each line of a single category and calculates its intersection with any other line on the chart.

Year - Gender - Number

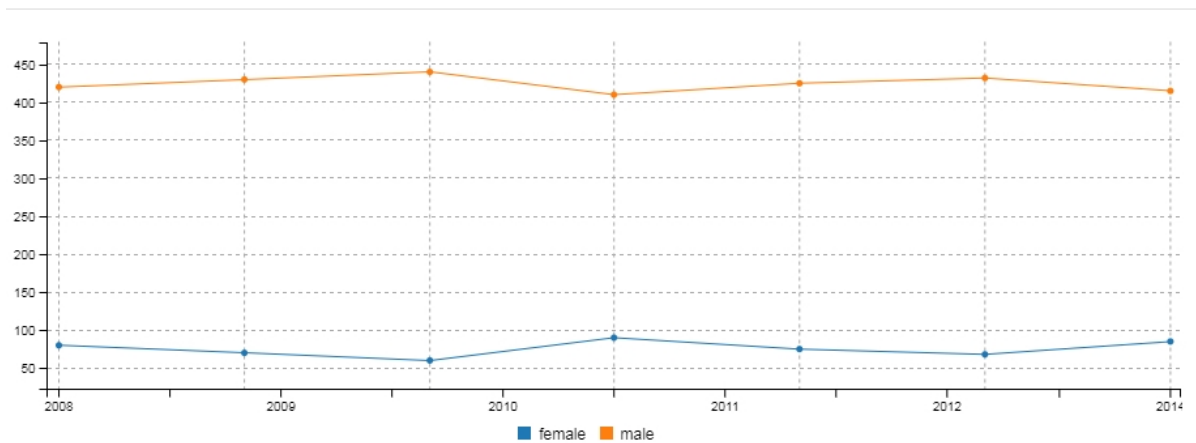


Figure 5.6: Example of line chart for calculating intersection points

The example of storing the coordinates and the principle of intersection calculation is given in Table 5.1. In order to get the x coordinates for the first line for category *Female*, we take the indexes of the values 2008 and 2009 in the array *xAxisValues*. Then we add +1 to that number to avoid the first position which has index = 0. Therefore our x coordinates are 1 and 2.

To get the y coordinates we look in the *data* array for the values for *Female* when *Year* = 2008 and 2009 and ask for their indexes in *yAxisValues* array. Our y coordinates are 80 and 70. We perform this step for the first line of the category *Male* and get the following coordinates  $(1, 420)(2, 430)$ . We pass these values to the *line-intersection* library for calculation. This process is repeated until we reach the last point in the chart.



## 5 Implementation

Table 5.1: Overview of a calculating intersection for two lines

Iteration	Line 1	Line 2	Intersection
1	(1, 80) (2,70)	(1, 420) (2, 430)	(-17, 250)
2	(1, 80) (2, 70)	(2, 430) (3, 440)	(-17, 250)
3	(1, 80) (2, 70)	(3, 440) (4, 410)	(21, -130)
4	(1, 80) (2, 70)	(4, 410) (5, 425)	(-11.4, 194)
.	...	...	...
n	(6, 68) (7, 85)	(6, 432) (7, 415)	(15.70, 250)

We say that there is intersection when the result coordinates (e.g. (-17, 250) for the first iteration) are between the coordinates of the two lines for which we calculate the intersection, in this case (1, 80) (2,70) and (1, 420) (2, 430). For this iteration we do not have an intersection as the x coordinate of the intersection point is not between 1 - 2.

---

**Algorithm 9** Line intersection

---

**Require:** lines-intersection

**Input:** data [], xaxis categories **Output:** intr []

```

1:  $xAxisValue \leftarrow data.map(f \Rightarrow f(xaxis))$ 
2:  $yAxisValue \leftarrow [0 - max(data)]$ 
3: for  $i \leftarrow range(0 - len(categories))$  do
4:    $cat1Data \leftarrow data.filter(f \Rightarrow f[categories[i]])$ 
5:   for  $d1 \leftarrow range(0 - len(cat1Data))$  do
6:      $startLine1coordX \leftarrow xAxisValue.indexOf(cat1Data[d1][xaxis])$ 
7:      $startLine1coordY \leftarrow yAxisValue.indexOf(cat1Data[d1][categories[i]])$ 
8:      $endLine1coordX \leftarrow xAxisValue.indexOf(cat1Data[d1 + 1][xaxis])$ 
9:      $endLine1coordY \leftarrow yAxisValue.indexOf(cat1Data[d1 + 1][categories[i]])$ 
10:    for  $j \leftarrow range(0 - len(categories))$  do
11:       $cat2Data \leftarrow data.filter(f \Rightarrow f[categories[j]])$ 
12:      for  $d2 \leftarrow range(0 - len(cat2Data))$  do
13:         $startLine2coordX \leftarrow xAxisValue.indexOf(cat2Data[d2][xaxis])$ 
14:         $startLine2coordY \leftarrow yAxisValue.indexOf(cat2Data[d2][categories[j]])$ 
15:         $endLine2coordX \leftarrow xAxisValue.indexOf(cat2Data[d2 + 1][xaxis])$ 
16:         $endLine2coordY \leftarrow yAxisValue.indexOf(cat2Data[d2 + 1][categories[j]])$ 
17:         $intr \leftarrow intersection($ 
18:           $\{x : startLine1coordX, y : startLine1coordY\},$ 
19:           $\{x : endLine1coordX, y : endLine1coordY\},$ 
20:           $\{x : startLine2coordX, y : startLine2coordY\},$ 
21:           $\{x : endLine2coordX, y : endLine2coordY\})$ 
22:        end for
23:      end for
24:    end for
25:  end for

```

---

*Standard deviation* is calculated for both categorical and time series chart. Therefore we include the library *stats-lite*<sup>7</sup>. This library besides calculating the deviation, provides as well other statistics like mean, median, mode, variance. Our Algorithm 10 calls this library with array of values for a single category. After receiving the value we check if

---

<sup>7</sup><https://www.npmjs.com/package/stats-lite>

the deviation is high. We define high deviation when it is bigger then twenty percent of the mean.

---

**Algorithm 10** Standard deviation
 

---

**Require:** stats-lite

**Input:** data [], categories

```

1: highDeviation ← false
2: for i ← range(0 – len(categories)) do
3:   values ← data.map(f => f[categories[i]])
4:   stdDev ← stats – lite.stdev(vals)
5:   if stdDev > stats – lite.mean(vals) then
6:     highDeviation ← true
7:   end if
8: end for

```

---

For checking *null values*, we simply go over each row in the data and ask if one of the categories has empty value or is not present in the data. Then we return true/false. Relevant for time series and categorical chart types.

The *correlation coefficient* is calculated only for the numerical charts. This metric tells us the strength of the relationship between two sets of values. We calculate by calling *correlation-coefficient-r*<sup>8</sup>. First we need to extract the values of each category in an array and call the respective function form the library with two parameters. Correlation between -0.6 and 0.6 indicates lower correlation whereas values bigger then 0.6 or smaller then -0.6 indicate positive or negative correlation.

---

<sup>8</sup><https://www.npmjs.com/package/correlation-coefficient-r>

---

**Algorithm 11** Correlation coefficient

---

**Require:** correlation-coefficient-r**Input:** data [], categories

```

1: highCorr ← false
2: values1 ← data.map(f => f[categories[0]])
3: values2 ← data.map(f => f[categories[1]])
4: corrCoeF ← correlation(values1, values2)
5: if corrCoeF > 0.6 || corrCoeF < -0.6 then
6:   highCorr ← true
7: end if

```

---

The last criteria in our utility metric is the *number of kpis*. It is an indicator for choosing numerical charts when more than two kpi names are stored in the kpi array of the chart object. Having one value points out to categorical or time series charts.

## 6 Results

In this Chapter, we present the results of the utility metric for good charts. First we present good charts generated from an artificially manufactured data-set. Then we take a data-set from a real-world. In both cases we provide as well charts that received scores less than 70 and were detected as bad visualizations. Finally we develop test cases to evaluate the metric and provide the results.

### 6.1 Synthetic Data-set

The data-set *Country Sales* is about e-commerce, having information regarding products' costs and profits made by selling them across countries and regions. A snippet of the data is given in Appendix E<sup>1</sup>. Some of the values in this data-set were manipulated to show different cases of the tool and the utility metric.

The following columns were recognized as data attributes:

- Time dimensions:

*Order date* - format: DD.MM.YYYY,

*Ship date* - format: DD.MM.YYYY,

- Dimensions: *Regions, Country, Item type, Sales channel, Order priority*
- Kpis: *Units sold, Unit price, Unit cost, Total revenue, Total cost, Total profit*

---

<sup>1</sup><http://eforexcel.com/wp/downloads-18-sample-csv-files-data-sets-for-testing-sales/>

## 6 Results

By combining the detected data-attributes were detected 240 chart objects, each containing the names of x and y axis, their unique values and the data. For each object we calculate 11 utility scores for each chart type, thus giving us at the end 2640 visualizations.

The utility metric has recommended 70 good charts:

- 30 bar charts,
- 12 pie charts,
- 6 grouped charts
- 6 slope charts
- 8 scatter-plots
- 8 bubble charts

Table 6.1 provides an overview of these charts by giving their scores and description of the characteristics.

<b>Figure</b>	<b>Chart type</b>	<b>Score</b>	<b>Characteristics</b>
Figure 6.1	Bar chart	75	<ul style="list-style-type: none"><li>• Functional - categorical data to provide comparison</li></ul>
Figure 6.2	Pie chart	70	<ul style="list-style-type: none"><li>• Functional - shows part-whole relation</li><li>• Human perception aware by provide only comparison of two categories</li></ul>

Figure	Chart type	Score	Characteristics
Figure 6.3	Slope chart	70	<ul style="list-style-type: none"> <li>• Human perception aware. Provides comparison of two categories</li> <li>• Clear slopes. Easy to follow</li> <li>• No clutter</li> <li>• Perceptual task aware. Uses position on a common scale (top of the hierarchy) to compare categories.</li> </ul>
Figure 6.4	Grouped-bar chart	70	<ul style="list-style-type: none"> <li>• Functional - comparing categories of two dimensions</li> <li>• No data overload. Limited number of groups presented with distance.</li> <li>• No clutter. The focus is on the data by dimming the grid lines. Labeling of categories and x-axis.</li> </ul>
Figure 6.5	Scatter plot	70	<ul style="list-style-type: none"> <li>• Functional - shows relationship between two numerical data-attributes</li> <li>• Highlights the data. It is distinguished from the background</li> <li>• Grid lines dimmed and widely positioned.</li> </ul>
Figure 6.6	Bubble chart	75	<ul style="list-style-type: none"> <li>• Functional - shows relationship between two numerical data-attributes</li> <li>• Uses the size of dots to present third fact.</li> </ul>

Table 6.1: Good charts with chart type and their score

## 6 Results

The visualization given in Figure 6.1 presents categorical data, showing the *Units sold* by *Regions*. As the dimension *Region* is sorted in ascending order by *Units sold* it can be easily understood which *Region* sold most or least *Units*. Therefore this chart is functional as it provides comparison between categories.

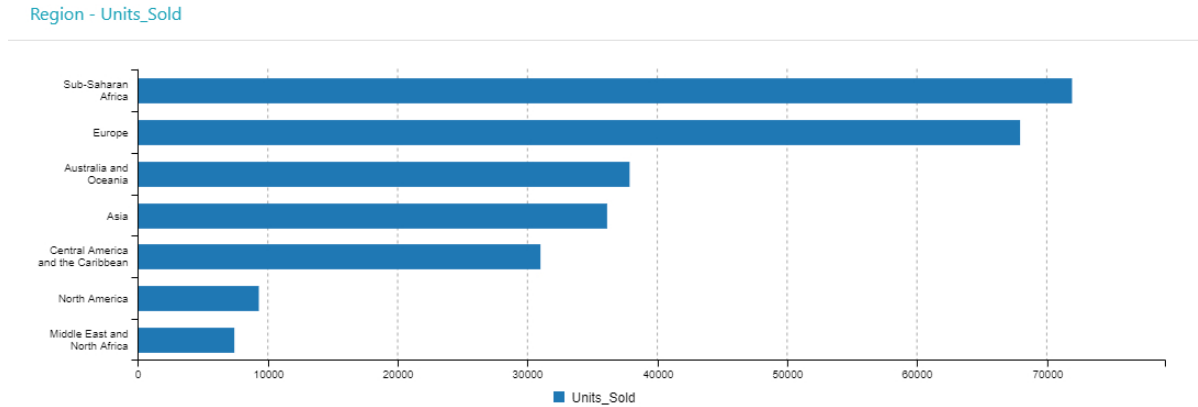


Figure 6.1: *Units sold per Regions*

Figure 6.2 shows as well categorical data. In this pie chart we can see the *Total costs* of each of the *Sales channels*. As this chart follows human perception of graphical objects having only two categories, we can easily judge the costs of the sales channels without looking at the numbers.

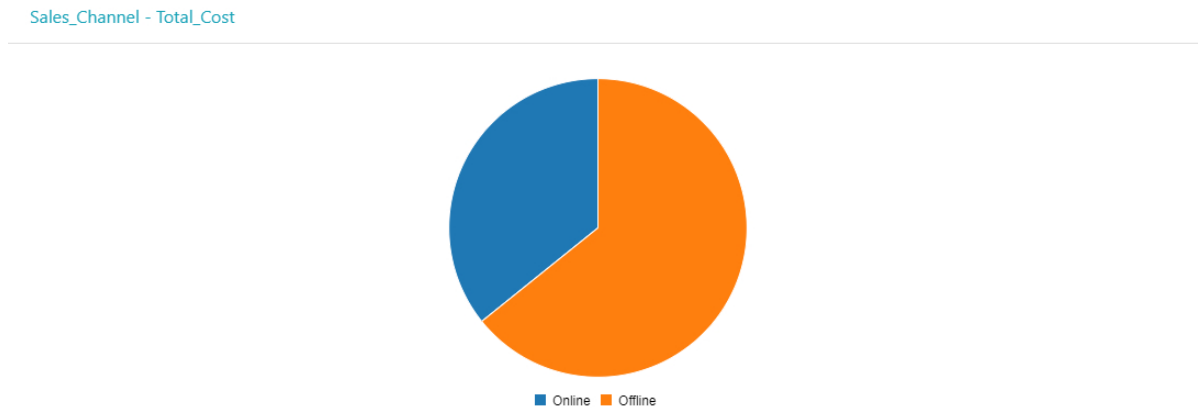


Figure 6.2: Example of a good pie chart showing *Total costs per Sales channel*

In the slope chart in Figure 6.3 are given the *Total costs* of *Order priority* for the



## 6 Results

two *Sales channels*. The purpose is to compare the categories of one dimension and their slope over another dimension. What clearly can be concluded here is that for all *Order priorities* the *Total costs* are higher for items sold *offline* then for items sold *online*. What is also easy to conclude is that, the *High Order priority* has biggest slope in comparison to the other categories. The human perception rules for creating slope charts (maximum of six categories with less intersection points) allow us to conclude this, as the chart shows four categories with only one intersection point.

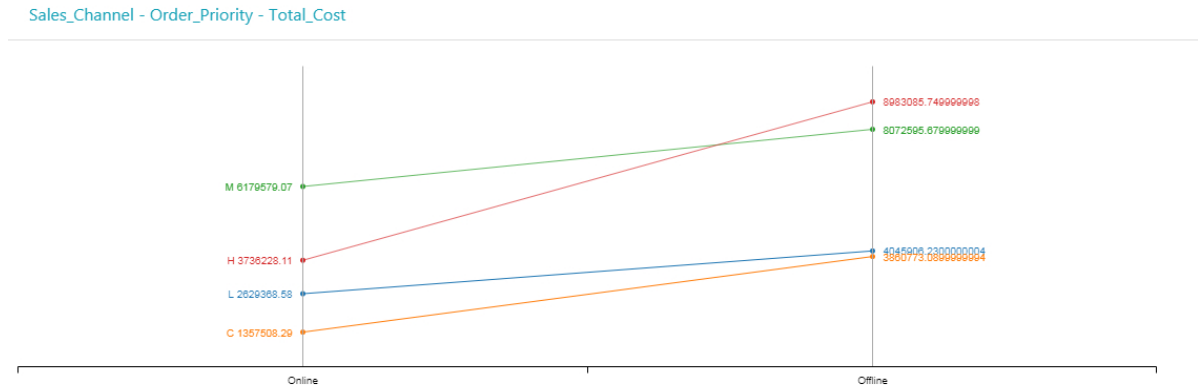


Figure 6.3: Good slope chart showing *Total costs* of *Order priorities* over two *Sales channels*

Similar to the previous charts, two dimensions can be shown with a grouped-bar chart. Figure 6.4 shows the number of *Units sold online* and *offline* per *Order priority*. Here as well we could take few conclusions: More items were sold *Offline* then *Online* for all four *Priority* groups; Biggest difference of items sold between each *Sale channel* is within *Priority C*.

