Knowledge at first sight: Building a model for a data visualization recommender system suited for non-expert users

Petra Kubernátová
Graduation Thesis, August 2017
Media Technology MSc program, Leiden University
Thesis advisors: Max van Duijn and Magda Friedjungová
pkubernatova@gmail.com

Abstract— **In today's age, there are huge amounts of data being generated every second of every day. Through data visualization, humans can explore, analyse and present it. Choosing a suitable visualization for data is a difficult task, especially for non-experts. Current data visualization recommender systems exist to aid in choosing a visualization, yet suffer from issues such as low accessibility and indecisiveness. The aim of this study is to create a model for a data visualization recommender system for non-experts that resolves these issues. Based on existing work and a survey among data scientists, requirements for a new model were identified and implemented. The result is a question-based model that uses a decision tree and a data visualization classification hierarchy in order to recommend a visualization. Furthermore, it incorporates both task-driven and data characteristics-driven perspectives, whereas existing solutions seem to either convolute these or focus on one of the two exclusively. Based on testing of the model against existing solutions, it is shown that the new model reaches similar results while being simpler, clearer, and more versatile, extendable and transparent. In the future, the presented model can be applied in the development of new data visualization software or as part of a learning tool.**

Key words—data science, data visualization, recommender systems, non-expert users

I. Introduction

In today's age, there are huge amounts of data being generated every second of every day and Big Data has been one of the hot topics of computer science in recent years. Being the curious species that we are, humans are looking for ways to get the most information out of this vast amount of data that we have available at our fingertips. We are always looking for methods to help us explore, analyze and present it.

A crucial part of this process is data visualization. Data visualization is the representation of information in a visual form, such as a chart, diagram or picture. It can find its place in a variety of areas such as art, marketing, social relations and scientific research. There were over 300 visualization types available at the time of writing this paper [1]. But how do we choose the most suitable one? This is where data visualization recommender systems come in: these systems help with this difficult task that becomes even more difficult when the user is a non-expert.

In this paper we define a 'non-expert user' as someone without professional or specialized knowledge of data visualization. We thus include both complete beginners and users who have general knowledge of data visualization types (e.g. bar charts, pie charts, scatter plots) but have no professional experience in the fields of data science and data communication.

In this study we focus on building a model for a data visualization recommender system aimed at non-expert users. We term our model NEViM: Non-Expert Visualization Model.

In Section II of this paper, we place data visualization recommender systems for non-experts in the context of data science. We discuss different types of systems and comment on where the model we are building fits in. Section III introduces our research aim and hypothesis. In Section IV we outline our method, which consists of several parts. We start with a literature review providing background, we analyze existing solutions and their history and perform an exploratory user survey with 88 participants. Based on the results of these, we put together requirements for our model. We construct a model incorporating our findings as well as findings from 20 books about data visualization. Finally, we perform two tests on our model. The first tests the model's ability to compete against existing solutions on 10 different data sets and the second tests its extensibility. Section V discusses the results of the work done within our method. We present results of our literature study, existing solutions analysis, survey, model requirements, model construction process and model testing process. We draw conclusions in Section VI and set an agenda for future work in Section VII.

II. Context

A. Data science

Data science plays an important role in scientific research, as it aids us in collecting, organizing, and interpreting data, so that it can be transformed into valuable knowledge.

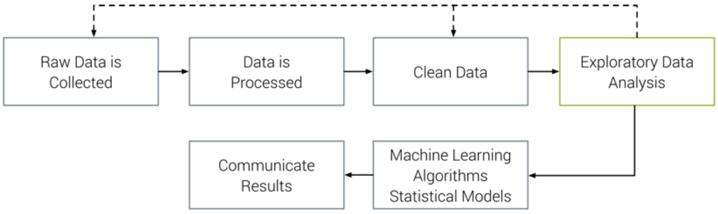

Raw Data is Collected
Data is Processed
Clean Data
Exploratory Data Analysis
Communicate Results
Machine Learning Algorithms Statistical Models

Fig. 1 The data science process [2]

Figure 1 shows a simplified diagram of the data science process as described by O'Neil and Schutt [2]. This diagram is helpful in demarcating the research objectives of this paper. According to O'Neil and Schutt, first real world raw data is collected, processed and cleaned through a process called data munging. Then exploratory data analysis (EDA) follows, during which we might find that we need to collect more data or dedicate more time to cleaning and organizing the current dataset. When finished with EDA, we may use machine learning algorithms, statistical models and data visualization techniques, depending on the type of problem we are trying to solve. Finally, results can be communicated [2].

Our focus here is on the part of the process concerning exploratory data analysis or EDA. EDA uses a variety of statistical techniques, principles of machine learning, but also, crucially, the data visualization techniques we study in this paper. Please note that data visualization can also be a part of the "Communicate Results" stage of the data science process (see Figure 1). There is a thin line between data visualizations made for *exploration* and ones made for *explanation*, as most exploratory data visualizations also contain some level of explanation and vice-versa.

B. Exploratory data analysis

Exploratory data analysis (EDA) is not only a critical part of the data science process, it is also a kind of philosophy. EDA does not yet revolve around a specific model or hypothesis. Your understanding of the problem is changing and evolving as you go. You are aiming to understand the data and its shape, then connect your understanding of the process that collected the data with the data itself [3]. EDA helps with suggesting hypotheses to test, evaluating the quality of the data, identifying potential need for further collection or cleaning, supporting the selection of appropriate models and techniques and, most importantly for the context of this study, it helps find interesting insights in your data [3].

C. Data visualization

There are many definitions of the term **data visualization**. The one used in this study is: data visualization is the representation and presentation of data to facilitate understanding [4]. According to Kirk, our eye and mind are not equipped to easily translate the textual and numeric values of raw data into quantitative and qualitative meaning. "We can look at the data, but we cannot understand it. To truly understand the data, we need to see it in a different kind of form. A visual form." [4]

According to Illinsky and Steele, data visualization is a very powerful tool for identifying patterns, communicating relationships and meaning, inspiring new questions, identifying sub-problems, identifying trends and outliers, discovering or searching for interesting or specific data points [5].

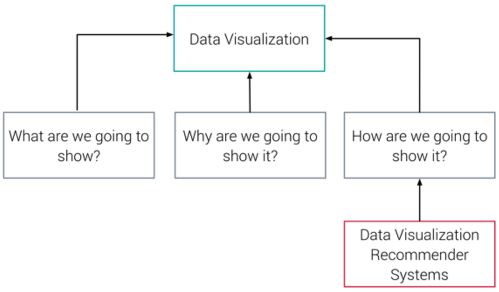

Data Visualization
What are we going to show?
Why are we going to show it?
How are we going to show it?
Data Visualization Recommender Systems

Fig. 2 Munzner's model for data visualization design with indication of where the topic of this study belongs [6]

Tamara Munzner made a 3-step model for data visualization design, depicted in Figure 2 [6]. According to this model, we first need to decide what we want to show. Secondly, we need to motivate why we want to show it. Finally, we need to decide how we are going to show it. There are many different types of data visualizations to help us with the third step. However, the challenge remains in choosing the most suitable one. Data visualization recommender systems were made to help with this difficult task. We find that the WHAT and the WHY greatly influence the HOW, thus we aim to build a system that reflects all three aspects of the data visualization design process in some way.

D. Data visualization recommender systems

Within this study we define **data visualization recommender systems** as tools that seek to recommend visualizations that highlight features of interest in data. This definition is based on combining common aspects of definitions in existing work.

While the output of data visualization recommender systems is always a recommendation for data visualization types in some shape or form, the input can differ. It can be, for example, just the data itself, a specification of goals or the specification of aesthetic preferences. The type of input affects the type of recommendation strategy used and consequently the type of the recommender system.

Kaur and Owonibi distinguish 4 types of recommender systems [7]:

- **Data Characteristics Oriented**: These systems recommend visualizations based on data characteristics.
- **Task Oriented**: These systems recommend visualizations based on representational goals as well as data characteristics.
- **Domain Knowledge Oriented**: These systems improve the visualization recommendation process with domain knowledge.
- **User Preferences Oriented**: These systems gather information about the user presentation goals and preferences through user interaction with the visualization system.

The line between different categories of recommendation systems is rather thin and some systems can have ambiguous classifications, as will be discussed in Section V below.

III. Research aims and hypothesis

Within this study our aim is to devise a new data visualization recommender system, which is simple and easy to use for non-experts, but can nonetheless compete with existing, often more complex systems. Clearly, we will avoid "reinventing the wheel": the current solutions are already good, but we want to see if we can make adjustments that make a system more suitable for non-expert users while maintaining effectiveness (still clearly distinguishing the data visualizations from each other) and performance (recommending the most suitable visualization type). For this, we will combine aspects of different kinds of recommendation systems into one. Also, we incorporate insights from an exploratory survey among 88 users and from 20 existing handbooks. After implementing these aspects and insights into our model NEViM, we test it on 10 example datasets against existing and widely available solutions.

Our hypothesis, thus, is that NEViM, while remaining simple and straightforward enough to be used by non-experts, can compete on effectiveness and performance with Tableau, Watson Analytics, Microsoft Excel Recommended Charts, Voyager and Google Sheets.

IV. Method

A. Existing solutions study

Our first step is a literature study of previous work done in the field of data visualization recommender systems. We focus on data characteristics-oriented and task-oriented data visualization recommender systems, as this is where our model belongs. We introduce each system and explain which aspects of it we incorporate in NEViM. We also determine which currently existing solutions are suitable for the testing of our model.

B. Survey

We run a survey among different data science communities on Facebook and LinkedIn. This way, we ask 88 respondents who have some sort of familiarity with data science and its terminology. The main goals of the survey are to aid us in decisions about our model and, as our model is aimed at non-expert users, to aid us in specifying who exactly these users are.

C. Model requirements

The findings we make from the literature study, as well as the results of the survey help us form requirements for our model.

D. Constructing the model

Once we have the requirements, we commence constructing the model. First we choose a suitable base structure. Then we establish the different components of the structure and specify what they will be in our model. Finally, we combine it all together into a model.

E. Testing the model

We perform two tests on the constructed model. The first test focuses on establishing whether the model is able to produce results similar or identical to existing solutions. The second focuses on testing the extendibility of the model by adding a new type of visualization.

V. Results

A. Existing solutions study

1) Data Characteristics Oriented systems

Systems based on data characteristics aim to improve the understanding of the data, of different relationships that exist within the data and of procedures to represent them. Some of the following tools and techniques are not recommendation systems per se but they were a crucial part of the history of this field and foundations for other recommender systems stated, thus we feel it is appropriate to list them as well.