## 6 Results

Order\_Priority - Sales\_Channel - Units\_Sold

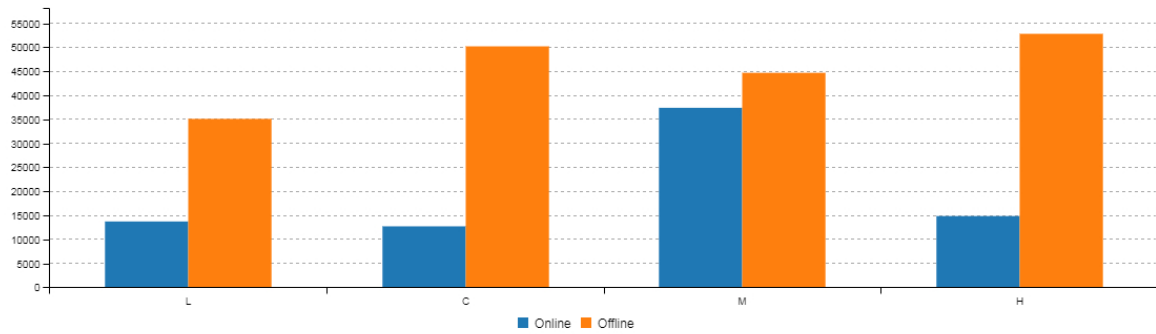


Figure 6.4: Example of a good grouped-bar chart displaying *Units sold* on different channels with different priority

The scatter chart from Figure 6.5 shows the positive correlation between *Total revenue* and *Total cost*. Therefore we can conclude that the more revenue the company makes the more costs it has.

Total\_Revenue - Total\_Cost

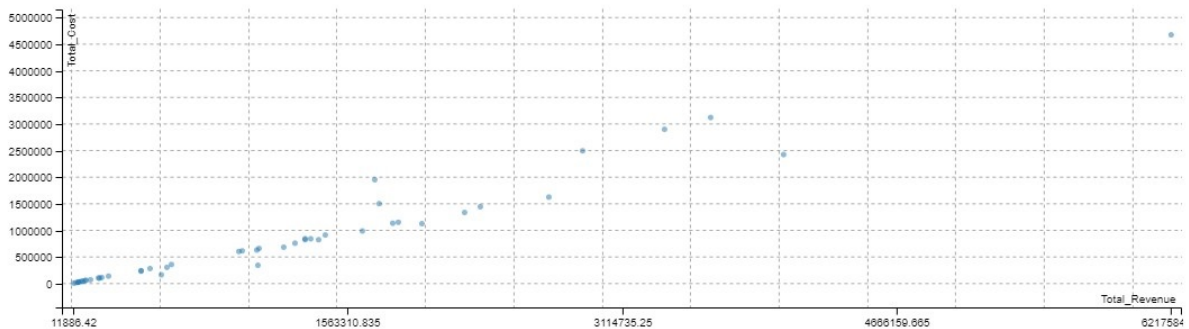


Figure 6.5: Scatter-plot showing *Total revenue* and *Total cost*

Lastly in the group of good chart we show the bubble chart in Figure 6.6. It shows three measures. We can detect the positive correlation between *Total revenue* and *Total cost*. The size of each bubble corresponds to the *Units sold*.

## 6 Results

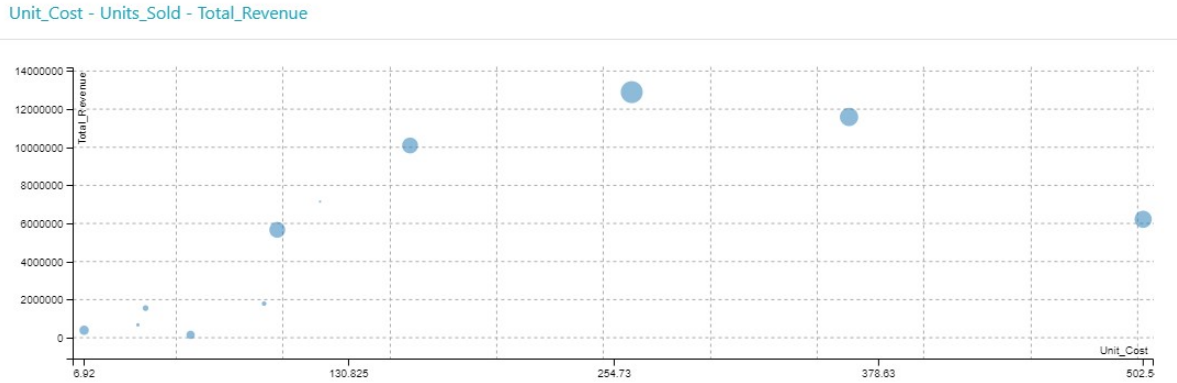


Figure 6.6: Example of bubble chart presenting *Total revenue* and *Total cost* with *Units sold* as size of dots

Now we present some of the charts that were marked as bad and received less than 70 utility score. Figures [6.7 - 6.13] show some examples of these charts. Furthermore we discussed their issues in Table 6.2 by giving the scores and the reasons why they were marked as bad visualizations.

Figure	Chart type	Score	Issues
Figure 6.7	Line chart	50	<ul style="list-style-type: none"> <li>• Missing values. Lines can neither be compared nor their trend understood</li> <li>• Clutter in chart. Big number of intersection points</li> <li>• Too many categories. Chart overload</li> </ul>
Figure 6.8	Multi-graph series	65	<ul style="list-style-type: none"> <li>• Missing data points</li> </ul>
Figure 6.9	Pie chart and Area chart	60	<ul style="list-style-type: none"> <li>• Chart overload with data</li> <li>• Too many categories displayed then recommended by human perception rules</li> <li>• Intersection points more than number of categories. Cluttered chart</li> </ul>

Figure	Chart type	Score	Issues
Figure 6.10	Slope chart	50	<ul style="list-style-type: none"> <li>• Data overload</li> <li>• More categories then recommended for slope charts (maximum 6)</li> <li>• Missing values results in no slopes for categories. Impossible to compare</li> </ul>
Figure 6.11	Stacked-bar chart	50	<ul style="list-style-type: none"> <li>• Missing a common baseline</li> <li>• Not functional. Time series data displayed with categorical chart</li> <li>• Missing stacks per bar. No comparison feasible</li> </ul>
Figure 6.12	Grouped-bar chart	50	<ul style="list-style-type: none"> <li>• Too many groups then recommended for grouped-bar chart (maximum 5 groups)</li> <li>• Missing values of categories in a group. Prevents comparing categories</li> <li>• Wrong type of data. Stacked bar relevant for displaying categorical values</li> </ul>
Figure 6.13	Scatter-plot chart	50	<ul style="list-style-type: none"> <li>• No correlation between the numerical attributes. Difficult to make a conclusion</li> </ul>

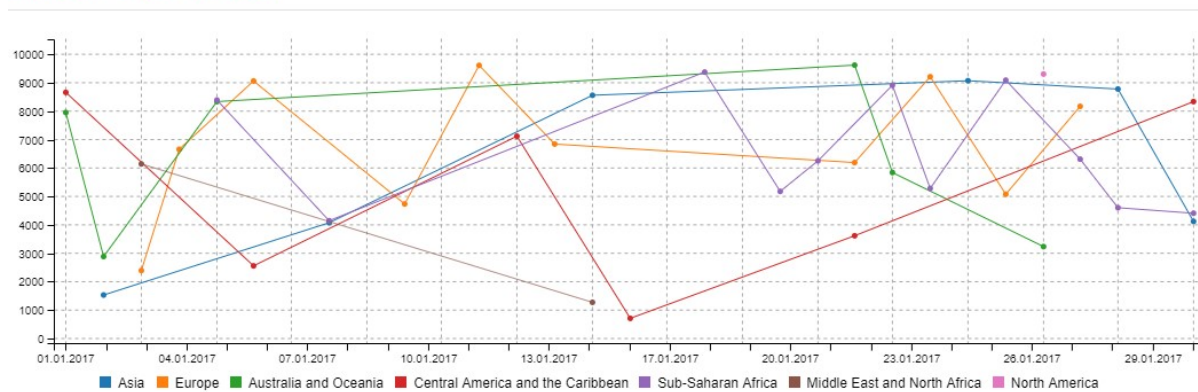
Table 6.2: Bad charts for the *Country sales* data-set and their issues

The two charts in Figure 6.7 show temporal data for the *Order date* time dimension. The chart in Figure 6.7a depicts the number of *Units sold* by *Regions*. The big fluctuation in

## 6 Results

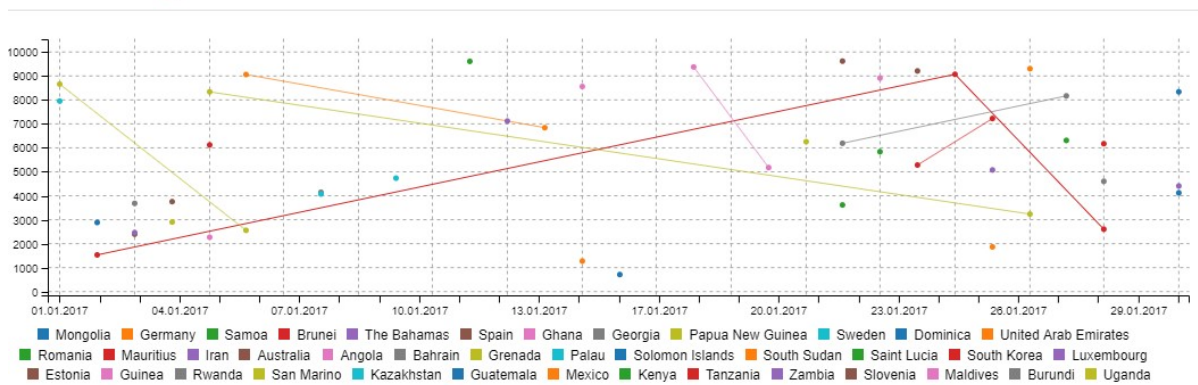
the data causes that the lines are neither clear to follow nor to compare. Although the number of categories (*Regions*) 7 is what line charts can display good concerning the human perception concept, due to the crisscrosses it is very difficult to make any conclusion. The visualization in Figure 6.7b gives similar data but for *Countries* instead of *Regions*. Two issues are to be detected: missing values and the big number of categories. We can not assess the trend line of *Units sold* by *Country* due to the missing values for some categories. Additionally charting 39 lines in one line chart is not recommended due to the chart load with graphical elements causing clutter.

Order\_Date - Region - Units\_Sold



(a) Cluttered line chart displaying high fluctuating data

Order\_Date - Country - Units\_Sold



(b) Chart overload showing 39 categories with missing values

Figure 6.7: Examples of bad lines charts due to big clutter or data overload

A multi-graph series is shown in Figure 6.8. The data shows *Units sold* of *Item type* over an *Order date*. For some of the categories in the chart it is easy to estimate the change over time, but it is difficult to make a conclusion for all *Item types*, especially when there

## 6 Results

is only one or two data points in the chart. Moreover, the comparison between charts in the series is becoming difficult.

Order\_Date - Item\_Type - Units\_Sold

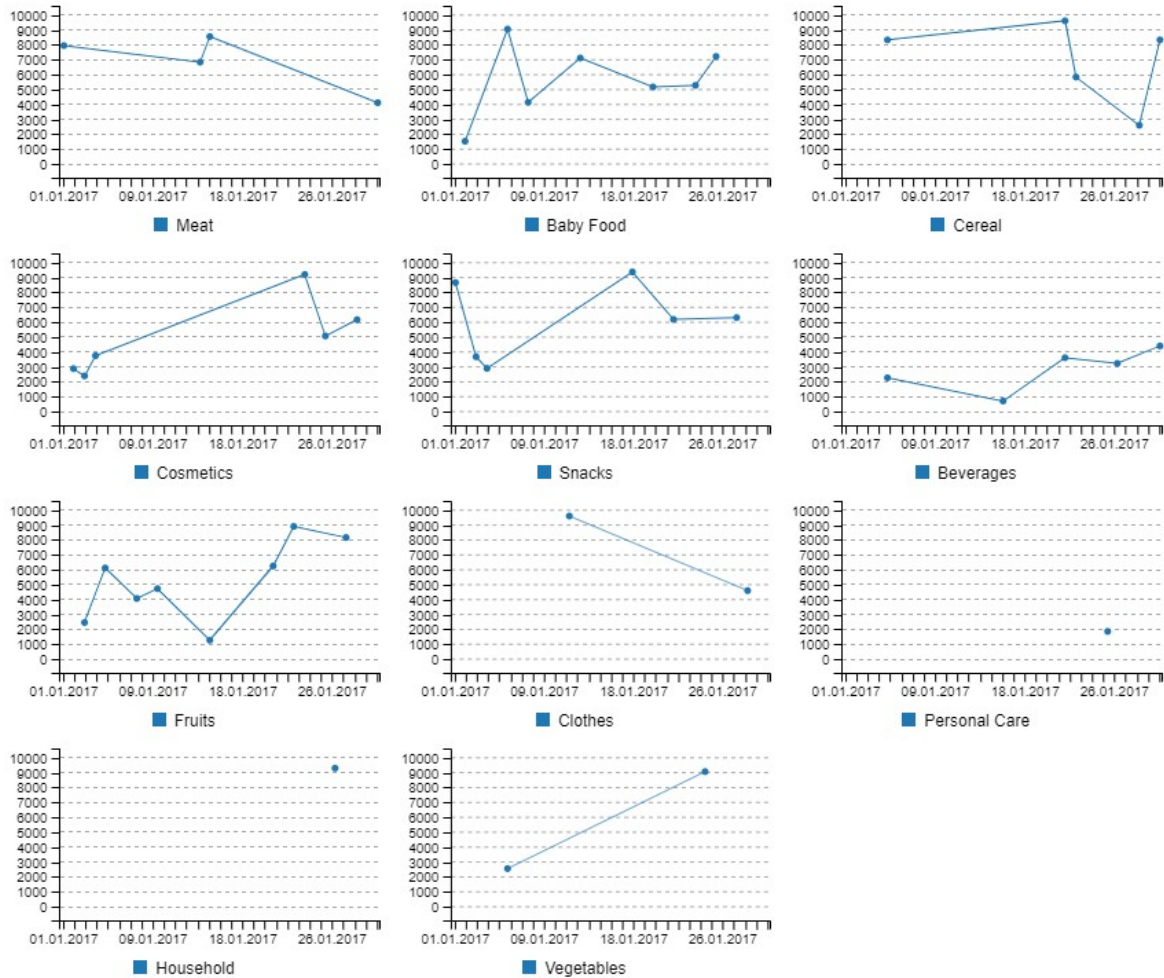
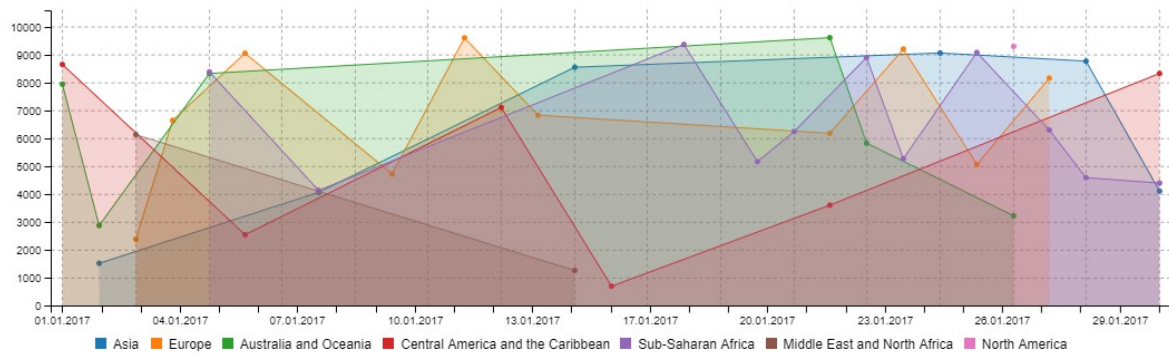


Figure 6.8: Multi-graph showing *Units sold* by *Item type*

The two visualizations in Figure 6.9 visualize dimensions that have part-whole relation. However in both charts this is difficult to see because either the number of data intersections is high (more than the number of categories), thus causing clutter as in Figure 6.9a or the number of categories is not appropriate for that chart type as in the example of pie chart in Figure 6.9b. Furthermore, as the slices in the pie charts are similar we cannot be sure which *Region* sold how many *Units*.