BHARAT

BHARAT was the first system that proposed some rules for determining which type of visualization is appropriate for certain data attributes [8]. As this work was written in 1981, the set of possible visualizations was not as varied as it is today. The system incorporated only the line, pie and bar charts and was based on a very simple design algorithm. If the function was continuous, a line chart was recommended. If the user indicated that the range sets could be summed up to a meaningful total, a pie chart was recommended and bar charts were recommended in all the remaining cases. Even though this system would now be considered very basic, it served as the basis for other systems that followed.

APT

In 1986, Mackinlay proposed to formalize and codify the graphical design specification to automate the graphics generation process [9]. His work is based on the work of Joseph Bertin, who, in 1983, came up with a semiology of graphics [10], where he specified visual variables such as position, size, value, color, orientation etc. and classified them according to which features they communicate best. For example, the shape variable is best used to show differences and similarities between objects. Mackinlay codified Bertin's semiology into algebraic operators that were used to search for effective presentations of information. He based his findings on the principals of expressiveness and effectiveness. Expressiveness is the idea that graphical presentations are actually sentences of graphical languages that have precise syntactic and semantic definitions, while effectiveness refers to how accurately these presentations are perceived. He aimed to develop a list of graphical languages that can be filtered with the expressiveness criteria and ordered with the effectiveness criteria for each input. He would take the encoding technique and formalize it with primitive graphical language (which data visualizations can show this), then he would order these primitive graphical languages using the effectiveness principle (how accurately perceived they are). APT's architecture was focused on how to communicate graphically rather than on what to say. Casner extended this work by comparing design alternatives via a measure of the work that was required to read presentations, depending on the task [11]. Roth and Mattis added additional types of visualizations [12] to this system.

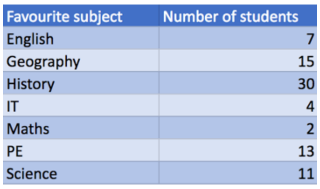
VizQL(Visual Query Language)

In 2003, Hanrahan revised Mackinglay's specifications into a declarative visual language known as VizQL [13]. It is a formal language for describing tables, charts, graphs, maps, time series and tables of visualizations. The language is capable of translating actions into a database query and then expressing the response graphically.

The discovery of VizQL gave us ideas on how to annotate data visualizations which could be useful for the backend part of a possible implementation of our model.

Tableau and its Show Me Feature

The introduction of Tableau was a real milestone in the world of data visualization tools. Due to the simple user interface, even inexperienced users could create impressive and informative data visualizations. It was created when Stolte, together with Hanrahan and Chabot, decided to commercialize Polaris [14] under the name Tableau Software, creating the most popular data visualization tool. Tableau offers an intuitive user experience. Let's say you want to draw a bar chart, all you have to do is specify a data source and then drag the data attributes you want to display in the column and row section. In Figures 3 and 4, the input and an example of output given by Tableau is shown.



Favourite subject	Number of students
English	7
Geography	15
History	30
IT	4
Maths	2
PE	13
Science	11

Fig. 3 Data which served as input into the interface in Fig. 4

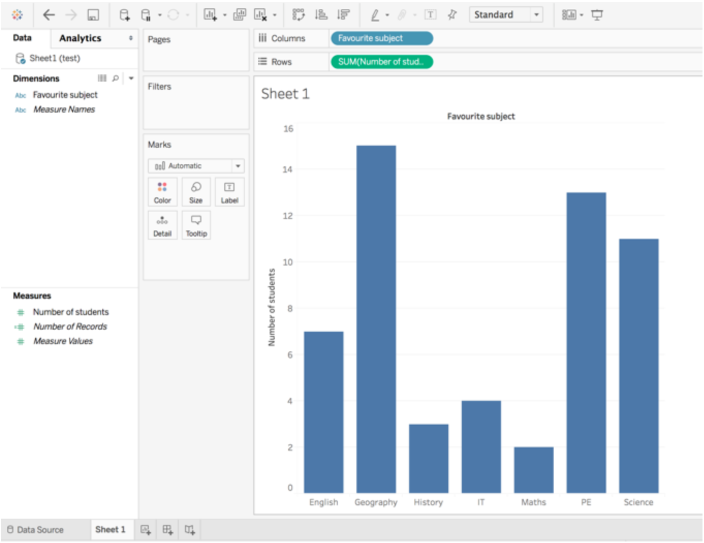


Fig. 4 Example of Tableau user interface. The aim was to make a bar chart of favourite subjects within a class of students. The data can be seen in Fig. 3.

In 2004, Mackinglay joined the Tableau team and helped them develop a feature called Show Me which was introduced in 2007 [15]. The Show Me functionality takes advantage of VizQL to automatically present data. At the heart of this feature is a data characteristics-oriented recommendation system. The user selects the data attributes that interest him and

Tableau recommends a suitable visualization. Tableau determines the proper visualization type to use by looking at specific attributes in the data. Each visualization requires specific attributes to be present before it can be recommended. Furthermore, it also ranks every visualization on familiarity and design best practices. Finally, it recommends the highest-ranked eligible visualization. Table I shows the underlying classification table of Tableau. Since 2007, the list of data visualizations has been expanded, but information about these new visualizations is not available.

Table I. Classification table used by Tableau [15]. In this case, attributes are synonymous to columns. A quantitative attribute represents a measurable quantity, for example the population of a city. A categorical attribute takes on values that are names or labels, for example the breed of a dog (shepherd, collie, terrier).

Data Visualization	***Condition***	***Rank***
Table	At least 1 attribute	1
Aligned Bars	At least 1 quantitative attribute	2
Stacked Bars	At least 2 categorical attributes, at least 1 quantitative attribute	3 with at least 3 categorical attributes
Discrete Lines	At least 1 categorical date attribute, at least 1 quantitative attribute	4
Scatter Plot	Between 2 – 4 quantitative attributes	5 with at least 2 quantitative attributes
Gantt Chart	At least 1 categorical attribute, at least 1 quantitative independent attribute, between 1-2 quantitative attributes	6

Mackinglay and his team have also performed interesting user tests with the Show Me feature. They tested it with new users as well as skilled ones. They created a mechanism which collects logs about Tableau user interface activity and stores it on their computer. Since they established that a typical Tableau user is a professional adult working with corporate data, they wanted to see if these type of skilled users were going to use the Show Me feature. They found that the Show Me feature is being used (very) modestly by skilled users (i.e. in only 5.6% of cases).

Tableau inspired us by it's simple user interface which is suitable for non-experts, reminding us that our model should enable a simple user interface implementation. Furthermore, we make use of their classification of data visualizations based on design best practices and familiarity as well as the conditions that the data must fulfill for a specific data visualization to be chosen. The fact that Tableau is so widely used and that a demo version is freely accessible determined it suitable for use in our tests.

ManyEyes

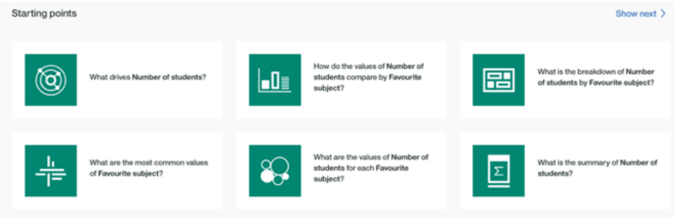
Viegas et al. created the first known public website where users may upload data and create interactive visualizations collaboratively: ManyEyes [16]. The tool was created for non-experts, as Viegas et al. wanted to make a tool that was accessible for anyone regardless of prior knowledge and training. Design choices were made to reflect the effort to find a balance between powerful data-analysis capabilities and accessibility to the non-expert visualization user. The visualizations were created by matching a dataset with one of the 13 types of data visualizations implemented in the tool. To set up this matching, the visualization components needed to be able to express its data needs in a precise manner. They divided the data visualizations into groups by data schemas. A data schema could be, for example, "single column textual data". Thus, a bar chart was described as "single column textual data and more than one numerical value". The dataset and produced visualization could then be shared with others for comments, feedback and improvement [16]. However, the tool closed down in 2015.

ManyEyes taught us that the way to attract non-expert users is to make the application resulting from our model as accessible as possible. This means that our model is suited to web-based implementations.

Watson Analytics

Since 2014, IBM have been developing a tool called Watson Analytics [17]. It carries the same name as another successful IBM project – the Watson supercomputer, which combines artificial intelligence and sophisticated analytical software to perform as a "question-answering" system. In 2011, it famously defeated top-ranked players in a game of Jeopardy!. Similarly to the Watson supercomputer, Watson Analytics uses principles of machine learning and natural language processing to recommend users either questions they can ask about their data, or a specific visualization. However, IBM has not revealed what values or attributes are used by the recommendation system to select a visualization. A demo version of the system is freely available, so we use it in our tests.

Watson Analytics reminded us that the structure of our model should be variable enough to be suitable for implementing machine learning and artificial intelligence techniques on it for the model to possibly improve itself.



What drives **Number of students**?

How do the values of **Number of students** compare by **Favourite subject**?

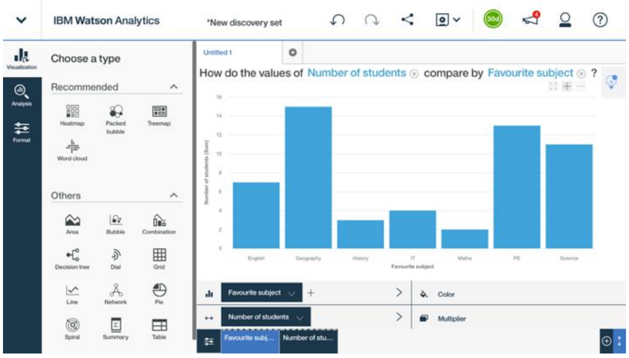
What is the breakdown of **Number of students** by **Favourite subject**?

What are the most common values of **Favourite subject**?

What are the values of **Number of students** for each **Favourite subject**?

What is the summary of **Number of students**?

Fig. 5 Example data analysis questions generated by IBM Watson Analytics

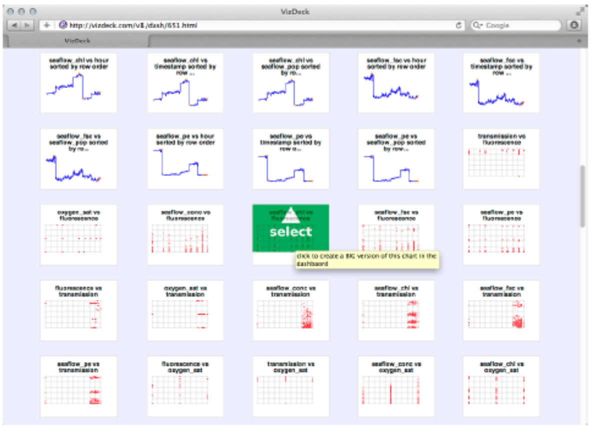


How do the values of Number of students compare by Favourite subject ?

Fig. 6 Example of IBM Watson Analytics interface. The task was the same as in Figure 4., we can also see which other visualizations were recommended

VizDeck

In 2012 Key et al. developed a tool called VizDeck [18]. The web-based tool recommends visualizations based on statistical properties of the data. It adopts a card game metaphor to organize multiple visualizations into an interactive visual dashboard application. Vizdeck was created as Key et al. found that scientists were not able to self-train quickly in more sophisticated tools such as Tableau. The tool supports scatter plots, histograms, bar charts, pie charts, timeseries plots, line plots and maps.



VizDeck
http://vizdeck.com/v1/dash/651.html
select
click to create a BIG version of this chart in the dashboard

Fig. 7 Example of the user interface of VizDeck dealing a “hand” of visualization recommendations [18].

Based on the statistical properties of the underlying dataset, VizDeck generates a “hand” of ranked visualizations and the user chooses which “cards” to keep and put into a dashboard and which to discard. Through this, the system learns which visualizations are appropriate for a given dataset and improves the quality of the “hand” dealt to future users. For the actual recommendation system part of the tool, they trained a model of visualization quality that relates statistical features of the dataset to particular visualizations. As far as we know VizDeck was never actually deployed and remained at the testing phase.

VizDeck again inspired us to think about the possibility of our model being self-improving and educative.

Microsoft Excel’s Recommended Charts Feature

In the 2013 release of Microsoft Excel, a new feature called Recommended Charts was introduced. The user can select the data they want to visualize and Excel recommends a suitable visualization [19]. However, Microsoft does not share exactly how this process is carried out, making it less suitable as a source of inspiration.

We use Microsoft Excel to test our model, because it is accessible.

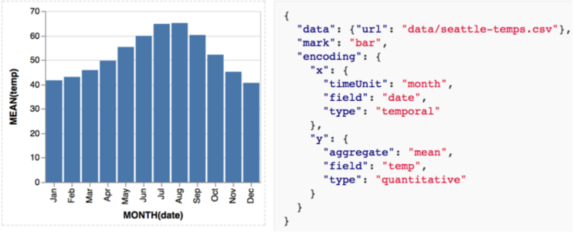
SEEDB

In 2015 Vartak et al. proposed an engine called SEEDB [20]. They judge the interestingness of a visualization based on the following theory: a visualization is likely to be interesting if it displays large deviations from some reference (e.g. another dataset, historical data, or the rest of the data). This helps them identify the most interesting visualizations from a large set of potential visualizations. They identified that there are more aspects that determine the interestingness of a visualization, such as aesthetics, user preference, metadata and user tasks. A full-fledged visualization recommendation system should take into account a combination of these aspects. A major disadvantage of SEEDB is that it only uses variations of bar charts and line charts. As far as we know SEEDB was never deployed.

SEEDB made us think about having multiple views in our model from different interestingness perspectives, because we want our model to be full-fledged, as they describe.

Voyager

In 2016, Wongsuphasawat et al. developed a visualization recommendation web application called Voyager [21], based on the Compass recommendation engine [22] and a high-level specification language called Vega-lite [23]. It couples browsing with visualization recommendation to support exploration of multivariate, tabular data. Vega-lite specifications consist of a set of mappings between visual encoding channels and data variables. The output is a JSON object that describes a single data source (data), a mark type (marktype), key-value visual encodings of data variables (encoding) and data transformations including filters (filter) and aggregate functions. We can see an example of a Vega-lite specification in Figure 8.



```
{
  "data": {"url": "data/seattle-temps.csv"},
  "mark": "bar",
  "encoding": {
    "x": {
      "timeUnit": "month",
      "field": "date",
      "type": "temporal"
    },
    "y": {
      "aggregate": "mean",
      "field": "temp",
      "type": "quantitative"
    }
  }
}
```

Fig. 8 Example of Vega-Lite specification to create a bar chart that shows the average temperature in Seattle for each month [23]

First, Compass selects variables by taking user-selected variable sets and suggesting additional variables. It then applies data transformations (e.g. aggregation or binning) to produce a set of derived data tables. For each data table, it designs encodings based on expressiveness and effectiveness criteria and prunes visually similar results to avoid exhaustive enumeration. The user then includes or excludes different variables to focus on a particular set of variables that are interesting.

Voyager is a tool which is freely available online, which makes it suitable for use in our tests.

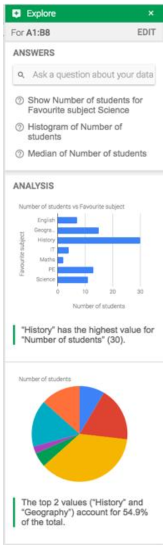
Google Sheets and its Explore feature

Google Sheets [24] is a tool which allows users to create, edit and share spreadsheets. It was introduced in 2007 and is very similar to Microsoft Excel. In June of 2017, the tool was extended with the Explore Feature, which helps with automatic chart building and data visualization. It uses elements of artificial intelligence and natural language processing to recommend users questions they might want to ask about their data, as well as recommending data visualizations that best suit their data. In the documentation for this feature, Google specifies each of the included data visualizations by functions and conditions that have to be fulfilled in order for that particular data visualization to be recommended. However, it does not reveal exactly how it chooses the most suitable data visualization, because as can be seen in Table II, a couple of visualizations have the same conditions. A minor downside of Google Sheets is that one needs a Google account to use it.

We make use of the classification of data visualizations presented in Google Sheets and thanks to its accessibility online, we use it in our tests.

Table II. Classification table used by Google Sheets [24]

Visualization	**Conditions**	**Function**
Line Chart	Column 1 – label Other columns - numeric	Look at trends within data or data over time period.
Column Chart	Column 1 – label for each row Other columns – numeric	Show one or more categories or groups of data especially if each category has subcategories
Bar Chart	Column 1 – label for each row Other columns – numeric	Difference between data points for one or more category
Pie Chart	Column 1 – label Column 2 – numeric	Proportions of a whole
Scatter Plot	Column 1 – x-axis value Other columns – y-axis values, each column displayed as series of data points	Show numeric coordinates, show trends and patterns between two variables



Show Number of students for Favourite subject Science

Histogram of Number of students

Median of Number of students

"History" has the highest value for "Number of students" (30).

The top 2 values ("History" and "Geography") account for 54.9% of the total.

Fig. 9 Example of the Google Sheets Explore feature, task and data were the same as in Figure 4.

2) *Task Oriented systems*

Task-oriented systems aim to design different techniques to infer the representational goal or a user's intentions. In 1990 Roth and Mattis were the first to identify different domain-independent information seeking goals, such as comparison, distribution, correlation etc. [12]. Also in 1990, Wehrend and Lewis proposed a classification scheme based on sets of representational goals [25]. It was in the form of a 2D matrix where the columns were data attributes, the rows representational goals and the cells data visualizations. To find a visualization, the user had to divide the problem into subproblems, until for each subproblem it was possible to find an entry in the matrix. A representation for the original complex problem could then be found by combining the candidate representation methods for the subproblems. Unfortunately, the complete matrix was not published so it is unknown which specific types of data visualizations were included.

BOZ

BOZ is an automated graphic design and presentation tool that designs graphics based on an analysis of the task which a graphic is intended to support [11]. The system analyzes a logical description of a task to be performed and designs an equivalent perceptual task. BOZ produces a graphic along with a perceptual procedure describing how to use the graphic to complete the task. It is able to design different presentations of the same information customized to the requirements of different tasks.

The BOZ system reminded us that the difference between a suitable and non-suitable data visualization could also lie in the way that humans perceive them. For example, a pie chart is generally considered not suitable, as humans have difficulty judging the size of angles accurately.

IMPROVISE

In the previous studies, the user task list was manually created. However, in 1998, Zhou and Feiner introduced advanced linguistic techniques to automate the derivation of the user task from a natural language query [26]. They introduced a visual task taxonomy to automate the process of gaining presentation intents from the text. The taxonomy interfaces between high level tasks that can be accomplished by low level visualization techniques. For example, the visual task "Focus" implies that visual techniques such as "Enlarge" or "Highlight" could be used. This taxonomy is implemented in IMPROVISE.

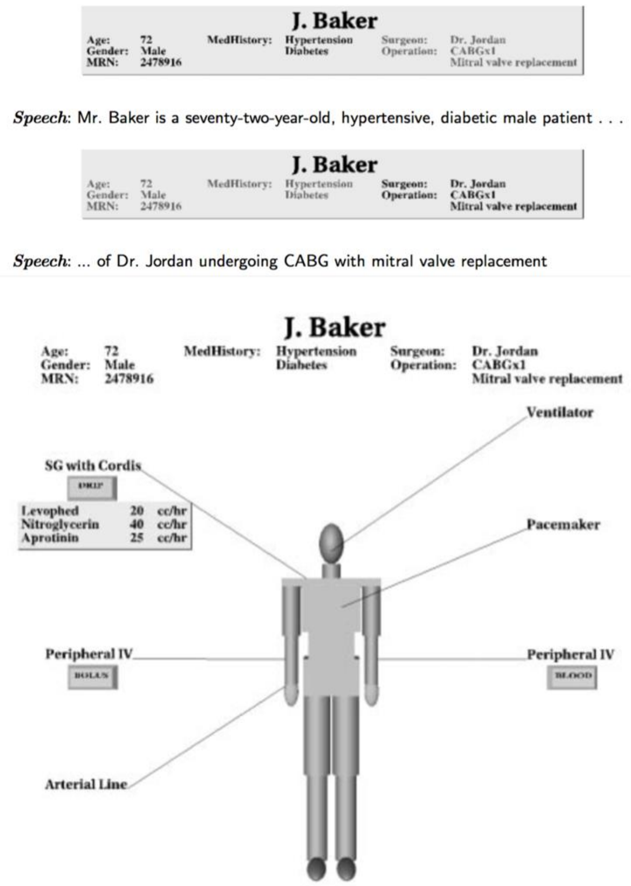

J. Baker
Age: 72
Gender: Male
MRN: 2478916
MedHistory: Hypertension Diabetes
Surgeon: Dr. Jordan
Operation: CABGx1 Mitral valve replacement
Speech: Mr. Baker is a seventy-two-year-old, hypertensive, diabetic male patient . . .
Speech: ... of Dr. Jordan undergoing CABG with mitral valve replacement
Ventilator
SG with Cordis
DRIP
Levophed 20 cc/hr
Nitroglycerin 40 cc/hr
Aprotinin 25 cc/hr
Pacemaker
Peripheral IV
BOLUS
Peripheral IV
BLOOD
Arterial Line

Fig. 10 Example of IMPROVISE [26].

In Figure 10 we can see an example of how IMPROVISE generates a visual narrative from speech to present an overview of a hospital patient's information to a nurse. To achieve this goal, it constructs a structure diagram that organizes various information (e.g., IV lines) around a core component (the patient's body). This decision was made based on the fact that nurses prefer to see this information arranged relative to the patient's body. In a top-down design manner, IMPROVISE first creates an 'empty' structure diagram and then populates it with components by partitioning and encoding the patient information into different groups.