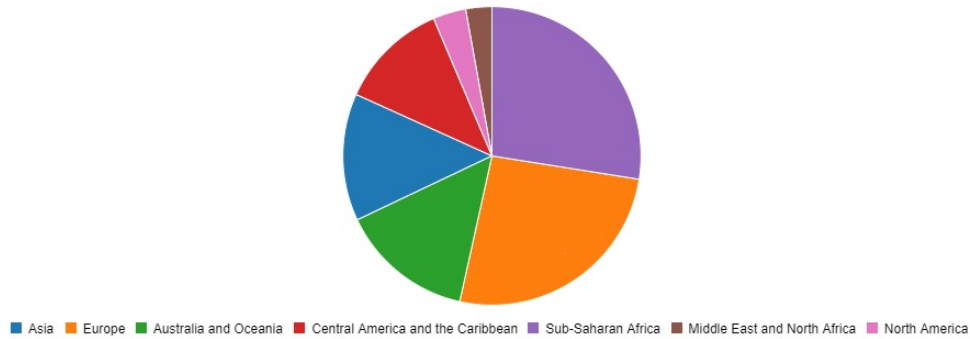
## 6 Results

Order\_Date - Region - Units\_Sold



(a) Area chart containing big clutter

Region - Units\_Sold



(b) Pie chart having too many evenly distributed categories

Figure 6.9: Examples of bad charts showing part-whole relation for *Country sales* dataset

Slope charts should be limited in number of categories and all categories must have values in order to apply the human perception rules and functionality of this chart type. When both of these characteristics are missing then such chart becomes chaotic, unclear and nonfunctional as given in Figure 6.10.

## 6 Results

Sales\_Channel - Country - Unit\_Price

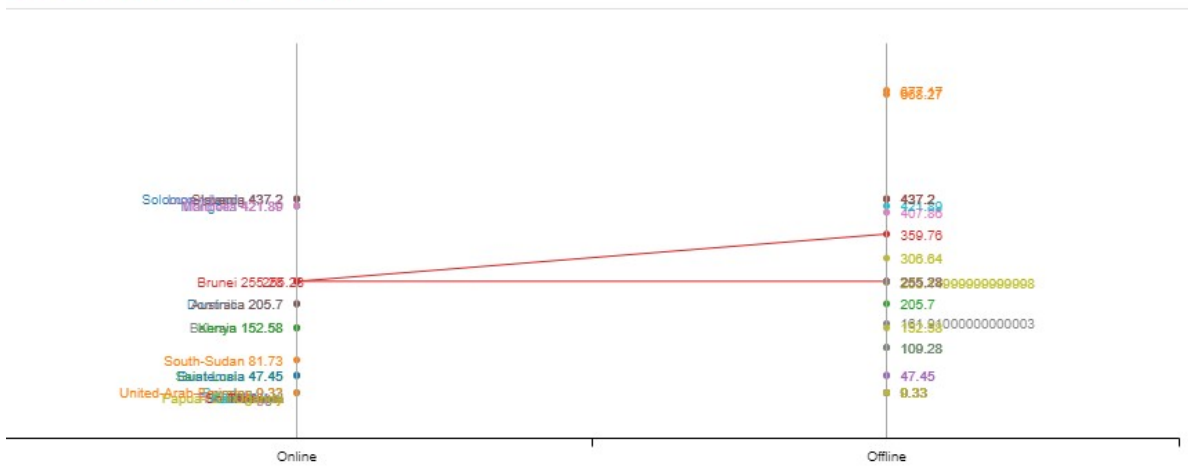
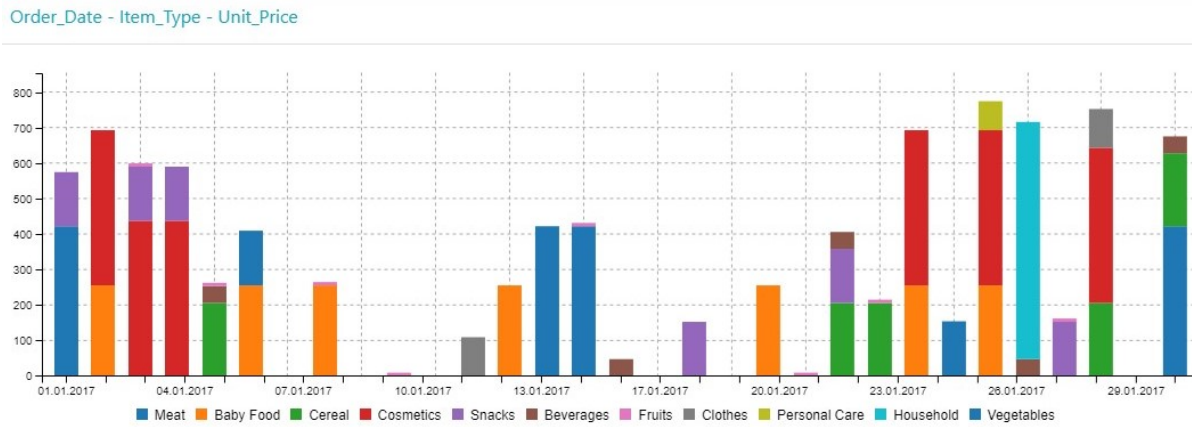


Figure 6.10: Overloaded slope chart containing few slopes and too many data points

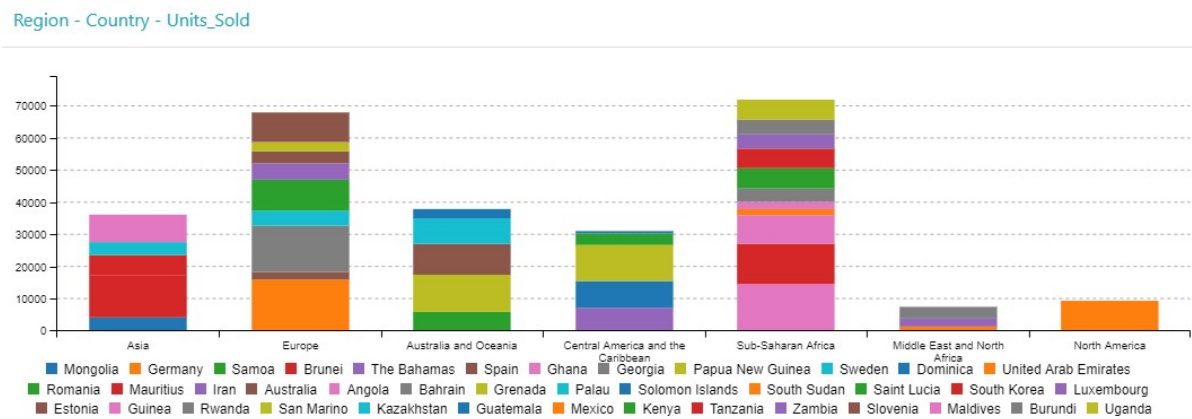
Now, let's take a look at the visualizations in Figure 6.11. The issues with the first stacked-bar are: it tries to show temporal data, which by our definition (sub-chapter 4.2.1) can only be shown with line charts; the baseline of the categories across x axis is not equal, thus it is difficult to compare which *Items* were sold most. Regarding the second chart, can we answer which *Country* sold most *Items* in *Asia* or *Europe*? The big number of stacks per bar and having missing categories in some x axis dimensions prevent to compare the data.



## 6 Results



(a) Temporal data displayed with stacked-bar chart



(b) Missing common baseline in a stacked-bar chart

Figure 6.11: Examples of bad stacked-bar charts

Grouped-bar charts are good when all bars are displayed and maximum up to five groups are shown to ensure no cognitive overload for the user. This is not the case with the chart in Figure 6.12. First, it shows temporal data resulting in not functional chart (compare categories in maximum five groups). Additionally, we cannot assess the trend of *Units sold* for the two *Sales channels* as no line is charted. Second it fails to provide comparison of the *Sales channels* as some of the bars are missing for certain *Order dates*. And lastly, too many x axis values are displayed for a grouped-bar to be considered as human perception aware.

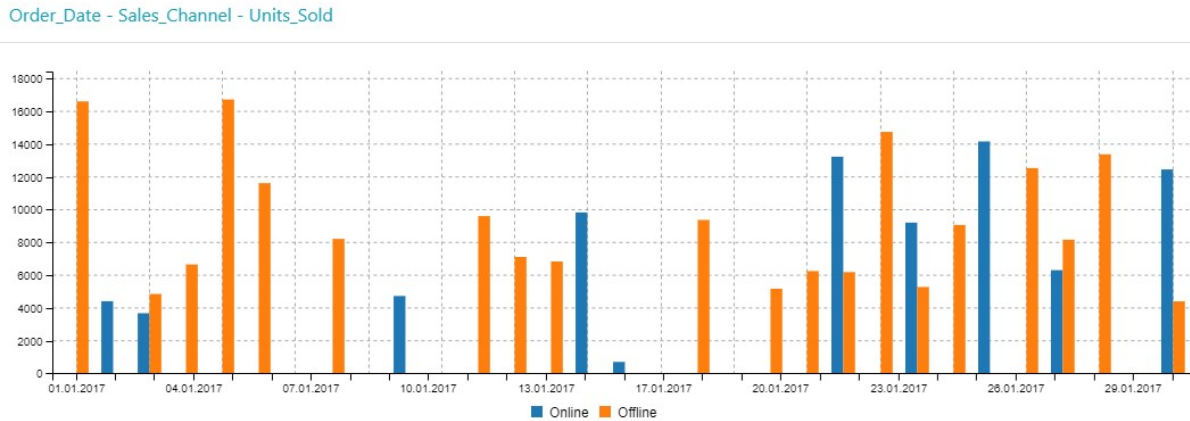


Figure 6.12: Too many groups of temporal data presented with grouped-bar chart

Our utility metric generates scatter or bubble charts only when the numerical dimensions have high positive or negative correlation coefficient. To quantify these terms, as already described in sub-chapter 4.2.2, we take values of more than 0.6 or less than -0.6 for positive or negative relationship respectively. The reason for this is that, when a correlation coefficient is a number between -0.6 - 0.6 then it becomes difficult for the user to assess the meaning of relationship. One such chart is given in Figure 6.13.

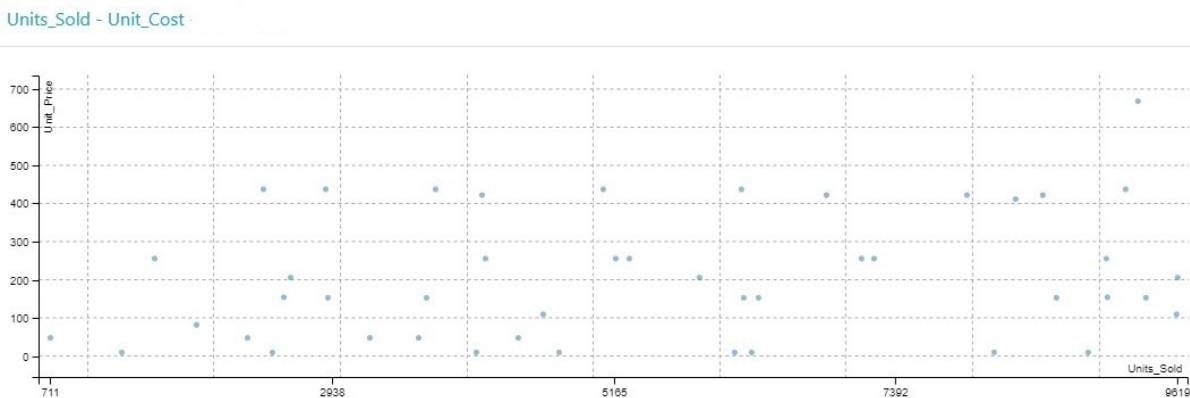


Figure 6.13: Two numerical attributes having no correlation displayed with scatter-plot

## 6.2 Anonymized Real World Data-set

We obtain a second data-set for which we generated good visualizations automatically. Now we use a real-world telecommunication data-set. It proves the feasibility of our

## 6 Results

utility metric for a real-world data. For easy notation we call this - *Customer* data-set. To ensure privacy of the data we anonymize the categorical values and the column names. A small subset is given in Appendix A.

The following columns were recognized as data attributes:

- Time dimension: Date; format: DD.MM.YYYY
- Dimensions: Area, Size, Weekday, calendarWeek, customerGroup, Product,
- Kpis: KPI1, KPI2, KPI3

It can be noticed that the column *calendarWeek* has temporal values but was not detected as a time dimension. The reason is that our module detects values of month, year and day separated by comma, dot or slash but not calendar weeks. The list of date-time formats our tool is able to detect is given in Appendix H.

The result from our *visualization detection* module is a list of 132 combinations of data attributes. These present chart objects that need to be evaluated. For each of the generated combinations we calculate 11 utility scores for each chart type, thus giving us at the end 1452 possible visualizations. The distribution of weights for each criteria for all 1452 charts is given in Figure 6.14. *Number of tuples*, *data type* and *number of categories* gave the most scores for the described data-set. All charts received scores based on the first criteria.

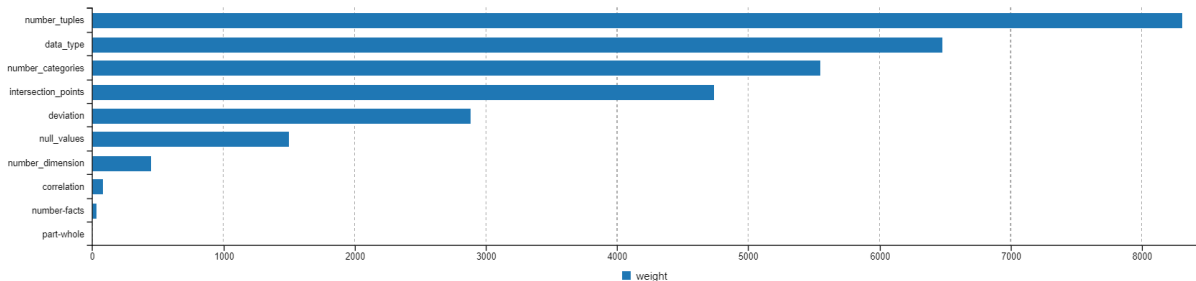


Figure 6.14: Scores for each criteria summed for all 1452 visualizations

The utility metric has recommended 86 good charts presenting same or different data. Figures [6.15 - 6.20] show some of the good charts grouped by their type. Table 6.3 provides an overview of these charts by giving the score and description of their characteristics.

## 6 Results

Figure	Chart type	Score	Characteristics
Figure 6.15	Line chart	70	<ul style="list-style-type: none"> <li>● Functional - shows changes over time</li> <li>● Allows comparison between categories</li> <li>● Easy to follow line. No crisscrossing</li> <li>● Focus on the data. No chart-junk</li> <li>● Chart type which people understand with high accuracy</li> </ul>
Figure 6.16	Multi-graph series	75	<ul style="list-style-type: none"> <li>● Functional - shows change in time</li> <li>● Big fluctuation in the data. No data overload in the chart. Lines easy to follow and compare</li> <li>● Removed clutter. Highlights the data</li> <li>● Uses chart type which people understand with highest accuracy</li> </ul>
Figure 6.17	Bar chart	75	<ul style="list-style-type: none"> <li>● Functional - categorical data to provide comparison</li> </ul>
Figure 6.18	Pie chart	70	<ul style="list-style-type: none"> <li>● Functional - shows part-whole relation</li> <li>● Four categories presented. Human perception aware</li> <li>● Categories are distinct in terms of size.</li> </ul>

Figure	Chart type	Score	Characteristics
Figure 6.19	Grouped-bar chart	70	<ul style="list-style-type: none"> <li>• Functional - comparing categories of two dimensions</li> <li>• No data overload. Limited to five groups and three categories per group to ensure human perception</li> <li>• No clutter. The focus is on the data by dimming the grid lines. Labeling of categories and x-axis.</li> </ul>
Figure 6.20	Slope chart	70	<ul style="list-style-type: none"> <li>• Human perception aware. Provides comparison of two categories.</li> <li>• Allows easy comparison of dimension over two categories</li> <li>• No clutter. Omitted background color, border lines and y axis</li> <li>• Perceptual task aware. Uses position on a common scale (top of the hierarchy) to compare categories.</li> </ul>
Figure 6.21	Scatter plot	70	<ul style="list-style-type: none"> <li>• Functional - shows relationship between two numerical data-attributes</li> <li>• Highlights the data. It is distinguished from the background</li> <li>• Grid lines dimmed and widely positioned.</li> </ul>

Table 6.3: Description of the good charts and their score for Customer data-set

Both visualizations in Figure 6.15 and 6.16 present time series chart, having *Date* on the x-axis, *KPI2* on y-axis and each line for single category of *Size* and *Product*. The charts are functional, as they provide change in trend of time, are clear and the categories are

## 6 Results

easy to follow and compare. Due to the big fluctuation in data, the multi-graph series are generated and given in Figure 6.16. All three *Products* are visualized separately to prevent overload.

Date - Size - KPI\_1

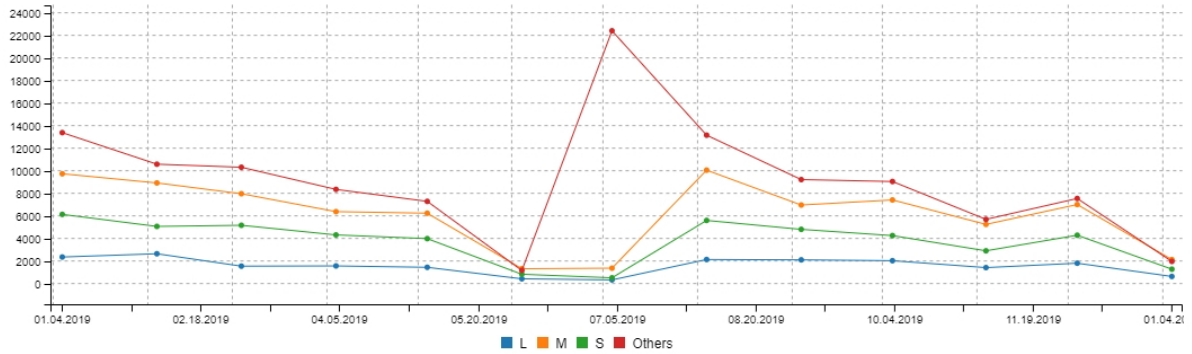


Figure 6.15: The change of *KPI1* for *Size* over *Date*

Date - Product - KPI\_2

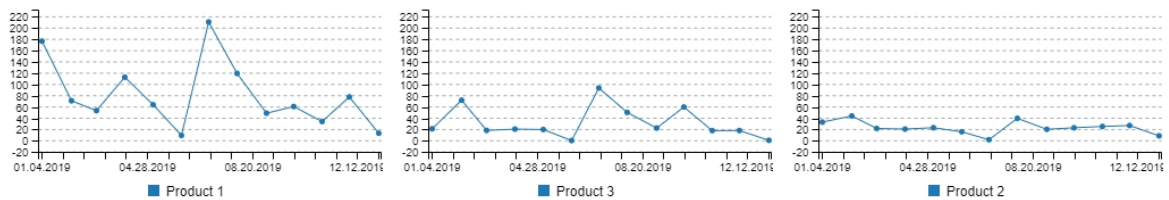


Figure 6.16: The change of *KPI2* for *Products* over *Date*

The visualizations in Figure 6.17 and Figure 6.18 show similar data. The bar chart has *KPI2* as value and *KPI3* is given in the pie chart. In both, we can easily assess what is the biggest or smallest category. Even though, the *Size Others* has the smallest value for *KPI2* in the bar chart, it shows biggest value for *KPI3* in the pie chart.

## 6 Results

Size - KPI\_1

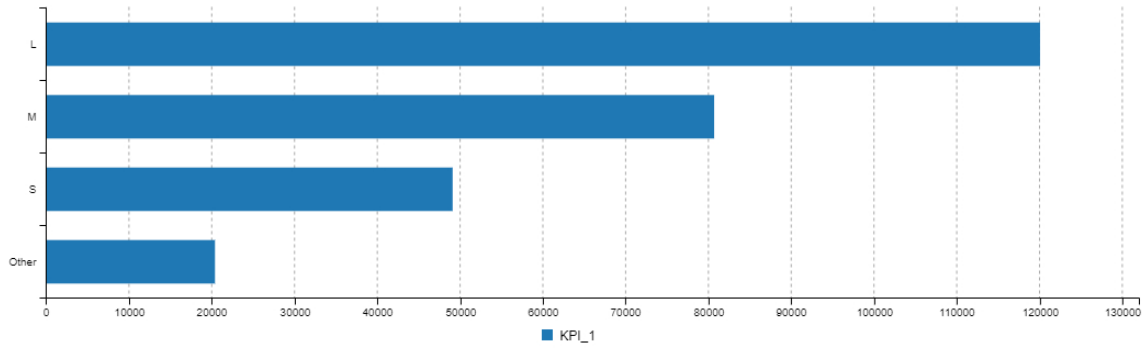


Figure 6.17: Values of *KPI1* for each *Size*

Size - KPI\_1

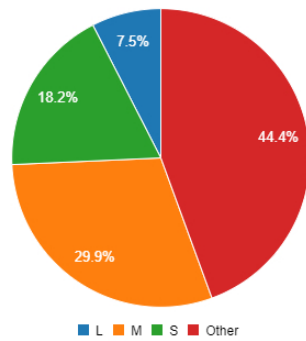
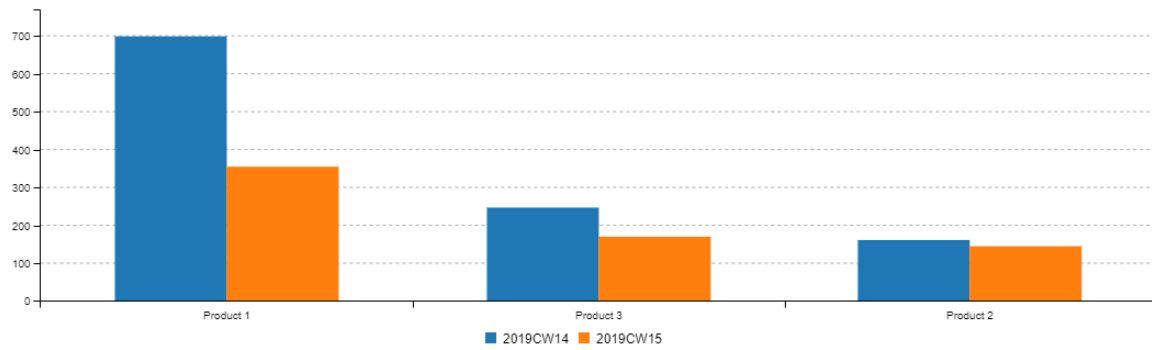


Figure 6.18: *KPI1* values for each *Size*

Good grouped-bar charts are given in Figure 6.19. The chart in Figure 6.19a shows *KPI3* per *calendarWeek* grouped by *Product*, so we can see that for all three *Products* they have bigger size of *KPI3* for *2019CW14*, then *2019CW15*. The purpose here is to compare the two dimensions. The visualization in Figure 6.19b provides the same dimension but for values of *KPI2*. Here we can see the *Products* grouped by *calendarWeek*, thus making it easier to compare products among *calendarWeeks*.

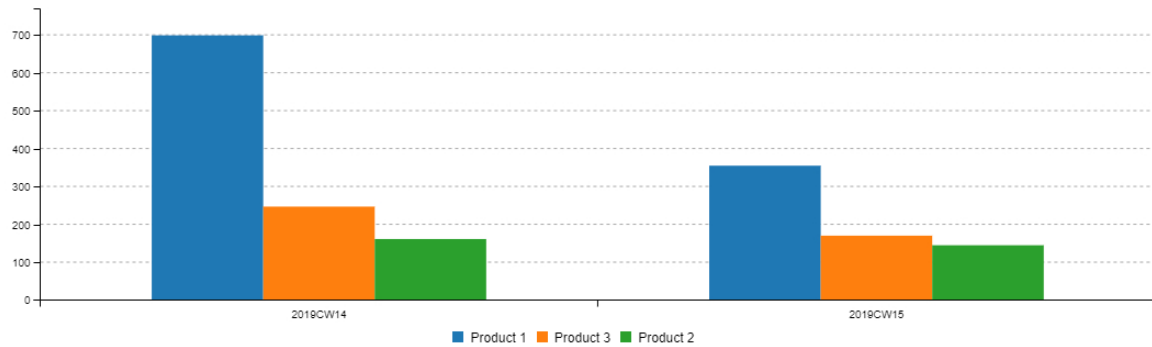
## 6 Results

Product - calendarWeek - KPI\_2



(a) Value of *KPI2* per *calendarWeek* grouped by *Product*

calendarWeek - Product - KPI\_2



(b) *KPI2* value for a single *Product* grouped by *calendarWeek*

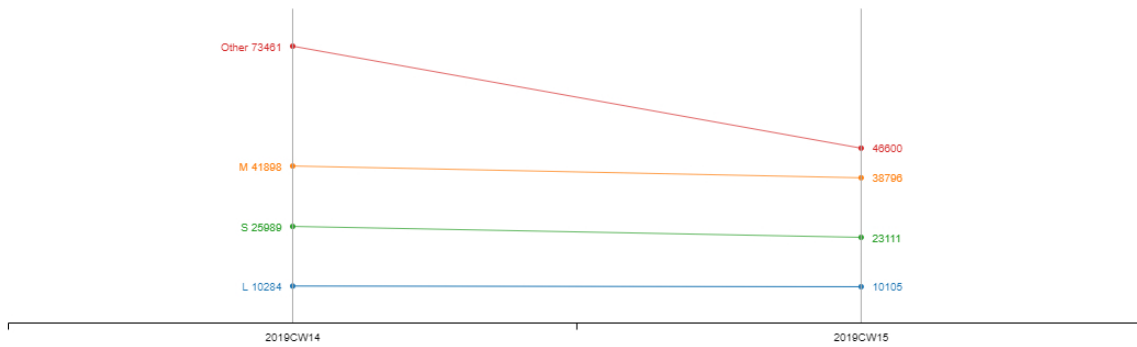
Figure 6.19: Examples of good grouped-bar charts

Similar to the previous charts, two dimensions can be shown with a slope chart. The chart in Figure 6.20a shows the change of *KPI1* value for *Size* per *calendarWeek*, Here we can conclude that *Other* has bigger change in value in contrast to the other values of *Size*, whereas the visualization in Figure 6.20a provides the change of *KPI2* for a single product per *calendarWeek*. Both charts are clear and easy to interpret.



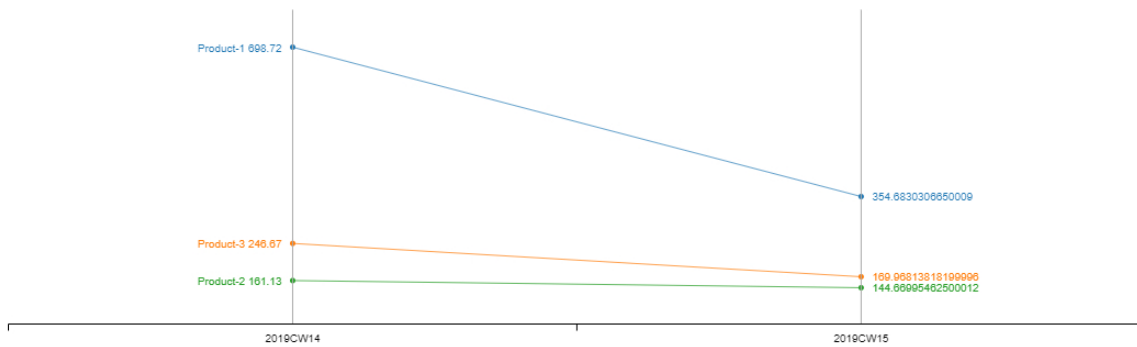
## 6 Results

calendarWeek - Size - KPI\_1



(a) Number of *KPI1* per *Size* for the two *calendarWeeks*

calendarWeek - Product - KPI\_2

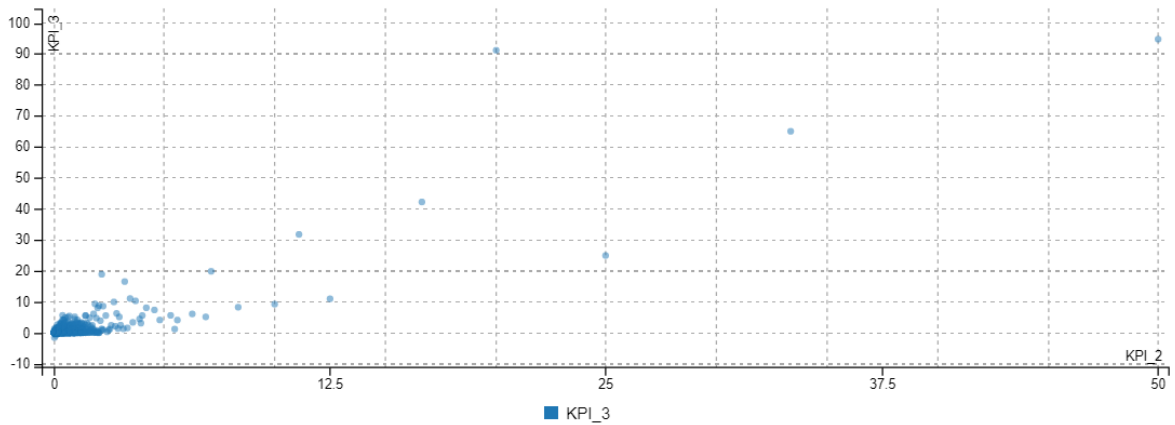


(b) Number of *KPI2* per *Product* for the two *calendarWeeks*

Figure 6.20: Slope charts

The scatter chart from Figure 6.21 shows the positive correlation between *KPI2* and *KPI3*.

KPI\_2 - KPI\_3 - KPI\_3

Figure 6.21: The correlation between *KPI2* and *KPI3*

Figures [6.22 - 6.26] show some examples of bad charts. Table 6.4 gives scores and issues related to the given score.

Figure	Chart type	Score	Issues
Figure 6.22	Line charts	60	<ul style="list-style-type: none"> <li>• Missing values. Categories can not be compared</li> <li>• Too many intersection points makes it difficult to follow a single line</li> </ul>
Figure 6.23	Multi-graph series	65	<ul style="list-style-type: none"> <li>• Too many data points on a single chart for categorical multi-graph series</li> <li>• Missing x-axis values prevents comparison of categories over this axis.</li> </ul>

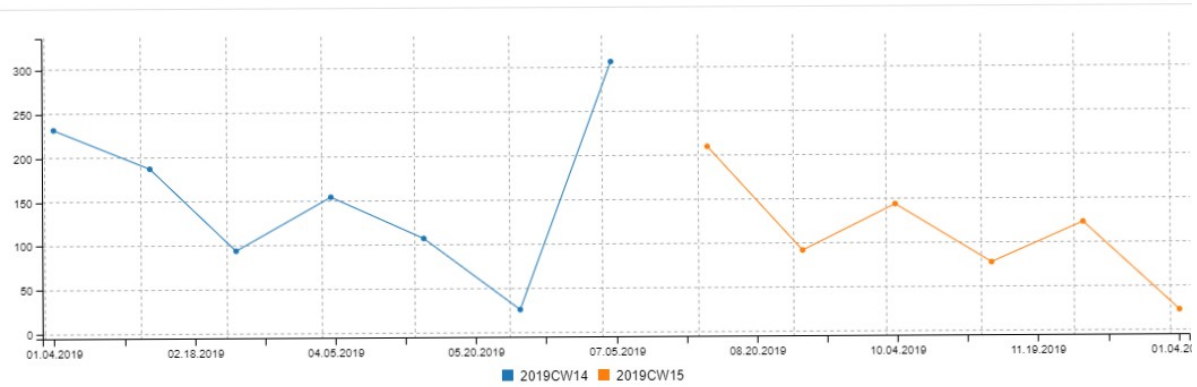
Figure	Chart type	Score	Issues
Figure 6.24a	Area charts	60	<ul style="list-style-type: none"> <li>• Chart overload with data</li> <li>• Intersection point more than categories.</li> <li>• Indication of relation when it does not exist</li> </ul>
Figure 6.24b	Pie and Grouped-bar charts	60	<ul style="list-style-type: none"> <li>• Chart overload with data</li> <li>• Too many categories shown. Difficult to make a conclusion.</li> <li>• Indication of relation where it does not exist</li> <li>• Missing values and too many groups in a grouped-bar chart</li> </ul>
Figure 6.25	Slope chart	60	<ul style="list-style-type: none"> <li>• Data overload</li> <li>• More categories than recommended for slope charts</li> <li>• Intersection of lines. Not clear to follow the slope of a certain category</li> </ul>
Figure 6.26	Stacked-bar chart	60	<ul style="list-style-type: none"> <li>• Difficult to compare categories as they do not have same base line</li> </ul>

Table 6.4: Bad charts with chart type and their score

The two charts in Figure 6.22 tend to show change in time for two dimensions: *calendarWeek* and *Size*. Even though both charts are functional, they fail to allow the user to compare the trend across the lines due to either different line length or their fluctuation which causes big clutter and unclear visualizations.