IMPROVISE prompted us that we can automate the process of deriving user tasks from natural language using advanced linguistic techniques.

HARVEST

In 2009 Gotz and Wen introduced a novel behavior-driven approach [27]. Instead of needing explicit task descriptions, they use implicit task information obtained by monitoring users' behavior to make recommendation more effective. In previous studies it was noted that rather than switching views, users spent significant time attempting to use visualizations provided by default even though they are a poor match for their task. Gotz and Wen aimed to change this. The Behavior-Driven Visualization Recommendation (BVDR) approach has two phases. In the first phase of BDVR, they detect four predefined patterns from user activity. In the second phase, they feed the detected patterns into a recommendation algorithm, which infers user intent in terms of common visual tasks (e.g. comparison) and suggests visualizations that better support the user's needs. The inferred visual task is used together with the properties of the data to retrieve a list of potentially useful visual metaphors from a visualization example corpus made by Zhou and Chen [28]. In the corpus visualization examples are annotated with the visual tasks and data properties for which they are suitable. It contains over 300 examples from a wide variety of sources, including newspapers (e.g., NY Times), design books, and automated graphics generation systems. Unfortunately, we were not able to access this corpus as it was unavailable. Whenever a new recommendation is available, the user interface gives a notification and the user can either accept or decline the recommendation. The authors carried out a user test in which they found that users accepted the visualization recommendations made in 88% of cases. The users also indicated that thanks to the system notifications, they realized that they were not using the best visualization for their tasks. The authors state that in future studies they want the system to also provide an explanation for why a particular recommendation was suggested because they feel this feature would enhance the learning process of the user.

The conclusions made from HARVEST gave us the idea to provide explanations why a certain data visualization was recommended to enhance the educative aspect of our model.

DATASLICER

A very recent study from March 2017 by Alborzi et al. takes yet another novel approach [29]. The authors' hypothesis is that for many data sets and common analysis tasks, there are relatively few "data slices" that result in effective visualizations. Data slices are different subsets of data. Their objective is to improve the user experience by suggesting data slices that, when visualized, present correct solutions to the user's task in an effective way. At any given time in working on the task, users may ask the system to suggest visualizations that would be useful for solving the task. A data slice is considered interesting if past users spent a considerable amount of time looking at its visualization. They first developed a framework which captures exemplary data slices for a user task, explores and parses visual-exploration sequences into a format that makes them distinct and easy to compare. Then they developed a recommendation system, DataSlicer, that matches a "currently viewed" data slice with the most promising "next effective" data slices for the given exploration task. In user tests, DataSlicer significantly improved both the accuracy and speed for identifying spatial outliers, data outliers, outlier patterns and general trends. The system quickly predicted what a participant was searching for based on their initial operations, then presented recommendations that allowed the participants to transform the data, leading them to desired solutions.

The system is interesting, because it deals with the problem of efficiently leading casual or inexperienced users to visualizations of the data that summarize in an effective and prominent way the data points of interest for the user's exploratory-analysis task. The authors do not specify exactly which tasks they include in their system.

3) Summary

All in all, we identify some pitfalls of the existing systems. Such as them not being accessible enough, too complicated, too formal and too secretive when it comes to their recommendation process. The biggest pitfall is that the result of their recommendation process is most commonly a set of data visualizations, which, in our opinion, leaves the users a bit further than they started, but still nowhere, because they still have to choose the most suitable visualization. The possibilities have been narrowed, but a decision still must be made. We hope to avoid these pitfalls within our model.

We establish that we are going to test our model against the solutions available to us. This means Tableau, Watson Analytics, Excel Recommended Charts, Voyager and Google Sheets. Please note that we are going to compare against the recommendation system features of the tools, not the tools as a whole.

B. Exploratory survey

We run a survey among different data science communities on Facebook and LinkedIn. This way, we get respondents who have some sort of familiarity with data science and its terminology. The main goals of the survey are to aid us in making decisions about our model and specifying the term non-expert user. The questions used in the survey are available in Appendix I.

Participants

In total, we gathered 88 valid responses *(n=88).* Out of the 88 respondents, 78% *(n=69)* were male and 22% *(n=19)* female. The average age was 29.86 years.

We had asked the respondents to indicate their knowledge level on a scale of 1 to 10, 1 being beginner and 10 being expert. The average knowledge level was 5.70. It might make sense to divide the scale into two non-expert and expert ranges, however, by definition, a non-expert is a person without professional or specialized knowledge in a particular subject. This implies that the person is not a complete beginner, but has some knowledge of the area. If we divided the scale in the mentioned way, we would also include beginners in the non-expert category. Instead, we opted to divide the scale into three ranges in the following way: 1-3 are beginners, 4-7 are non-experts and 8-10 are experts. According to our ranges we had 26% *(n=23)* beginner level, 44% non-expert *(n=39)* level and 30% *(n=26)* expert level respondents.

Results

We make the following findings from the results of our survey:

- For all groups, the main purpose of making data visualizations was for analysis (65% of beginners, 64% of non-experts, 58% of experts).
- All types of users choose data visualizations mainly according to: the characteristics of their data (57% of beginners, 62% of non-experts, 65% of experts) and the tasks that they want to perform (48% of beginners, 51% of non-experts, 62% of experts). See Appendix I. for all offered options.
- For all groups, the two most used visualizations are bar charts (17% of beginners, 38% of non-experts, 35% of experts) and scatter plots (43% of beginners, 26% of non-experts, 31% of experts).
- All groups were mostly unable to name an existing data visualization recommendation system (0% able vs. 100% unable for beginners, 5% able vs.

95% unable for non-experts and 4% able vs. 96% unable for experts).

- All groups would be willing to use a data visualization recommendation system, although experts were less willing than beginners and non-experts (100% willing vs. 0% not willing for beginners, 87% willing vs. 13% not willing for non-experts and 77% willing vs. 23% not willing for experts).

To summarize, we have learned that non-experts make data visualizations mainly for the purpose of analysis. When they select a suitable data visualization type, they do so according to the characteristics of their data and the tasks they want to perform. Their most used visualization types are bar charts and scatter plots. They are not familiar with data visualization recommender systems but are mostly willing to use one. We also learned that there is not much difference between the approaches of beginners, non-expert and expert users, which was unexpected.

C. Model Requirements

We have decided to name our model **NEViM**. It stands for Non-Expert Visualization Model. Based on research of previous approaches to our problem and the results of our survey, we have identified the following requirements which NEViM should fulfill:

Simplicity

The model should be simple enough to be used by non-experts. It must have good flow and a very straightforward base structure.

Clarity

We aim for the result of our recommendation system to be one data visualization. Not a set, like in some current tools. This means that the underlying classification hierarchy of data visualizations must be clear and unambiguous.

Versatility

We want our model to combine different kinds of recommendation systems. From our survey we learn that when users select a suitable data visualization type, they do so based on the characteristics of their data and the tasks they want to perform. Based on this we incorporate a data characteristics-oriented and task-oriented approach. Furthermore, we want our model to be easily implemented in different programming languages and environments.

Extensibility

Our aim is for our model to be easily extendable. Different types of visualizations are introduced all the time, so we want the process of adding visualizations into the model to be as easy as possible. We want it to be a useful "skeleton" which can be easily extended to include automatic visualizations etc…

Education

We want our model to not only function as a recommender system, but also as a learning tool.

Transparency

Once we recommend a visualization, we want the users to see, why the particular visualization was recommended, meaning that the path to a visualization recommendation through our model has to be retraceable.

Self-learning

We want our model to be able to improve itself. This means, amongst other things, that it should be machine learning friendly.

Competitiveness

We want our model to still produce results which are comparable to results from other systems.

D. Constructing NEViM

1) *What base structure to use for NEViM?*

We started thinking about what kind of existing structure we could use as a base for our model. Since the aim of our model is to help a user *decide* which data visualization to use, the obvious choice seemed to be to consider the structure of decision trees. By definition, a decision tree is a graphical representation of possible solutions to a decision based on certain conditions. It is a classification technique. A decision tree has four main parts: a root node, internal nodes, leaf nodes and branches. Figure 11 shows a basic example of a decision tree along with labels for the different parts. The biggest advantages of decision trees are that they can help uncover unknown alternative solutions to a problem and that they are well suited for machine learning methods.

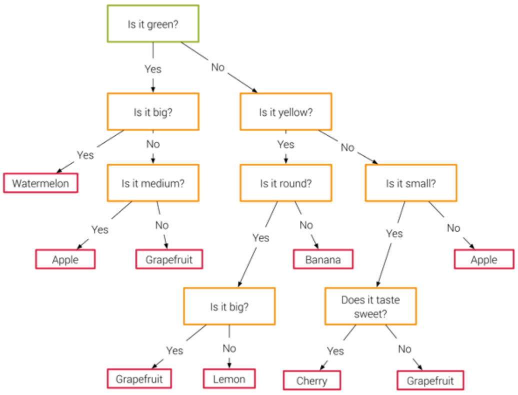


Fig. 11 Simple decision tree example with different parts indicated

Once we determined that the decision tree was a possible base structure, we needed to think about the details. We needed to specify what our root node, internal nodes, leaf nodes and branches would be. It was clear that the leaf nodes would be the different types of data visualizations since that was the outcome that we wanted to achieve. The root node, internal nodes and branches are inspired by Akinator, the Web Genie. Akinator is a game that attempts to determine which person the player is thinking of by asking a series of questions. The structure hidden under the user interface is a decision tree, as in the case of NEViM. Figure 12 shows the interface of Akinator.

Is your character a female?

Fig. 12 User interface of Akinator, the game that inspired our model

Our model's root and internal nodes are questions which possess the ability to clearly distinguish different types of data visualizations. The branches are yes or no answers to those questions.

2) What questions to ask? (Establishing the internal nodes and root node)

The biggest problem of constructing questions for our model was that they must be understandable for non-experts, yet every question should get the user closer to a data visualization recommendation. This means that the subjects of the questions must be features that distinguish the different data visualizations from each other. The key to solving this problem is to base the questions on a clear classification hierarchy. As far as we know, there is no one specific classification hierarchy of data visualizations which would be used globally. We researched different methods of classification and combined them together to derive a classification of our own. This was a very time-consuming process. We went through a total of 20 books [2,4-6,30-45] and for each one, we constructed a diagram showing the classification that was described in the text. Figure 13 shows part of a diagram for the book "Data Visualization: A Handbook for Data Driven Design" by Andy Kirk [4].