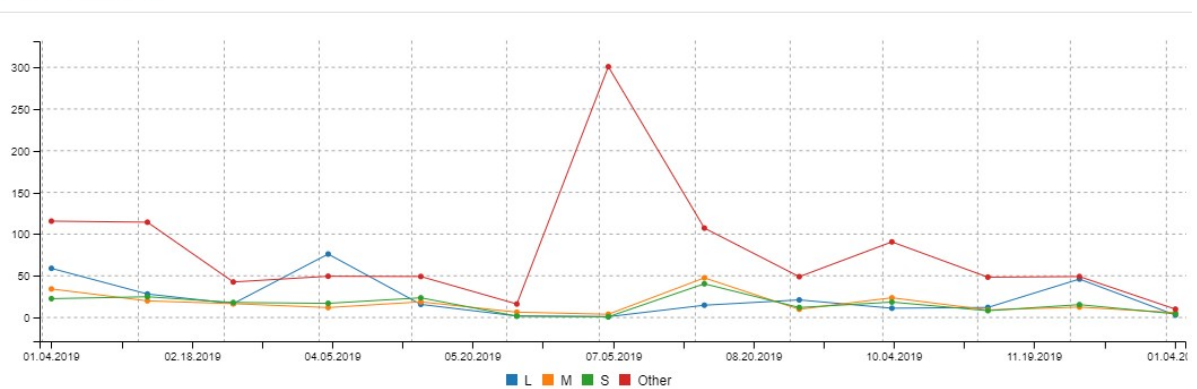
## 6 Results

Date - calendarWeek - KPI\_2



(a) Change of  $KPI_2$  for each *calendarWeek* per *Date*

Date - Size - KPI\_2



(b) Value of  $KPI_2$  for different *Size* per *Date*

Figure 6.22: Bad line charts

Showing line charts for categorical data can be tricky, especially when not all categories are written as values on the x-axis as in the example in Figure 6.23. We can clearly estimate the change per single category, but cannot make any conclusion for a single value of x-axis as we don't see labels. On the other side, showing all labels for this data, would affect the clarity and functionality considering the size of the single chart.

## 6 Results

Area - Size - KPI\_2

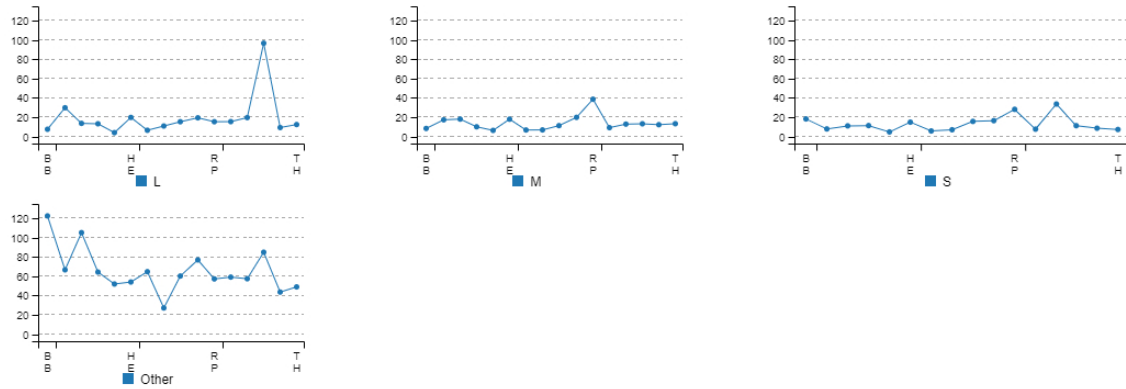
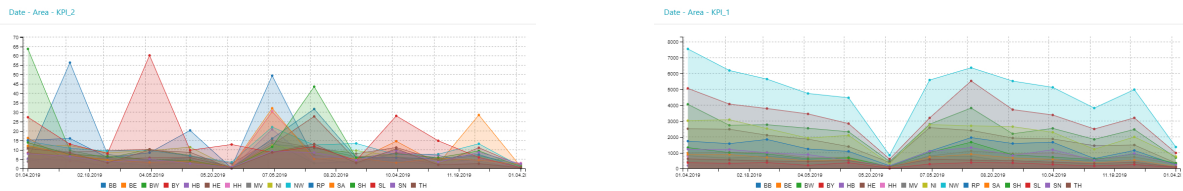
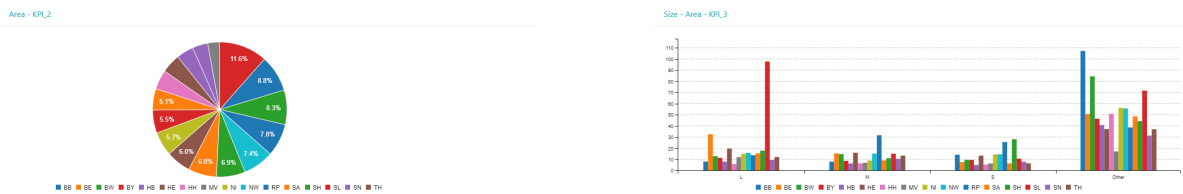


Figure 6.23: Change of  $KPI_2$  for Area given by Size

All charts given in Figure 6.24 try to visualize dimensions that have a big number of categories (e.g., more than 10). Both area charts given in Figure 6.24a have the same problem. The number of categories is too big and the trend line for single category varies too much for this data to be shown with area chart. They fail to provide any message as the clutter is too big and user cannot make a conclusion about a single category. Also, we cannot decide which category is bigger by looking at the pie chart nor the grouped-bar chart.



(a) Area charts



(b) Pie and grouped-bar charts

Figure 6.24: Examples of bad charts when too many categories need to be visualized

## 6 Results

A big overload for slope charts is when we try to visualize more than five categories as in the example in Figure 6.25. This example is chaotic, unclear and gives too much information that overloads the user and the graph.

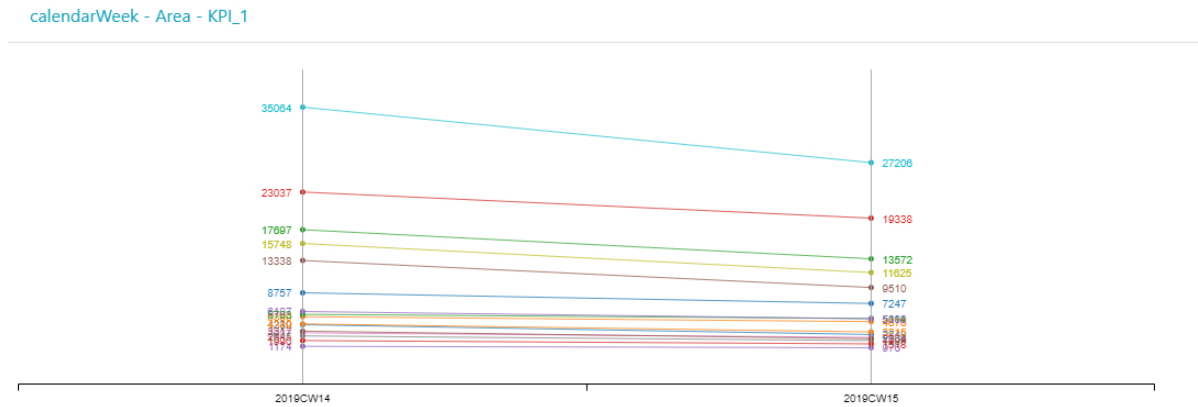


Figure 6.25: Size of *KPI1* for each *Area* per *calendarWeek*

At the end, let's take a look at the visualization in Figure 6.26. Can we assess the difference of *calendarWeek* among the x axis? The length of the bars can be measured correctly only if the baseline is the same for all bars. This example fails to provide the equal baseline for the two categories, thus making it difficult to compare the categories.

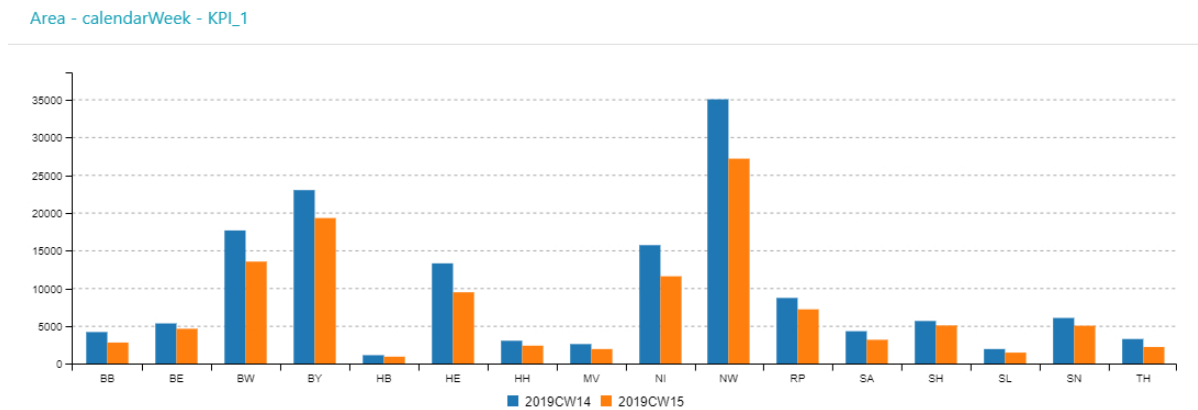


Figure 6.26: Number of *KPI1* per *calendarWeek* grouped by *Area*

### 6.3 Performance Measure

To measure the performance, we have computed the run-time for each process our tool executes from loading a page to showing the charts to the user. For the original customer data-set the following results were noted: To form all possible visualization objects and calculate their utility, our tool has required 25944.83ms. For ranking all 1452 visualization and selecting only the good ones, the tool needed in total 0.65ms. For showing the good charts to the user, the tool took 48595.64ms.

In order to test the run-time performance, we have made changes to the original Customer data-set by first duplicating the number of data attributes and second by duplicating the data rows. The goal is to compare the run-times and find out what affects the performance more. The results of the analyse are given in Table 6.5. The columns represent the processes while the rows represent each performance test. All values are expressed in milliseconds.

First, we have duplicated the number of data-attributes from the original Customer data-set. The new data-set now has 20 data-attributes. From the original 1 time dimension, 6 dimensions and 3 facts, we have: 2 time dimensions, 12 dimensions and 6 facts. In total 564 good charts were generated. For visualization objects creation and utility calculation, the tool needed 177852.63 ms, for ranking and selection, 4.22ms were required. The drawing of the good charts process took 1201453.86ms. As we were interested in finding out which single attribute contributes to increase of the run-time, we have performed this evaluation by duplicating the number of:

- Time dimensions. The data-set consists of: 2 time dimensions, 6 dimensions and 3 facts. The tool needed 31633.04ms for creation and calculation of utility metric, 0.62ms for ranking and 61195.76ms for presenting 95 good charts.
- Dimensions. Data-set includes: 1 time dimension, 12 dimensions and 3 facts. 45373.47ms for combining data attributes and utility metric calculation, 2.10ms for ranking and 299103.36ms for presenting 260 good charts.
- Facts: The new data-set has: 1 time dimensions, 6 dimensions and 6 facts. 133699.55ms were needed for the tool to loop over all the data attributes and evaluate the metric, 1.32ms for ranking and 148792.62ms for drawing 180 good charts.

For the second performance evaluation, we have uploaded altered version of the original Customer data-set by duplicating its rows. The tool required: 68040.019 - creating

visualizations objects and utility measure, 0.66ms - ranking and selecting good charts and 52145.44ms - for drawing good charts on the screen.

Table 6.5: Run-time comparison matrix for each process of TAG<sup>2</sup>S<sup>2</sup>

Test case	Utility metric calculation	Ranking	Drawing charts
Original data-set	25944.83	0.65	48595.64
Double the data-attributes	177852.63	4.22	1201453.86
Double numb. of time dimensions	31633.04	0.62	61195.76
Double number of dimensions	45373.47	2.10	299103.36
Double number of facts	133699.55	1.32	148792.62
Double rows	68040.02	0.66	52145.44

From this we can conclude that the performance time for calculation of the utility metric increases with the increase of the number of facts. The reason for that is this facts are part in almost all criteria (with the exception of the data type, number of dimensions, category, data-tuples) from the utility metric. All statistical calculations are performed on the facts by combining with other data attributes, thus affecting the run-time for this process.

## 6.4 Validating the Utility Metric

TAG<sup>2</sup>S<sup>2</sup> uses a utility metric of ten criteria in order to detect and automatically generate good visualization. To validate this utility metric, we obtain a ground truth data. The ground truth data is the recommended 86 charts generated by the utility metric and were described in sub-chapter 6.2. This data has been generated by running the utility metric against the Customer data-set already given in Table A.1. From the generated 86 good charts, 9 were time series charts, 75 show categorical data and 2 numerical charts. As shown in Figure 6.14, four criteria that received most scores are: number of data



tuples, data type, number of categories and number of intersection points, whereas three criteria that received least scores are number of dimensions, correlation and number of facts. In order to validate the metric, we have created two validation cases in which we have modified the criteria in different ways. As we received similar results after running the second test case, the further validation has been omitted.

### Case 1

For the first iteration of validating the utility metric, we have changed the weights assigned in the metric, thus the criteria with most scores have decreased values (High Importance (HI) = 3, Low Importance (LI) = 1), and the criteria that is present in all generated charts which received less scores have increased scores. Table 6.6 provides the altered weights for each criteria. The first column has the criteria name, second column gives the high importance weights and the last column provides the lower importance weights.

Criteria	Weight - HI	Weight - LI
Deviation	20	15
Null values	20	15
Number of dimensions	10	5
Number of facts	10	5
Part whole relation	10	5
Correlation coefficient	10	5
Number of categories	3	1
Data type	3	1
Number of data tuples	3	1
Intersection points	3	1

Table 6.6: Altered weights of the utility metric used for validation

The altered utility metric has been executed against the same data-set and the following results were noted: The distribution of scores for each criteria throughout the whole data-set is given in Figure 6.27. The number of total charts generated (1452) has not been changed, due to the same number of data attributes. As expected, the two criteria *deviation* and *null values* have received most scores while, the criteria *number of tuples*, *data type*, *number of categories* have less points. TAG<sup>2</sup>S<sup>2</sup> has generated 147 good charts.

## 6 Results

After comparing the generated charts with the ground truth, it was concluded that 85 out of 147 are actually good. From 10 time series charts, 9 were found in the ground truth data. From 135 categorical charts, only 74 are good and all of the numerical charts (2) were correctly detected as in the unmodified metric.

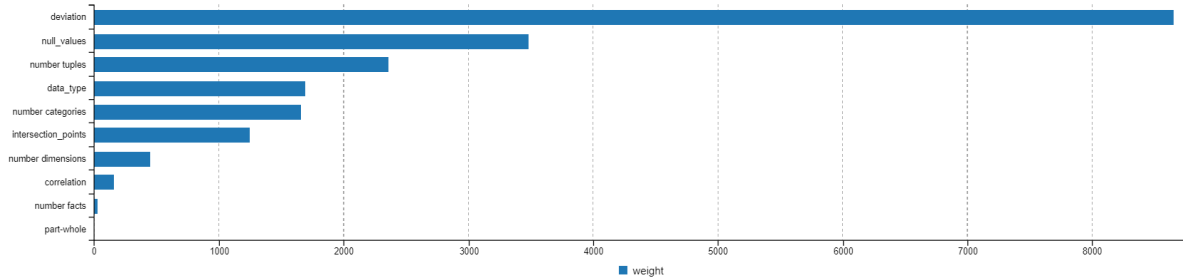
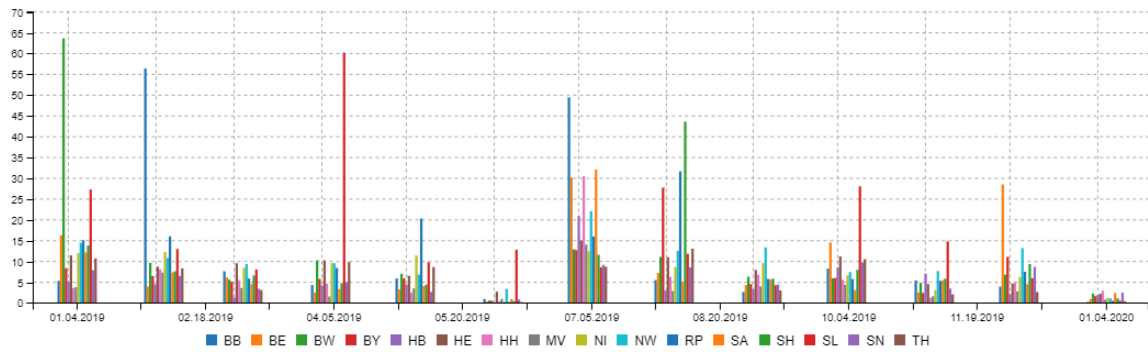


Figure 6.27: Good visualization automatically generated by TAG<sup>2</sup>S<sup>2</sup>

Figure 6.28 presents a grid of some of the visualizations generated by modifying the utility metric which were not found in the ground truth data. We noticed that majority of the visualizations present number of categories that are not appropriate for a specific chart type, missing values fail to provide an entire picture for easy comparison, crossing lines cause noise and add extra load to the graphs. Finally we have noticed that some of the charts as in the example in Figure 6.28a affect the functionality of the charts, meaning we receive categorical chart for showing time series data. Furthermore the generated charts are not human perception aware and have too many groups and bars per group.