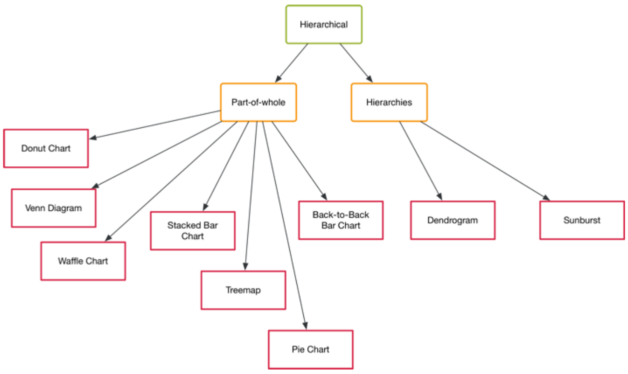


Fig. 13 Example of a classification hierarchy from [4]

We examined the classification hierarchies from books together with hierarchies available from web resources and existing tools (e.g. Table I and Table II). We also made note of any advantages or disadvantages of a specific data visualization, if they were listed. For example in several sources [2,4,5] the authors stated that the pie chart is not suitable for when you have more than 7 parts. The advantages and disadvantages reflected features of the data visualizations that could determine whether they are candidates for recommendation or not, so they are crucial for the final model.

We identified that there are two basic views that the classifications incorporate. The first one is a view from the perspective of the task the user wants to perform. The second is a view from the perspective of the characteristics of the data the user has available. This is in line with data characteristics and task oriented recommendation systems [7].

Some of the classification hierarchies have issues; the most common one being that they mix the different views into one without making a clear distinction between them. We have decided to incorporate different views in our model as well, but we have taken steps to avoid this issue. The views that we have incorporated in our model are: task-based and data characteristics based. The first view is from a task-based perspective, where we are aiming to use the representational goal or user's intentions behind visualizing the data to aid us in recommending a suitable visualization as we believe that the differences in goals can greatly alter the effectiveness of graphical designs. For example, you cannot show composition of something using a line chart. The other view that our model utilizes is from a data-driven perspective, where we gather information about the user's data to make an informed visualization recommendation. Different visualizations are more suitable for different types of data. To exaggerate slightly, you cannot show location data using a pie chart. We have selected the root node of our model to be a question which would distinguish the two views. The root node of NEViM is a question asking "Do you know what your main task is?" if the user answers "Yes", he is first taken in to the task-based branch. If he answers "No", he is taken straight into the data characteristics-based branch.

Once we established the root node, we had to come up with internal nodes. The internal nodes are questions which possess the ability to clearly distinguish different types of data visualizations. The subject of such a question must be something that we define as a distinguishing feature. Based on the findings we made in previous paragraphs, we have established a list of distinguishing features and their hierarchy, which is available in Appendix II.A.

Based on the distinguishing features, we have constructed questions that ask whether that feature is present or not. The list of questions is available in the Appendix II.B.

3) What data visualizations to include? (Establishing the leaf nodes)

Once we had figured out our model's base structure, distinguishing features and questions, the challenge was, which data visualizations to include. We knew that we would not be able to cover all the 300 types of data visualizations available [1] in the initial version of our model. We took a rather quantitative approach to the problem. We went through all the different classification hierarchies we constructed in Section V.D.1 and extracted a list of the data visualizations that occur. We removed duplicates (different names for the same visualization, different layouts of the same visualization) and we counted how many times each data visualization occurred. The ones that occurred 5 times or more were included in our final model. The final list contains 29 data visualizations and you can see it below. Since one of our requirements for the final model is easy extensibility, we feel that 29 data visualizations are appropriate for the initial model.

Bar Chart
Bubble Chart
Cartogram
Choropleth Map
Clustered Bar Chart
Connected Dot Plot
Connection Map
Density Plot
Dot Map
Flow Map
Heat Map
Histogram
Line Chart
Network

Pie Chart
Proportional Symbol Map
Radar Plot
Scatter Plot
Scatter Plot Matrix
Slope Graph
Small Multiples
Stacked Area Chart
Stacked Bar Chart
Stacked Line Chart
Table
Timeline
Tree Map
Word Cloud

Parallel Coordinates Plot
Pie Chart

4) Putting it all together

We classified each of our leaf nodes (data visualizations) using the distinguishing features we constructed previously and that revealed the answers to the questions that lead to a certain leaf node. An example classification of a Pie Chart can be found in Appendix III.A.

We then combined all the classifications together to construct the final model. It contains 107 internal nodes and 105 leaf nodes. The model always results in a recommendation. If no other suitable visualization is found, we recommend to use a table by default. Tableau does this as well. A snapshot from the final model can be seen in Appendix II.C.

E. Testing the Model

1) Can the model compete with existing solutions?

We carried out tests to determine whether our model was able to compete with existing systems in terms of similarity of solutions. We obtained 10 different test data sets with various features (See Table III). The data sets were preprocessed to remove invalid entries and to ensure that all the attributes were of the correct data type.

For each data set, we formulated an example question that a potential user is aiming to answer. This was done in order to determine which attributes of the data would be used in the recommendation procedure. Most existing tools require the user to select the specific attributes that they want to use for their data visualization. By specifying these for each data set we attempt to mimic this behavior. Table III shows the data sets along with their descriptions.

We tested our model against existing solutions which are freely available: Tableau (10.1.1), Watson Analytics (latest version), Microsoft Excel (15.28 Mac), Voyager (2) and Google Sheets (latest version). Figures 14-19 illustrate this process. For each system and every data set, we aimed to achieve a recommendation for a data visualization that would answer the question and incorporate all the specified attributes in one graph as there is no possible way to answer the question without incorporating the specified attributes. Some systems solve more complex questions by creating a series of different data visualizations, with each visualization incorporating a different combination of attributes. We excluded such solutions from our test results because we feel that it is a workaround. For Microsoft Excel and Google Sheets, the recommendation process results in several recommendations and the systems do not rank them. For these cases we recorded all valid recommendations.

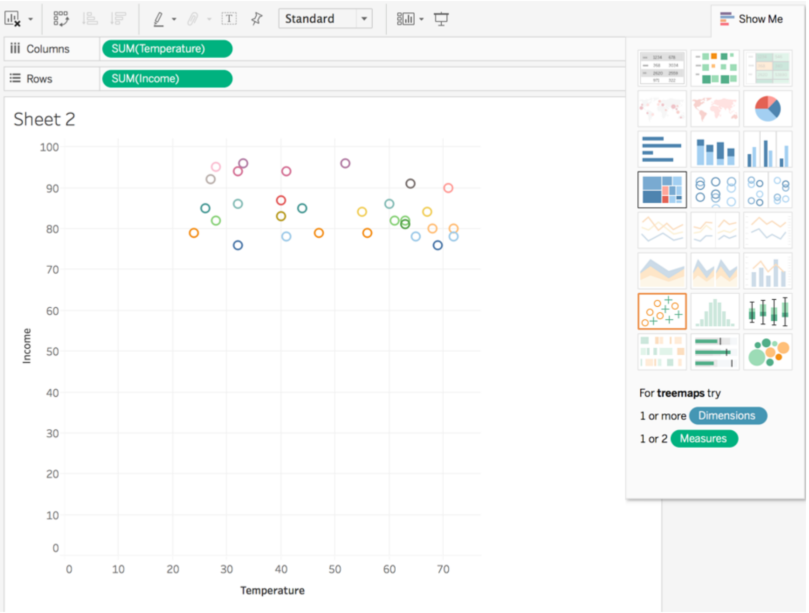


Fig. 14 Example of recommendations made by Tableau for data set 5

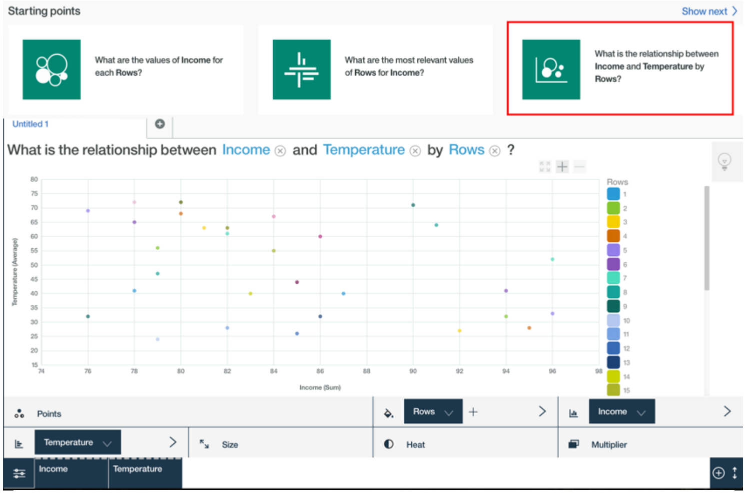
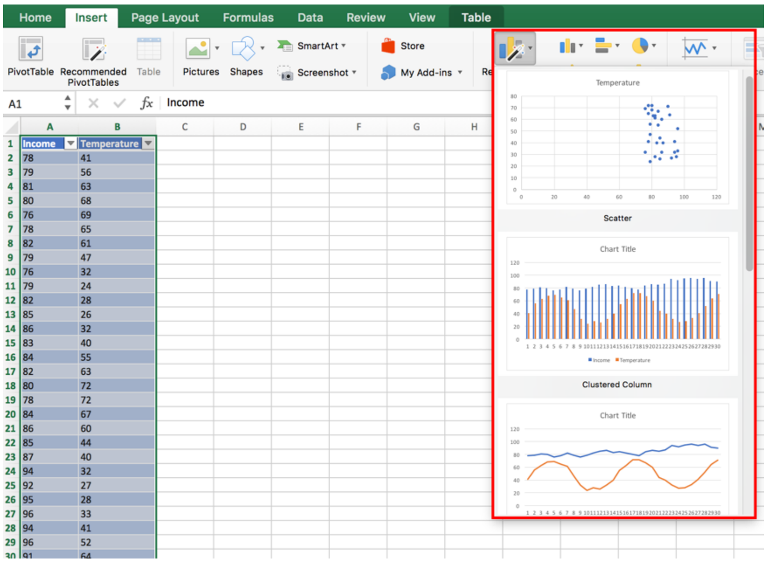

Starting points
Show next
What are the values of Income for each Rows?
What are the most relevant values of Rows for Income?
What is the relationship between Income and Temperature by Rows?
Untitled 1
What is the relationship between Income and Temperature by Rows ?
Temperature (Average)
Income (Sum)
Rows
Points
Rows
Income
Temperature
Size
Heat
Multiplier
Income
Temperature

Fig. 15 Example of recommendations made by Watson Analytics for data set 5



	A	B
1	Income	Temperature
2	78	41
3	79	56
4	81	63
5	80	68
6	76	69
7	78	65
8	82	61
9	79	47
10	76	32
11	79	24
12	82	28
13	85	26
14	86	32
15	83	40
16	84	55
17	82	63
18	80	72
19	78	72
20	84	67
21	86	60
22	85	44
23	87	40
24	94	32
25	92	27
26	95	28
27	96	33
28	94	41
29	96	52
30	91	64