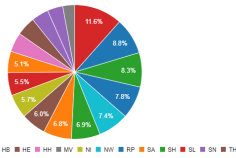
## 6 Results

Date - Area - KPL\_2

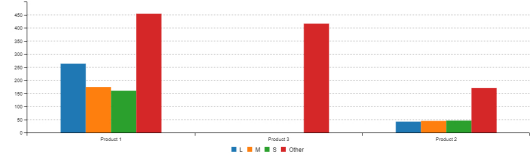


(a) Grouped-bar chart

Area - KPL\_2

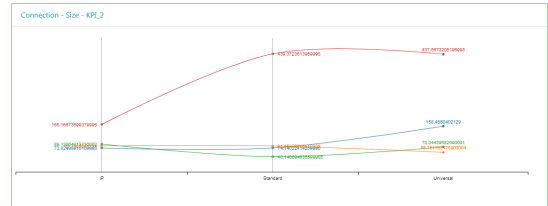
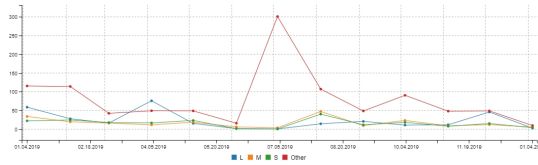


Product - Size - KPL\_2



(b) Pie and grouped-bar charts

Date - Size - KPL\_2



(c) Line and slope charts

Figure 6.28: Visualizations generated after modifying the utility metric

### Case 2

For the second validation, we have changed the number of criteria by giving score of 0 to the most scored criteria. The goal is to find out if a utility metric without these criteria will produce good charts. Therefore the first three criteria for this iteration of validation have 0 score for both high and low importance, and all other have not changed their weights. Table 6.7 provides the altered weights. The first column contains the name, the second gives weights for high importance and the last column provides the lower importance weights.

## 6 Results

Table 6.7: Altered weights of the utility metric used for second validation

Criteria	Weight - HI	Weight - LI
Deviation	10	5
Null values	10	5
Number of dimensions	10	5
Number of facts	10	5
Part whole relation	10	5
Correlation coefficient	10	5
Intersection points	10	5
Number of categories	0	0
Data type	0	0
Number of data tuples	0	0

Again as in the first case, this metric has been calculated for the Customer data-set and the following results were noted: The distribution of scores for each criteria throughout the whole data-set is given in Figure 6.29. We notice that good visualizations were selected upon only six criteria. As we did not change anything in the data, the number of total charts generated has stayed the same. TAG<sup>2</sup>S<sup>2</sup> has now generated 183 charts. After comparing with the truth data (the 86 visualizations generated by the utility metric described in sub-chapter 6.2), we concluded that only 77 are really good, meaning that we have received these charts with the unmodified metric. From 18 time series charts, 8 are part of the ground truth, 160 categorical charts from which 68 are true positive and 1 out of 5 numerical charts were seen as good.

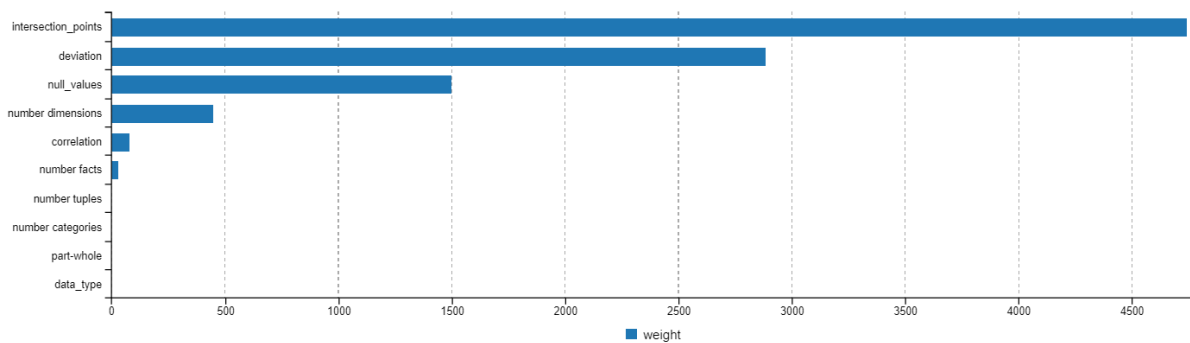
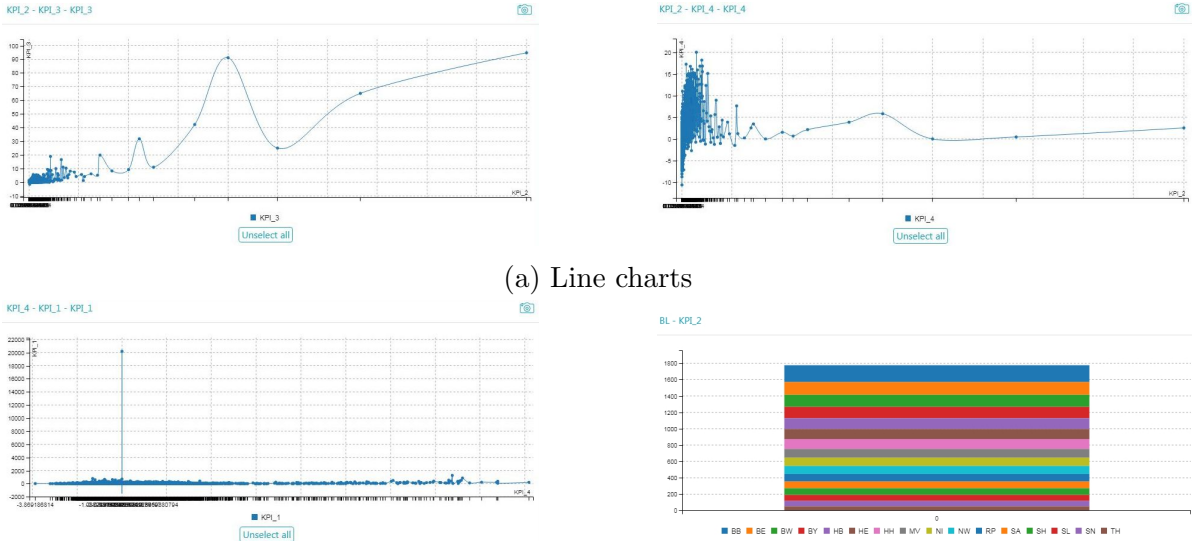


Figure 6.29: Good visualization automatically generated by TAG<sup>2</sup>S<sup>2</sup>

Figure 6.30 shows some of the visualizations generated by modifying the utility metric

## 6 Results

for second time which were not found in the ground truth data as well as were not generated in the first iteration. Even though the utility metric has been changed, the generated visualizations have the same reasons why they were not included in the ground truth. We notice that majority of the visualizations present numerical data. All charts here do not follow any distribution and it is difficult to make any conclusion. The data is too cluttered around one place which causes noise and unclear values on x-axis.



(a) Line charts

(b) Line and grouped-bar charts

Figure 6.30: Visualizations generated after re-modifying the utility metric

## 7 Conclusion and Future Work

This thesis focuses on defining and generating good visualizations. It mainly tackles the problem of automation in generating such visualizations. To establish a definition for good visualization, we took into account various criteria and considered different approaches in automation of visualization such as data-related metrics and human perception rules. Our approach is completely automated and considers data type, relations among data columns, statistical metrics and human perception for generating good charts.

We analyzed research approaches and commercial tools and we have identified weaknesses and strengths. There are different approaches to accomplish automation of generating visualization, but the majority of them generate visualizations of a single type of chart or provide visualization requiring reference data-set. They fail to consider new chart types (e.g., slope or multi-graph series) or human perception rules in order to avoid clutterness or to avoid the big cognitive load required for reading the charts when many objects are presented in the charts.

Another significant part of this thesis is the approach of scoring visualizations. This is an important step of our approach as it evaluates the utility (good or bad) of a specific chart. For this, a metric has been defined with 10 data related and user awareness criteria. This approach allowed us to measure how good one chart is for a given combination of data attributes. To our knowledge, this metric is unique and prolific and has not been used by any other other tools.

The main findings that answer the research questions of this thesis are:

1. No agreed definition exist of what good visualization is and how to generate one automatically. The current approaches only consider a single criteria for automation, provide interesting charts or are limited in providing different types of good charts;

## 7 Conclusion and Future Work

2. Good visualizations are those visualizations which are functional, present the data accurately by choosing graphical objects which people can associate with the meaning of the data and consist only of chart elements for the viewer to understand the data.
3. A utility metric has been defined for assessing visualization as good or bad. This metric is used for automatic detection of good visualization by assigning a utility score on each detected visualization. It consists of ten criteria, each criteria having a relevance depending on the chart type for which a score has been calculated.
4. TAG<sup>2</sup>S<sup>2</sup> has been developed. It presents a Tool for Automatic Generation of Good viSualization using Scoring. Employs the previously described approach for generating good visualization. Table 7.1 shows how the comparative matrix (Table 3.6) looks for TAG<sup>2</sup>S<sup>2</sup>. It can be noticed that our tool satisfies almost all criteria.

Evaluation criteria	TAG <sup>2</sup> S <sup>2</sup>
Automatic visualization	
No chart types restriction	
Clutter free	
Interactivity	
Graphical perception	
Multi-graph series	
Design princ. multi-series	
Time series data	
Categorical data	

Evaluation criteria	TAG <sup>2</sup> S <sup>2</sup>
Numerical data	
Data type	
Cardinality	
Deviation	
Correlation coefficient	
Part-Whole relation	
Calculation of intersection	
Aggregation	
Reference data-sets	
Binning parameter	
Data sorting	
Allow missing values	

Table 7.1: Evaluation criteria observed for TAG<sup>2</sup>S<sup>2</sup>

One possible direction for future research work is developing a configuration for TAG<sup>2</sup>S<sup>2</sup>. As each criteria part of the utility metric has different importance depending on the chart type, or their purpose, a configuration can be developed by providing a new minimum score for a certain type of data, chart, user or purpose. Alternatively the new metric could consider the user experience or their knowledge in data visualization, thus extending the score of good charts. Additionally, the configuration of the criteria weights can be enlarged to consider the new goal or the newly extended metric.



## 7 Conclusion and Future Work

Training a machine learning model that will improve the criteria scores would refine the recommendation results. To accomplish this one could follow user actions or measure the time user takes to analyse one visualization. These models trained on a big set of good visualizations will continuously learn from existing data and adapt, providing more accurate visualizations while simultaneously improving the quality of the recommended charts.

When it comes to the utility metric, we see possibility for future work in adding new criteria regarding the user experience in data visualization or in business intelligence. By differentiating between user types (e.g., experts, manager or regular user) additional information could be given or visualized, such as prediction of trend line. Two different users may not have same conclusions depending on their previous experiences and level of expertise. On the one hand, certain type of users could draw conclusion which will put the company in additional costs and on the other, users incorrect conclusions could put the company at risk.

Another possible future direction would be including more chart types, thus allowing the tool to work with other data types (e.g. geo related data-sets). For this to be done, the actual utility metric and its weights can be altered or complemented with other criteria relevant. Furthermore, including more chart types to the set of chart types that TAG<sup>2</sup>S<sup>2</sup> already work with. For instance heat maps, histograms, box plot or this can be easily achieved by updating the chart properties object and understanding the importance of each of the ten criteria from the utility metric.

Regarding to the developed tool, one possible future direction would be making the page and the charts generated more responsive, thus charts with more data points would be omitted when the tool is opened on small devices. Alternatively, the presented approach and metric do not consider different resolutions and therefore can be extended to include the size of the display in order to generate charts which their content will fit better depending on the size of the device.

We identify two weaknesses in our approach: the detection of time dimensions and recognition of ID columns in a data-set to prevent charting them as kpis.

- Our visualization detection module works only with a limited set of predefined date time formats. These formats contain only the day, month and the year. Another aspect is the correct detection of date time formats. The charting library is able

## 7 Conclusion and Future Work

to generate time series visualizations only if the date time format is correctly provided.

- In our visualization detection module we distinguish between measures, dimensions and time series values. As some tables come with a column for storing ID for each row, we take this attribute and chart it as a measure or dimensions, which in fact should be omitted.

# A Anonymized telecommunication data-set

Date	Area	Size	Weekday	Calender week	Customer group	Product	KPI1	KPI2	KPI3
01.04.2019	BB	L	Mo	2019cw14	Group 1	Product 1	2	0.03852	0.07787
01.04.2019	BB	L	Mo	2019cw14	Group 5	Product 8	2	0.0475	0.1177
01.04.2019	BB	L	Mo	2019cw14	Group 1	Product 6	6	0.1897	0.3771
01.04.2019	BB	M	Mo	2019cw14	Group 1	Product 1	18	0.0240	0.3250
01.04.2019	BB	M	Mo	2019cw14	Group 2	Product 6	2	0.0251	0.0142
01.04.2019	BB	M	Mo	2019cw14	Group 3	Product 1	21	0	0
...	...	...	...	...	...	...	...	...	...

Table A.1: Subset of the Customer data-set

# B reports2go Visualizations



Figure B.1: Size of *KPI1* for each *Area* distributed by *Date*

## B reports2go Visualizations



Figure B.2: *KPI1* values for each *Customer Group* per *Date*

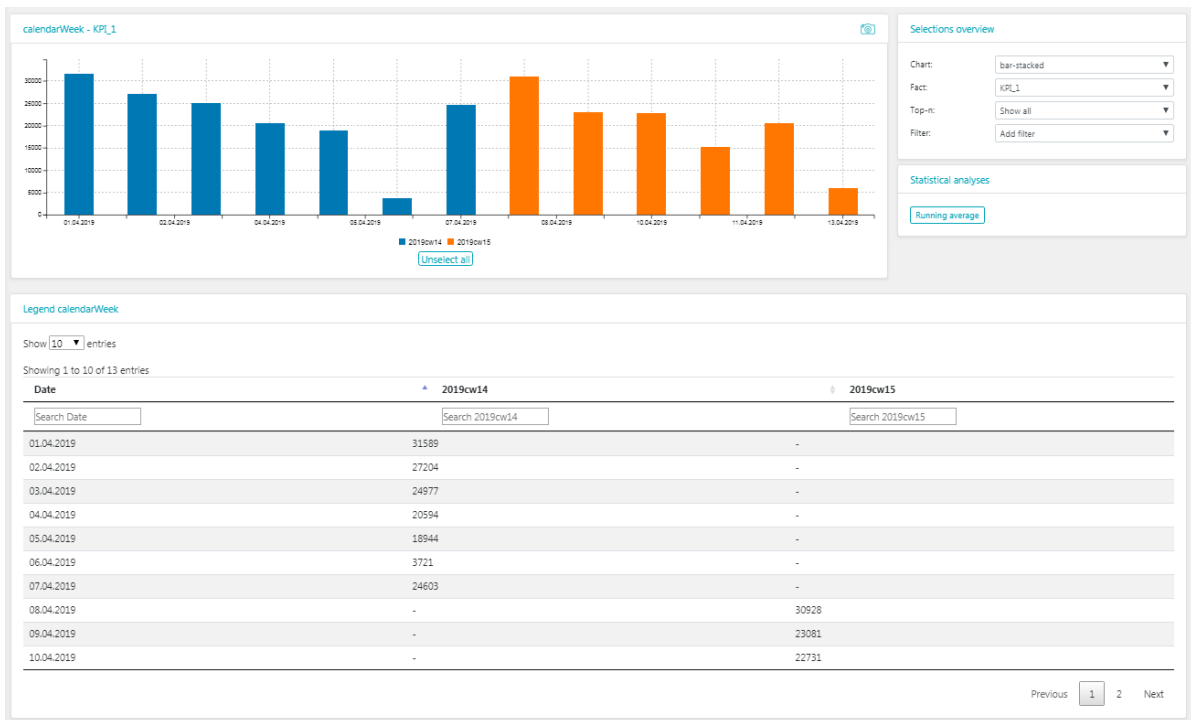


Figure B.3: *calendarWeek* per *Date* given for the size of *KPI1*

## B reports2go Visualizations

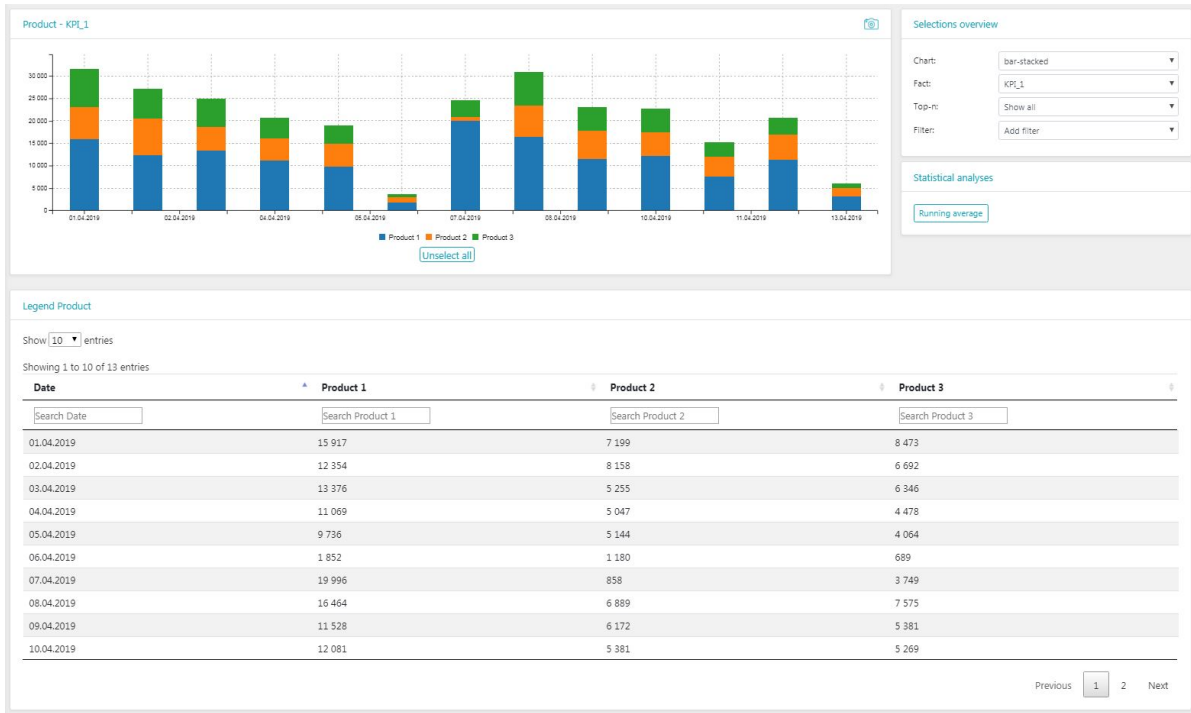


Figure B.4: *KPI1* values for *Product* given per *Date*

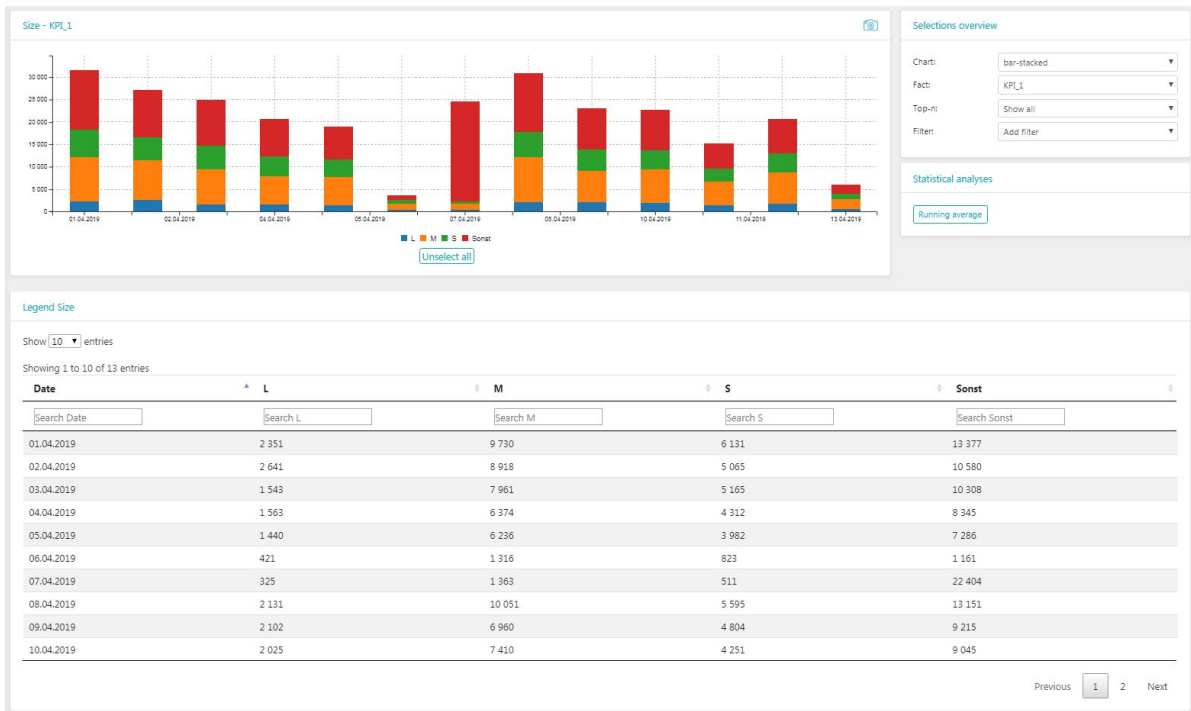


Figure B.5: *KPI1* Size per *Date*

# C TAG<sup>2</sup>S<sup>2</sup> visualizations

Date - Area - KPI\_1

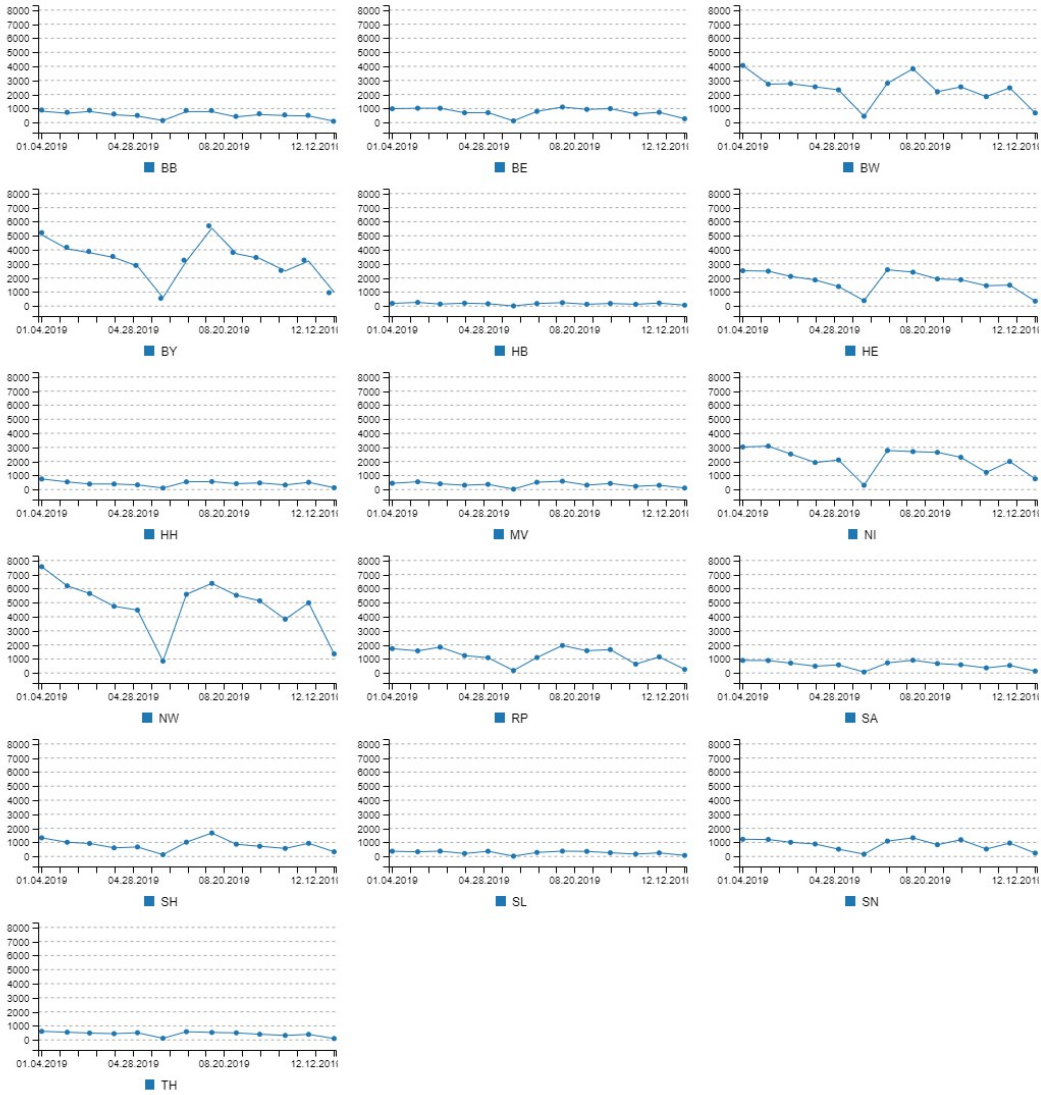


Figure C.1: KPI1 value by Area over a Date

Date - Product - KPI\_1

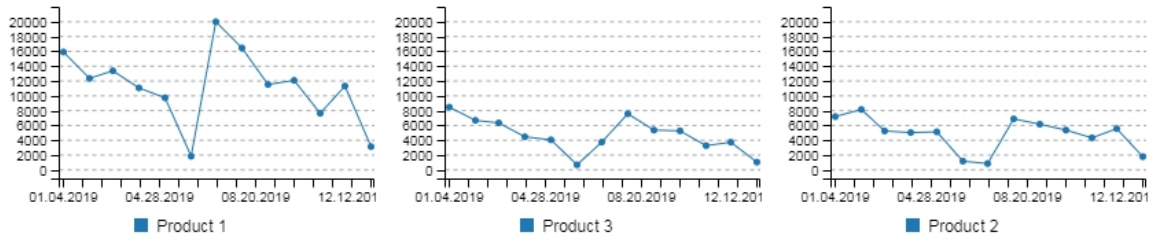


Figure C.2: *KPI1 Product size per Date*

Date - Size - KPI\_1

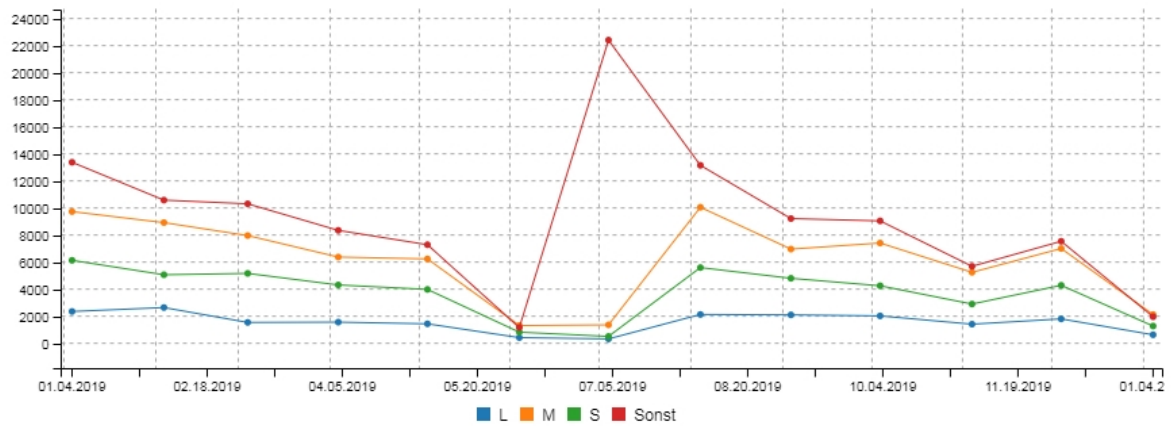


Figure C.3: *Size categories shown with measure KPI1 per Date*



## D Chart properties object

```
"categories": {
  "slope": {
    "xAxisLabel": "",
    "yAxisLabel": "",
    "legendPosition": "right",
    "showLegend": false,
    "axesRotate": false,
    "xCulling": 0,
    "xTickCount": 0,
    "xGridShow": true,
    "yGridShow": false,
    "dataLabelsShow": true,
    "showYAxis": false,
    "optionalGridLines": [
      {"value": data[0][x_axis]},
      {"value": data[1][x_axis]}
    ],
    "optionalYGridLines": [],
    "showDataPoint": true,
    "dataOrder": null
  },
  "bar": {
    "xAxisLabel": "",
    "yAxisLabel": "",
    "legendPosition": "bottom",
    "showLegend": true,
    "axesRotate": true,
    "xCulling": 0,
    "xTickCount": 0,
    "xGridShow": false,
    "yGridShow": true,
    "dataLabelsShow": false,
    "showYAxis": true,
```

## D Chart properties object

```
    "optionalGridLines": [],
    "optionalYGridLines": [],
    "showDataPoint": true,
    "dataOrder": null
  },
  "hor-stacked-bar": {
    "xAxisLabel": "",
    "yAxisLabel": "",
    "legendPosition": "bottom",
    "showLegend": false,
    "axesRotate": true,
    "xCulling": 10,
    "xTickCount": 30,
    "xGridShow": false,
    "yGridShow": false,
    "dataLabelsShow": true,
    "showYAxis": false,
    "optionalGridLines": [],
    "optionalYGridLines": [],
    "showDataPoint": false,
    "dataOrder": null
  },
  "pie": {
    "xAxisLabel": "",
    "yAxisLabel": "",
    "legendPosition": "bottom",
    "showLegend": true,
    "axesRotate": false,
    "xCulling": 10,
    "xTickCount": 30,
    "xGridShow": true,
    "yGridShow": true,
    "dataLabelsShow": false,
    "showYAxis": true,
    "optionalGridLines": [],
    "optionalYGridLines": [],
    "showDataPoint": true,
    "dataOrder": "desc"
  },
  "vert-stacked-bar": {
    "xAxisLabel": "",
    "yAxisLabel": "",
```

## D Chart properties object

```
    "legendPosition": "bottom",
    "showLegend": true,
    "axesRotate": false,
    "xCulling": 10,
    "xTickCount": 30,
    "xGridShow": true,
    "yGridShow": true,
    "dataLabelsShow": false,
    "showYAxis": true,
    "optionalGridLines": [],
    "optionalYGridLines": [],
    "showDataPoint": true,
    "dataOrder": "desc"
  },
  "multi-graph": {
    "xAxisLabel": "",
    "yAxisLabel": "",
    "legendPosition": "bottom",
    "showLegend": true,
    "axesRotate": false,
    "xCulling": 5,
    "xTickCount": 4,
    "xGridShow": false,
    "yGridShow": true,
    "dataLabelsShow": false,
    "showYAxis": true,
    "optionalGridLines": [],
    "optionalYGridLines": [],
    "showDataPoint": true,
    "dataOrder": null
  }
},
"timeseries": {
  "stacked-bar": {
    "xAxisLabel": "",
    "yAxisLabel": "",
    "legendPosition": "bottom",
    "showLegend": true,
    "axesRotate": false,
    "xCulling": 10,
    "xTickCount": 30,
    "xGridShow": true,
```