Fig. 16 Example of recommendations made by Microsoft Excel for data set 5

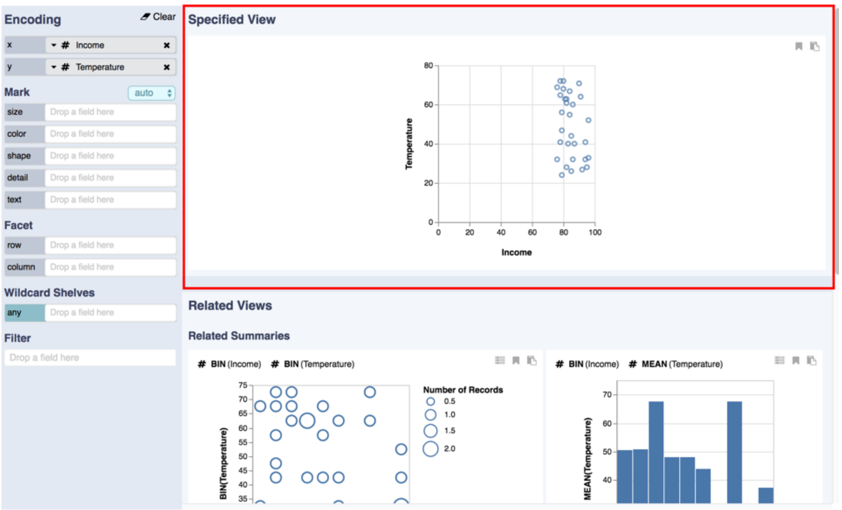
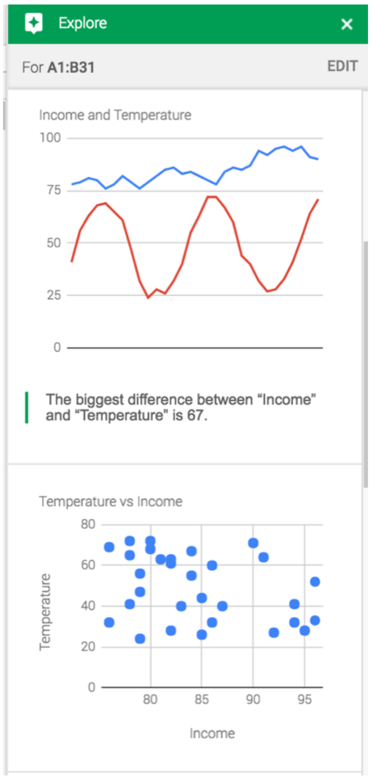


Fig. 17 Example of recommendations made by Voyager for data set 5



The biggest difference between “Income” and “Temperature” is 67.

Fig. 18 Example of recommendations made by Google Sheets for data set 5

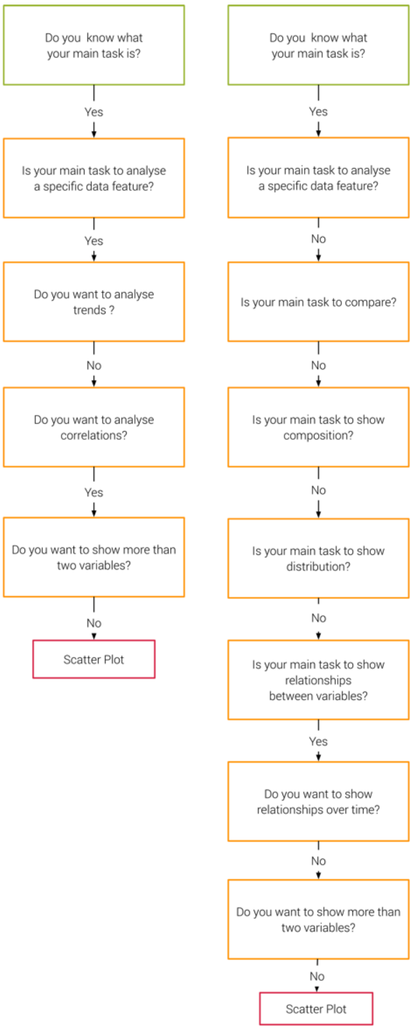

Do you know what your main task is?
Yes
Is your main task to analyse a specific data feature?
Yes
Do you want to analyse trends ?
No
Do you want to analyse correlations?
Yes
Do you want to show more than two variables?
No
Scatter Plot
Do you know what your main task is?
Yes
Is your main task to analyse a specific data feature?
No
Is your main task to compare?
No
Is your main task to show composition?
No
Is your main task to show distribution?
No
Is your main task to show relationships between variables?
Yes
Do you want to show relationships over time?
No
Do you want to show more than two variables?
No
Scatter Plot

Fig. 19 Example of recommendations generated made by NEViM for data set 5. There were two possible paths.

Data set	Description	No. of records	Question	Used attributes	Excel	Google Sheets	Tableau	Voyager	Watson Analytics	NEViM
1	Favourite subjects within a class of students.	7	What does the composition of the data look like?	subject, number of students	Bar chart, Pie chart	Bar chart, Pie chart	Bar chart	Bar chart	Bar chart	Bar chart
2	Average prices of cigarettes in the state of Arizona over several years.	8	What was the development of the price of cigarettes over the years?	year, average price	Line chart, Bar chart	Line chart	Line chart	Bar chart	None	Line chart
3	Percentage of men and women in EU countries for 2016.	28	Which 5 countries have the highest percentage of females?	country, percentage of women, percentage of men	Clustered bar chart, Scatter plot, Stacked bar chart	Clustered bar chart	Proportional symbol map	Scatter plot	Clustered bar chart	Clustered bar chart
4	Causes of death in Kenya in 2012	12	How big of a part does each cause take?	cause of death, number of deaths, percentage of total	None	Pie chart	Tree map	None	None	Tree map
5	Daily ice cream sales along with air temperatures.	30	Are ice cream sales related to the weather?	income, temperature	Scatter plot, Clustered bar chart, Line chart, Stacked bar chart	Line chart, Scatter plot, Clustered bar chart	Scatter plot	Scatter plot	Scatter plot	Scatter plot
6	Email communication between researchers working together.	461	Which researcher is connected to most people?	sender, receiver	None	None	None	Scatter plot	None	Network
7	Finishing times of runners in the 2014 Boston Marathon.	32K	Which finishing time interval was the most common?	finishing time	Scatter plot, Line chart	Line chart, Histogram	Histogram	None	Histogram	Histogram
8	Records of UFO sightings with detailed information.	80K	Are there any clusters of locations where UFOs have been seen more often?	latitude, longitude	None	None	None	None	None	Dot map
9	List of cars and their parameters.	393	Are there any relationships between the different parameters?	miles per gallon, number of cylinders, displacement horsepower, weight acceleration, year	Stacked line chart	None	None	None	None	Parallel coordinates
10	Origins and destinations of flights within the US.	4K	Which city has the most ingoing and outgoing flights?	flight origin, flight destination	None	None	Proportional symbol map	None	None	Connection map

Table III. Results of our competitiveness experiment for each data set and each system

Results

For data set 1, all systems recommended a bar chart. Excel and Google Sheets also recommended a pie chart. The recommendations for data set 2 were either line charts or bar charts. The specified question could be answered by either of these. Watson Analytics was not able to give a recommendation because it couldn't recognize that the average price attribute was a number. We have attempted resolving this issue but were not able to. For data set 3, the majority recommendation was a clustered bar chart, in line with the recommendation made by NEViM. Data set 4 proved to be challenging for Voyager and Watson Analytics. Since the data was hierarchical and the question was asking to see parts-of-whole, a suitable solution would be a tree map. A pie chart shows parts-of-whole, but does not indicate hierarchy. The question asked for data set 5 could be answered using different types of data visualizations. Since it is asking to analyze the correlation between 2 variables, a scatter plot is a suitable solution. All systems recommended it. Data set number 6 was an example of a social network, thus the most suitable visualization would be a network. However, the answer to the specified question could also be answered with a scatter plot as suggested by Voyager. This is because networks can also be represented as adjacency matrices and the scatter plot generated by Voyager is essentially an adjacency matrix. Data set 7 and its question were aimed at visualizing distributions. Distributions can be visualized, among others with histograms, scatter plots and line charts. Data set 8 was an example of spatial data. Spatial data is best visualized through maps. Tableau offers map visualizations but we suspect that it cannot plot on the map according to latitude and longitude coordinates. Watson Analytics and Google Sheets have the same issue. Microsoft Excel and Voyager do not support maps at all. In Data set 9 the answer to the question was revealed through comparing 7 attributes. This meant that the visualization has to support 7 different variables. Both stacked line chart and parallel coordinates are valid solutions. The final data set 10 was again spatial. This time it could be solved through plotting on a map but also by analyzing the distribution of the data set. Both proportional symbol map and connection map (as a flight implies a connection between two cities) are valid solutions.

Overall, we can observe that NEViM provided usable solutions in all cases. The users have several paths that they can take through NEViM to get to a recommendation, depending on what information they know about their data or their task. NEViM has an advantage that it is not limited by implementation. Since two of our data sets were aimed at spatial data visualization (9 and 10) and one at network data visualization (6), some systems were not able to make recommendations simply because they do not support such visualization types. Furthermore, NEViM includes more types of visualizations than any of the current systems, which results in recommendations for specialty visualizations that can be more suitable for a certain task. Another advantage is that it always results in only one recommendation, unlike Microsoft Excel or Google Sheets, where the user has to choose which one out of the set of recommendations to use. According to our survey, the most used visualization tool which incorporates a recommender system is Tableau *(28% of non-expert respondents)*. From the result table, we can see that in 5 out of 7 valid cases, NEViM made the same recommendation as Tableau. Furthermore, in data set 3 Tableau also made a recommendation for a Clustered Bar Chart, like NEViM did, but it was not the resulting recommendation. One of the attributes was the name of a country, so Tableau evaluated the data as spatial. We have noticed that whenever there is a geographical attribute, Tableau prefers to recommend maps, even though they might not be the most suitable solution.

2) Adding a new data visualization

We demonstrate that our model is easily extensible by showing how a new data visualization type would be added to it; a Sankey diagram, for instance. Sankey diagrams are specific types of flow diagrams and they display quantities in proportion to one another. An example of a Sankey diagram can be seen in Figure 20.

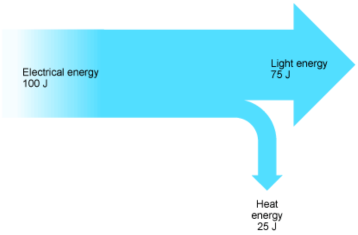


Fig. 15 Example of a Sankey diagram showing the distribution of energy in a filament lamp [30]

We look into the classifications that we already have and search for the most similar one. We find out that the Tree Map has the same classification. So we need to find a distinguishing feature between a Tree Map and a Sankey diagram. That feature is, that a Sankey diagram shows flow. We search through the model and find occurrences of a Tree Map. We then add a question asking “Do you want to show flow?”. If the user answers “Yes”, he gets a recommendation for a Sankey diagram. If he answers “No” he gets a tree map. Figure 21 shows the two paths that a user can take to get to the Sankey diagram.