## D Chart properties object

```
    "yGridShow": true ,
    "dataLabelsShow": false ,
    "showYAxis": true ,
    "optionalGridLines": [] ,
    "optionalYGridLines": [] ,
    "showDataPoint": false ,
    "dataOrder": null
  },
  "multi-graph": {
    "xAxisLabel": "" ,
    "yAxisLabel": "" ,
    "legendPosition": "bottom" ,
    "showLegend": true ,
    "axesRotate": false ,
    "xCulling": 4 ,
    "xTickCount": 0 ,
    "xGridShow": false ,
    "yGridShow": true ,
    "dataLabelsShow": false ,
    "showYAxis": true ,
    "optionalGridLines": [] ,
    "optionalYGridLines": [] ,
    "showDataPoint": true ,
    "dataOrder": null
  },
  "XY": {
    "xAxisLabel": "" ,
    "yAxisLabel": "" ,
    "legendPosition": "bottom" ,
    "showLegend": true ,
    "axesRotate": false ,
    "xCulling": 0 ,
    "xTickCount": 10 ,
    "xGridShow": true ,
    "yGridShow": true ,
    "dataLabelsShow": false ,
    "showYAxis": true ,
    "optionalGridLines": [] ,
    "optionalYGridLines": [] ,
    "showDataPoint": true ,
    "dataOrder": null
  },
}
```

## *D Chart properties object*

```
"ver-stacked-bar": {
  "xAxisLabel": "",
  "yAxisLabel": "",
  "legendPosition": "bottom",
  "showLegend": true,
  "axesRotate": false,
  "xCulling": 10,
  "xTickCount": 30,
  "xGridShow": false,
  "yGridShow": true,
  "dataLabelsShow": true,
  "showYAxis": true,
  "optionalGridLines": [],
  "optionalYGridLines": [],
  "showDataPoint": false,
  "dataOrder": null
},
},
"numerical": {
  "xAxisLabel": "x_axis",
  "yAxisLabel": "kpi",
  "legendPosition": "bottom",
  "showLegend": true,
  "axesRotate": false,
  "xCulling": 10,
  "xTickCount": 5,
  "xGridShow": true,
  "yGridShow": true,
  "dataLabelsShow": false,
  "showYAxis": true,
  "optionalGridLines": [],
  "optionalYGridLines": [],
  "showDataPoint": true,
  "dataOrder": null
}
```

## E Synthetic data-set

Region	Country	Item type	Sales Channel	Order priority	Order date	Ship date	Units sold	Unit price	Unit cost	Total revenue	Total cost	Total profit
Asia	Mongolia	Meat	Online	L	31.01.2017	02.03.2017	4121	421.89	364.69	1738608.69	1502887.49	235721.20
Europe	Germany	Baby Food	Offline	L	06.01.2017	15.02.2017	9061	255.28	159.42	2313092.08	1444504.62	868587.46
Australia and Oceania	Samoa	Cereal	Offline	C	23.01.2017	25.01.2017	5840	205.70	117.11	1201288.00	683922.40	517365.60
Central America and the Caribbean	The Bahamas	Baby Food	Offline	H	13.01.2017	16.01.2017	7119	255.28	159.42	1817338.32	1134910.98	682427.34
Sub-Saharan Africa	Ghana	Baby Food	Offline	M	20.01.2017	06.02.2017	5177	255.28	159.42	1321584.56	825317.34	496267.22
...	...	...	...	...	...	...	...	...	...	...	...	...

Table E.1: Subset of the Country sales data-set

## F Data-set

Year	Gender	Number
2009	male	430
2008	male	420
2011	male	410
2010	male	440
2012	male	425
2012	female	75
2011	female	90
2010	female	60
2008	female	80
2013	female	68

Table F.1: Overview of number of males and females in years

# G Chart suggestions

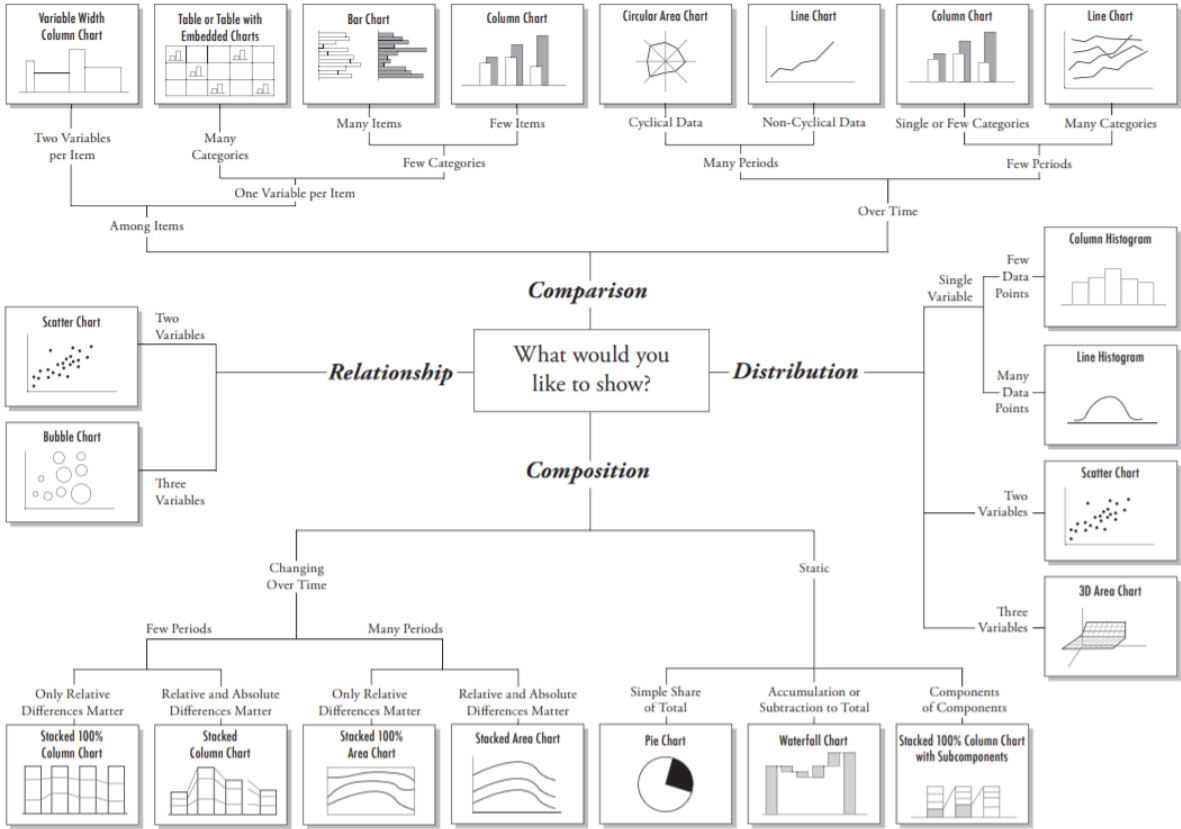


Figure G.1: Chart selector developed by Dr. Abel [KNA13]



# H date-time formats

D - Day (8); DD - Day (28)

M - Month (2); MM - Month (12)

YY - Year (19); YYYY - Year (2019)

- D.M.YY
- M.D.YY
- YY.M.D
- YY.D.M
- D.M
- M.D
- M.YY
- YY.M
- YY
- M
- D.M.YYYY
- M.D.YYYY
- YYYY.M.D
- D.M
- M.D
- M.YYYY
- YYYY.M
- YYYY
- M
- D.MM.Y
- MM.D.YY
- YY.MM.D
- YY.D.MM
- D.MM
- MM.D
- MM.YY
- YY
- MM
- D.MM.YYYY
- MM.D.YYYY
- YYYY.MM.D
- YYYY.MM.D
- MM.YYYY
- YYYY.MM
- DD.M.YY
- M.DD.YY
- YY.M.DD
- YY.DD.M
- DD.M
- M.DD
- YY.MM
- YYYY.D.M
- M.DD.YYYY
- YYYY.M.DD
- YYYY.DD.M
- DD.MM.YY
- MM.DD.YY
- YY.MM.DD
- YY.DD.MM
- DD.MM
- MM.DD
- MM.YY
- ...

The list continues with the same constructs but with different separators: comma, slash or without any. The complete list contains 408 date-time formats.

# Bibliography

- [Agr13] Alan Agresti. *Categorical Data Analysis (3 ed.)* John Wiley & Sons, 2013. ISBN: 978-0-470-46363-5 (cit. on p. 9).
- [Bha19a] Adi Bhat. *INTERVAL DATA: DEFINITION, CHARACTERISTICS AND EXAMPLES*. 2019. URL: <https://www.questionpro.com/blog/interval-data/> (visited on 06/21/2019) (cit. on p. 9).
- [Bha19b] Adi Bhat. *QUANTITATIVE DATA: DEFINITION, TYPES, ANALYSIS AND EXAMPLES*. 2019. URL: [https://www.questionpro.com/blog/quantitative-data/#Quantitative\\_Data\\_Definition](https://www.questionpro.com/blog/quantitative-data/#Quantitative_Data_Definition) (visited on 06/21/2019) (cit. on p. 8).
- [BA96] J.M. Bland and D.G. Altman. *Statistics notes: measurement error*. *BMJ*, 1996, p. 744 (cit. on p. 14).
- [Cai16] Alberto Cairo. *The truthful art*. New Riders, 2016. ISBN: 0321934075 (cit. on pp. 1, 4, 27).
- [CM84] William S. Cleveland and Robert McGill. “Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods”. In: *Journal of the American Statistical Association*. 1984, pp. 531–554 (cit. on pp. 25, 58).
- [CSP96] Ronald Jay Cohen, Mark E Swerdik, and Suzanne M. Phillips. *Psychological Testing and Assessment: An Introduction to Tests and Measurement (3rd ed.)*. Mountain View, CA: Mayfield, 1996, p. 685. ISBN: 1-55934-427-X (cit. on p. 9).
- [Cor19] Michael Correll. “Ethical Dimensions of Visualization Research”. In: *Proceedings of the 2019 ACM annual conference on Human Factors in Computing Systems*. 2019, p. 1 (cit. on p. 17).

## Bibliography

- [ESC18] Humaira Ehsan, Mohamed A. Sharaf, and Panos K. Chrysanthis. “Efficient Recommendation of Aggregate Data Visualizations”. In: *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*. 2018, pp. 263–277 (cit. on pp. 1, 15, 42).
- [Eve17] Stephanie D.H. Evergreen. *Effective Data Visualization. The right chart for the right data*. SAGE Publications, Inc, 2017. ISBN: 9781506303055 (cit. on pp. 2, 4, 5, 27, 30–32, 58, 59).
- [Few16] Stephan Few. *The Encyclopedia of Human-Computer Interaction, 2nd Ed.* 2016. URL: <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/data-visualization-for-human-perception> (visited on 04/18/2019) (cit. on p. 20).
- [Few05] Stephen Few. *Effectively Communicating Numbers - Selecting the Best Means and Manner of Display*. Tech. rep. Principal, Perceptual Edge, 2005 (cit. on pp. 7, 8, 10).
- [Few12] Stephen Few. *Show me the numbers. Designing Tables and Graphs to Enlighten*. Brian Pierce, 2012. ISBN: 9780970601971 (cit. on pp. 11, 18, 20, 22).
- [GAN19] AKHILESH GANTI. *Correlation Coefficient*. 2019. URL: <https://www.investopedia.com/terms/c/correlationcoefficient.asp> (visited on 06/21/2019) (cit. on p. 11).
- [How+19] Cindi Howson et al. *Magic Quadrant for Analytics and Business Intelligence Platforms*. 2019. URL: <https://www.gartner.com/doc/reprints?id=1-68720FP&ct=190213&st=sb> (visited on 03/23/2019) (cit. on pp. 35, 37).
- [Jac94] Harold R. Jacobs. *Mathematics: A Human Endeavor (Third ed.)* W. H. Freeman, 1994, p. 547. ISBN: 0-7167-2426-X (cit. on p. 13).
- [KNA13] COLE NUSSBAUMER KNAFLIC. *chart chooser*. 2013. URL: <http://www.storytellingwithdata.com/blog/2013/04/chart-chooser> (visited on 06/23/2019) (cit. on pp. 32, 134).
- [Kna15] Cole Nussbaumer Knaffic. *Storytelling with data*. Wiley, 2015, pp. 43–70. ISBN: 9781119002260 (cit. on pp. 18, 19, 29).

## Bibliography

- [KW18] Kuno Kurzhals and Daniel Weiskopf. “Exploring the Visualization Design Space with Repertory Grids”. In: Eurographics Conference on Visualization (EuroVis) 2018, 2018 (cit. on pp. 41, 42).
- [Luo+18] Yuyu Luo et al. “DeepEye: Towards Automatic Data Visualization”. In: 2018 IEEE 34th International Conference on Data Engineering (ICDE), 2018 (cit. on pp. 1, 2, 43).
- [Par16] Noah Parsons. *Scientific Reasons Why You Should Present Your Data Visually*. 2016. URL: <https://medium.com/lighting-out/scientific-reasons-why-you-should-present-your-data-visually-7f57dcf6110f> (visited on 06/22/2019) (cit. on p. 22).
- [Pea16] Teagan Pease. *Data Visualization with Exploratory Vol. 1 - Introduction*. 2016. URL: <https://blog.exploratory.io/introduction-to-data-visualization-vol-1-introduction-88112157a8fb> (visited on 06/22/2019) (cit. on p. 24).
- [Pow18] Anna Powell-Smith. *How to choose the right visualization for your data*. 2018. URL: <https://flourish.studio/2018/09/28/choosing-the-right-visualisation/> (visited on 04/14/2019) (cit. on p. 29).
- [Rue19] Jeremy Rue. *Visualizing Data: A Guide to Chart Types*. 2019. URL: <https://multimedia.journalism.berkeley.edu/tutorials/visualizing-data-a-guide-to-chart-types/> (visited on 06/22/2019) (cit. on p. 18).
- [RP06] Gordon Rugg and Marian Petre. *A Gentle Guide To Research Methods*. McGraw-Hill International, 2006, pp. 18–183. ISBN: 9780335219278 (cit. on p. 8).
- [SI11] Julie Steele and Noah Iliinsky. *Designing Data Visualizations*. O’Reilly Media, Inc., 2011. ISBN: 9781449314774 (cit. on pp. 23, 24, 29).
- [Ste18] Jay Stevenson. *What Is Data Visualization? Definition, History, and Examples*. 2018. URL: <https://www.anychart.com/blog/2018/11/20/data-visualization-definition-history-examples/> (visited on 04/14/2019) (cit. on p. 25).
- [Str19] Jeffrey Strickland. *What is Time Series Analysis?* 2019. URL: <https://www.linkedin.com/pulse/what-time-series-analysis-jeffrey-strickland-ph-d-cmsp> (visited on 06/21/2019) (cit. on p. 10).

## Bibliography

- [Tay14] Twain Taylor. *To Explain, or Explore: That is the Question in Data Visualization*. 2014. URL: <https://www.fusioncharts.com/blog/to-explain-or-explore-that-is-the-question-in-data-visualization-podv/> (visited on 04/12/2019) (cit. on pp. 23, 24).
- [Tom19] Kyjean Tomboc. *Data Visualization Guide: Choosing the Right Chart to Visualize Your Data*. 2019. URL: <https://www.easel.ly/blog/types-of-graphs-and-charts-for-visualizing-data/> (visited on 07/08/2019) (cit. on p. 7).
- [Tuf83] Edward Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 1983, p. 107 (cit. on p. 58).
- [Var+15] Manasi Vartak<sup>1</sup> et al. “SEEDB: Efficient Data-Driven Visualization Recommendations to Support Visual Analytics”. In: *Proceedings VLDB Endowment*. 2018 IEEE 34th International Conference on Data Engineering (ICDE), 2015 (cit. on pp. 1, 44, 45).
- [Vol08] Dan Volitich. *IBM Cognos 8 Business Intelligence*. McGraw Hill, 2008. ISBN: 978-0-07-149852-4 (cit. on p. 37).
- [War13] Colin Ware. *Information visualization: Perception for design*. Elsevier Inc, 2013. ISBN: 1-55860-819-2 (cit. on pp. 20, 27).
- [Yau13] Nathan Yau. *Data points visualization that means something*. John Wiley and Sons, Inc, 2013. ISBN: 9781118462195 (cit. on p. 27).

# Eidesstattliche Erklärung

Hiermit versichere ich, dass ich diese Master thesis selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe und alle Ausführungen, die wörtlich oder sinngemäß bernommen wurden, als solche gekennzeichnet sind, sowie, dass ich die Master thesis in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt habe.

Koblenz, September 23, 2019

---

Slobodan Kocevski