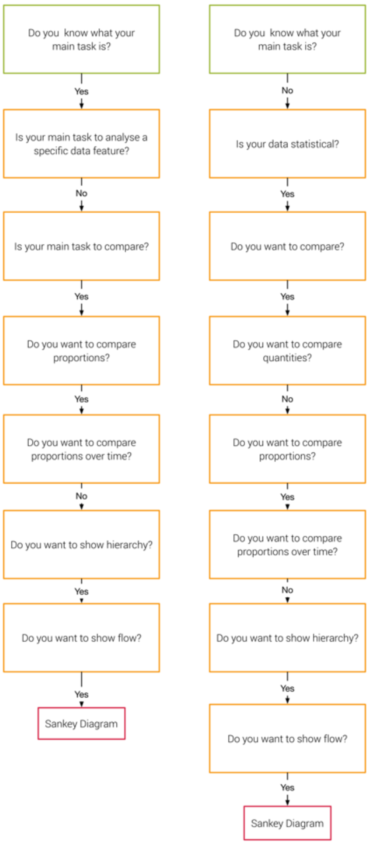

Do you know what your main task is?
Yes
Is your main task to analyse a specific data feature?
No
Is your main task to compare?
Yes
Do you want to compare proportions?
Yes
Do you want to compare proportions over time?
No
Do you want to show hierarchy?
Yes
Do you want to show flow?
Yes
Sankey Diagram
Do you know what your main task is?
No
Is your data statistical?
Yes
Do you want to compare?
Yes
Do you want to compare quantities?
No
Do you want to compare proportions?
Yes
Do you want to compare proportions over time?
No
Do you want to show hierarchy?
Yes
Do you want to show flow?
Yes
Sankey Diagram

Fig. 16 Two possible paths to reach a Sankey diagram (left: task-based, right: data-based)

Our more detailed classification of a Sankey diagram can be found in Appendix III.B.

VI. Discussion & Conclusions

To recapitulate our aim, ever since we began the research on this topic, we knew that our result was not going to be a fully-blown software tool, but rather a model that can be used in the backend of such a tool. We use this approach because it does not constrain the usability of our solution: programmers and developers can implement it in a way that is suitable for their specific needs. Furthermore, we were not aiming to incorporate actual visualization creation, display or editing into our model.

We managed to build a model for a data visualization recommender system suited to non-experts called NEViM. Through testing, we have managed to show that the resulting recommendations are similar or identical to the ones generated by existing solutions. Based on a review of existing work and a explorative survey among users, we have put together requirements. This is an evaluation of how NEViM managed to fulfill these:

Simplicity

Thanks to its question-based structure, using the model is simple. The user only has to answer questions saying yes or no. The basic structure is very straightforward. However, we have recognized a potential weak point of our model. With the use of data science terminology in our questions, we risk that a non-expert user might not be familiar with it and that might result in not reaching a suitable recommendation.

Clarity

The result of our recommendation system is a single data visualization, making it very clear. In the case that none of the data visualizations within the model are determined as suitable, the model still makes a recommendation to visualize using a table.

Versatility

NEViM combines two different types of data visualization recommendation systems as defined in [7]: task-oriented and data characteristics-oriented. These two types are distinguished by two different starting points within our model. Thanks to the base structure of a decision tree, we can see the model being easily implemented in various different programming languages and environments.

Extensibility

To illustrate the extensibility of the model, we have added the Sankey diagram visualization. This proved to be a doable task. Thanks to the base structure of a decision tree, NEViM is suitable to be used with machine learning.

Education

This requirement has not been met yet. For suggestions on how we mean to fulfill it in the future, see the Future Work section.

Transparency

The traversal through our model is logical enough that it is clear why a certain type of data visualization was recommended.

Self-learning

As stated previously, our model is machine learning friendly and techniques can be applied for it to be able to self-learn. See our Future Work section.

Competitiveness

Through testing we have proved that our model produces recommendations similar or identical to existing solutions. Furthermore, unlike, other systems, it provided suitable solutions for all problems that were asked within testing.

A disadvantage of NEViM that we have identified is that the user has to either know what their main task is, or know what type of data they have. The question is, whether non-expert users will be able to determine this by themselves. We believe that this disadvantage could be fixed through user testing to validate the overall structure of the model as well as the quality of the questions. Furthermore, the questions could be checked by a linguistics expert to see whether there are some difficulties in the wording leading to possible ambiguous interpretations.

We identified that another disadvantage might lie in the fact that since we use data science terminology in our questions, we risk that non-experts might not be familiar with it and as a result might not be able to answer the question. A solution to this problem could be to clarify the terms using a dictionary definition, which could for example pop up when the user hovers over the unfamiliar term. Another option could be to add an “I don’t know” option (as in Akinator) and when the user selects this option, he would be given a further explanation of the terms. As we can see, the solutions to the problem are more part of the implementation phase, not the theoretical model phase which we discuss here, but it is important to keep this in mind for future work.

A difficulty in the usability of our model might be that the traversal through it is quite lengthy. This is due to the chosen question-based approach. In the current state of the model, the user has to answer many questions to get to a recommendation, because all the other possibilities have to be ruled out in the process. For example, if my main task is to show relationships, I have to answer questions about whether I want to compare, show distribution or show composition before I get there. A potential fix for this could be to present the tasks in the form of a multiple choice question. This way, the user could see beforehand what other options are available and might find a more suitable task they want to perform. We also wouldn’t have to worry about whether the ordering of the question might be a source of bias. This is once again a problem that could be fixed easier in the implementation phase.

We have questioned whether the choice to recommend a table when no other suitable visualization is found is the correct one. The reason we implemented this behavior in the first place was that it is implemented in Tableau. Since our model is rather extensive, it is quite unlikely that no visualization will be recommended. There is an ongoing debate about when it is best to not visualize things, as discussed by Stephanie Evergreen [31]. We might choose to go with a different solution in the future. A possible aid for the fix to this problem in the implementation phase could be to give the users a possibility to rate the resulting recommendation and suggest improvements. See Future Work for a more detailed elaboration on this. Data could also be collected to find out in how many cases the Table option is reached, to identify whether it is necessary to further concern ourselves with this issue.

VII. Future work

During the testing phase we found that it was quite challenging to get a recommendation out of some of the tools, especially as we also classify as non-experts. This made us wonder whether using our model to get a recommendation before actually using available data visualization software tools would aid non-experts in navigating through them. NEViM could help them establish what their goals are and which attributes they should use.

We have proved that there is definitely a place for our model in the data science world. The positive feedback we have gotten from our survey respondents surprised us and motivated us to work on this model further. The logical next step would be to perform more tests with more data sets and make improvements to the model. Then the model could be tested with non-expert users.

Another way of improving the model could be to implement it as a web application to make it accessible and users could rate the resulting recommendations, suggest new paths through the model or request new visualization types to be included. Furthermore, in a possible implementation of the model, the final recommendation could be enhanced with useful information about the data visualization type, tips on how to construct it, which tools to use and examples of already made instances. This would transform the model into a very useful educative tool and fulfill the Education requirement that we have set.

Another possible extension to the model could be to add another view which would incorporate information about the domain that the user's data comes from. There are data visualizations that are more suited for a specific data domain than others. For example, the area of economics has special types of data visualizations that are more suited to exposing different economic indicators. This would result in the model being a combination of data characteristics, task and domain knowledge oriented data visualization systems recommender systems according to the classification in [7].

Thanks to its structure, NEViM is machine learning friendly. For example, neural networks could be used to make the model self-learning and self-improving.

We could also introduce different features that could influence the ranking of the visualizations, for example by taking into consideration perceptual qualities of the different data visualization types. Now that we have established a successful base, the possibilities for further development are endless.

References

[1] M. Bostock (n.d.). Data-Driven Documents. Retrieved August 4, 2017, from https://d3js.org/

[2] C. O'Neil and R. Schutt, *Doing Data Science: Straight Talk From The Frontline*, Sebastopol, CA: O'Reilly Media, (2014).

[3] J.W. Tukey, *Exploratory Data Analysis*, 1st ed., Reading, Mass.: Addison-Wesley, (1970).

[4] A. Kirk, *Data visualization: A handbook for data driven design*, London: SAGE, (2016).

[5] N. Iliinsky and J. Steele. *Designing data visualizations: representing informational relationships*. " O'Reilly Media, Inc.", (2011).

[6] T. Munzner and E. Maguire, *Visualization analysis & design*. Boca Raton, FL: CRC Press, (2015).

[7] P. Kaur and M. Owonibi, "A Review on Visualization Recommendation Strategies." (2017).

[8] S. Gnanamgari. "Information presentation through default displays". Ph.D. dissertation, Philadelphia, PA, USA (1981).

[9] J. Mackinlay, "Automating the design of graphical presentations of relational information." *Acm Transactions On Graphics* (Tog) 5.2 (1986): 110-141.

[10] J. Bertin, "Semiology of graphics: diagrams, networks, maps." (1983).

[11] S. Casner and J. H. Larkin, "Cognitive efficiency considerations for good graphic design." No. AIP-81. CARNEGIE-MELLON UNIV PITTSBURGH PA ARTIFICIAL INTELLIGENCE AND PSYCHOLOGY PROJECT, (1989).

[12] S. F Roth and J. Mattis, "Data characterization for intelligent graphics presentation." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, (1990).

[13] P. Hanrahan, "Vizql: a language for query, analysis and visualization." *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*. ACM, (2006).

[14] C. Stolte, D. Tang and Pat Hanrahan. "Polaris: A system for query, analysis, and visualization of multidimensional relational databases." *IEEE Transactions on Visualization and Computer Graphics* 8.1 (2002): 52-65.

[15] J. Mackinlay, P. Hanrahan and C. Stolte, "Show me: Automatic presentation for visual analysis." *IEEE transactions on visualization and computer graphics* 13.6 (2007).

[16] F. Viegas et al. "ManyEyes: a site for visualization at internet scale." *IEEE transactions on visualization and computer graphics* 13.6 (2007).

[17] Smart data analysis and visualization. (n.d.). Retrieved August 4, 2017, from https://www.ibm.com/watson-analytics

[18] A. Key et al. "Vizdeck: self-organizing dashboards for visual analytics." *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*. ACM, 2012.

[19] Available chart types in Office. (n.d.). Retrieved August 4, 2017, from https://support.office.com/

[20] M. Vartak et al. "SeeDB: supporting visual analytics with data-driven recommendations." VLDB, 2015.

[21] K. Wongsuphasawat et al. "Voyager: Exploratory analysis via faceted browsing of visualization recommendations." *IEEE*

transactions on visualization and computer graphics 22.1 (2016): 649-658.

[22] Vega Compass. (n.d.). Retrieved August 4, 2017, from https://github.com/vega/compass

[23] A. Satyanarayan et al. "Vega-lite: A grammar of interactive graphics." *IEEE transactions on visualization and computer graphics* 23.1 (2017): 341-350.

[24] Chart and Graph Types. (n.d.). Retrieved August 9, 2017, from https://support.google.com/

[25] S. Wehrend and C. Lewis. "A problem-oriented classification of visualization techniques." *Proceedings of the 1st Conference on Visualization'90*. IEEE Computer Society Press, 1990.

[26] M. X. Zhou and S. K. Feiner, "Visual task characterization for automated visual discourse synthesis." *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co., 1998.

[27] D. Gotz and Z. Wen. "Behavior-driven visualization recommendation." *Proceedings of the 14th international conference on Intelligent user interfaces*. ACM, 2009.

[28] M. X. Zhou, M. Chen, and Y. Feng. "Building a visual database for example-based graphics generation." *Information Visualization, 2002. INFOVIS 2002*. IEEE Symposium on. IEEE, 2002.

[29] F. Alborzi et al. "DataSlicer: Task-Based Data Selection for Visual Data Exploration." arXiv preprint arXiv:1703.09218 (2017).

[30] Bbccouk. (2017). Bbccouk. Retrieved 15 August, 2017, from http://www.bbc.co.uk/schools/gcsebitesize/science/aqa/energyefficiency/energytransfersrev3.shtml

[31] S. D. Evergreen. *Effective data visualization: The right chart for the right data*. SAGE Publications, (2016).

[32] N. Yau. *Visualize This: The FlowingData Guide to Design, Visualization, and Statistics*. John Wiley & Sons, (2011).

[33] N. Yau. *Data points: Visualization that means something*. John Wiley & Sons, (2013).

[34] J. Heer et al. “A tour through the visualization zoo”. *Queue, 8*(5), 2010.

[35] M. Hardin et al. “Which chart or graph is right for you?. Tell Impactful Stories with Data”. *Tableau Software*, 2012.

[36] M. Yuk and S. Diamond. *Data visualization for dummies*. John Wiley & Sons, (2014).

[37] R. Brath and D. Jonker. *Graph analysis and visualization: discovering business opportunity in linked data*. John Wiley & Sons, (2015).

[38] K. Börner and D. E. Polley. *Visual insights: A practical guide to making sense of data*. MIT Press, (2014).

[39] A. C Telea. *Data visualization: principles and practice*. CRC Press, (2007).

[40] K. Börner. *Atlas of knowledge: anyone can map*. MIT Press, (2015).

[41] C. Ware. *Visual thinking: For design*. Morgan Kaufmann (2010).

[42] C. Ware. *Information visualization: perception for design*. Elsevier, (2012).

[43] M. Stacey et al. *Visual intelligence: Microsoft tools and techniques for visualizing data*. John Wiley & Sons, (2013).

[44] B. Hinderman. *Building responsive data visualization for the web*. John Wiley & Sons, (2015).

[45] Z. Gemignani et al. *Data fluency: Empowering your organization with effective data communication*. John Wiley & Sons, (2014).

APPENDIX I.

Survey Questions

Q1 How old are you?

Q2 What is your gender?

Q3 Which country are you from?

Q4 Please indicate your level of data visualization knowledge. (Scale of 1 (Beginner) to 10 (Expert)

Q5 How long have you been working in a data visualization related field? (in months)

Q6 If you had to estimate, how many data visualizations have you made in the past year?

- I haven't made any
- Less than 10
- 10-50
- 51-100
- More than 100

Q7 Which software do you mostly use to create your data visualizations?

- Tableau
- Excel
- Gephi
- Other (please specify) _________

Q8 According to you, what is the main benefit of data visualization?

Q9 Which basic type of data visualizations do you use the most?

- **Tables -** A table is an ordered arrangement of rows and columns in a grid.
- **Charts** - Charts visually depict quantitative and qualitative data without using a well-defined reference system.
- **Graphs** - A graph plots quantitative and/or qualitative data variables using a well-defined reference system, such as coordinates on a horizontal or vertical axis.
- **Maps -** Maps display data records visually according to their physical (spatial) relationships and show how data are distributed geographically.
- **Networks -** Network layouts use nodes to represent sets of data records, and links connecting nodes to represent relationships.

Q10 What are your **top 3 favourite** visualization techniques? (e.g. bar chart, scatterplot, line chart, treemap...)

Q11 What are your **top 3 most used** visualization techniques? (e.g. bar chart, scatterplot, line chart, treemap...)

Q12 What is your main goal when you make data visualizations?

- Analysis
- Presentation
- Enjoyment
- Other (please specify) ____

Q13 Which of the following tasks do you usually perform using your data visualization?

- Categorization
- Clustering
- Comparing
- Analysing trends
- Ordering, ranking and sorting
- Analysing distribution
- Finding correlations and relationships
- Analysing geospatial location
- Other (please specify) _________________________

Q14 When you choose a suitable visualization for your data, according to what criteria do you choose?

- The characteristics of my data.
- The tasks that I want to perform.
- Knowledge of the domain.
- The preferences of the potential users.
- Ease of understanding.
- Aesthetic appeal.

Q15 Do you know any data visualization recommender systems?

Q16 Would you be open to use a data visualization recommender system?

APPENDIX II.

A. Distinguishing features used in our model with indication of their hierarchy

1. Suitability for a specific task
    a. Comparing
        i. Over time
        ii. Quantities
        iii. Proportions
        iv. Other
    b. Analyzing
        i. Trends
        ii. Correlations
        iii. Distribution
        iv. Patterns
        v. Clusters
    c. Showing
        i. Composition
            1. Hierarchy
        ii. Distributions

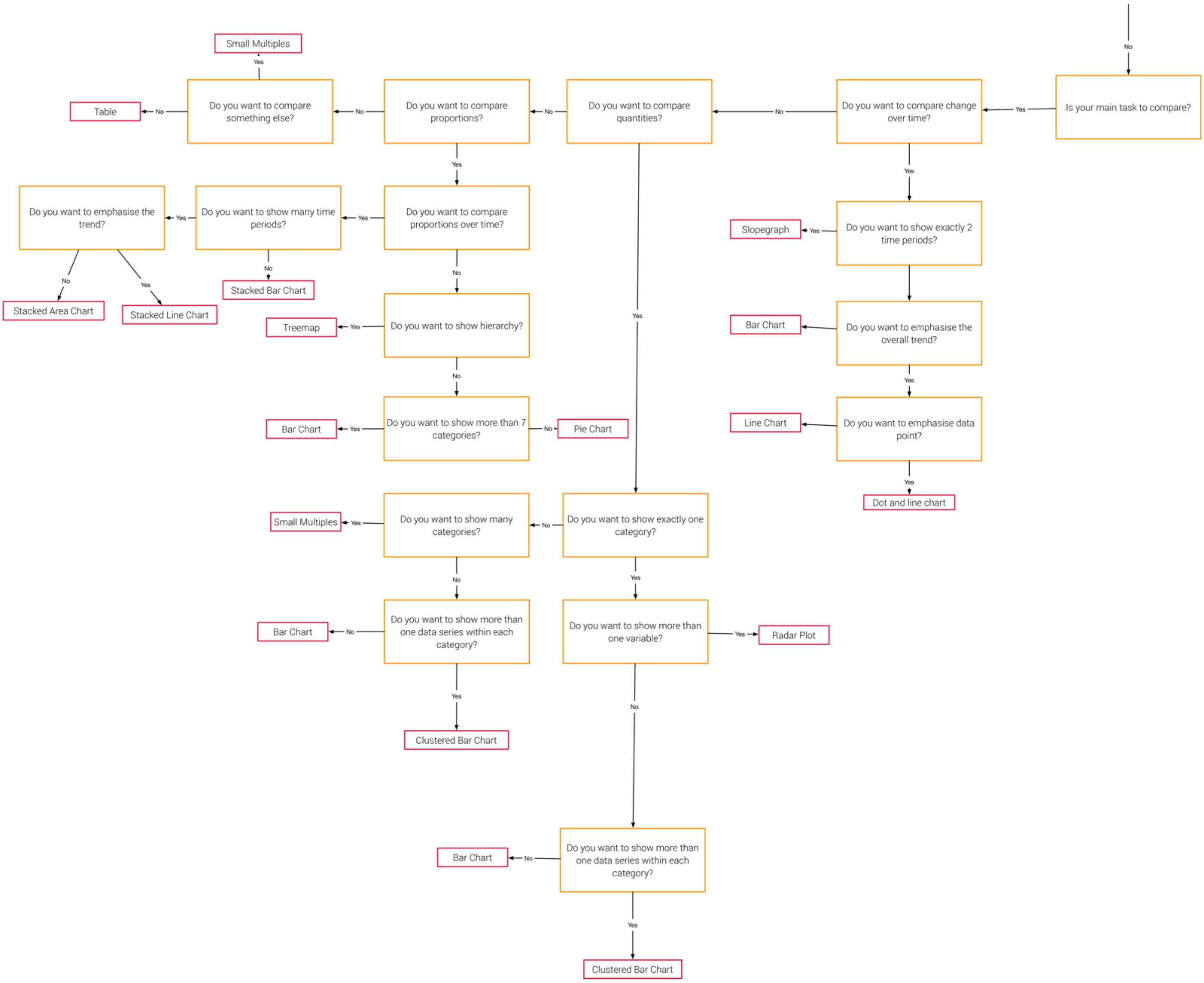
iii. Relationships
iv. Connections
v. Locations

2. Suitability for displaying a specific data type
    a. Statistical
    b. Temporal
    c. Geospatial
3. Ability to emphasize specific data features
    a. Trend line
    b. Shape of distribution
    c. Data points
4. Number of dimensions the visualization can show at once (by default, not including the use of color etc…)
    a. Specific number
5. Number of variables the visualization can show at once
    a. Specific number
6. Number of time periods the visualization can show at once
    a. Specific number
7. Number of categories the visualization can show at once
    a. Specific number
8. Number of data series the visualization can show at once
    a. Specific number

A. Questions used in our model

1. Is your main task to compare over time?
2. Is your main task to compare quantities?
3. Is your main task to compare proportions?
4. Is your main task to compare something else?
5. Is your main task to analyze trends?
6. Is your main task to analyze correlations?
7. Is your main task to analyze distribution?
8. Is your main task to analyze patterns?
9. Is your main task to analyze clusters?
10. Is your main task to show composition?
11. Is your main task to show composition with a hierarchy?
12. Is your main task to show distribution?
13. Is your main task to show relationships between variables?
14. Is your main task to show connections?
15. Is your main task to show locations?
16. Is your data statistical?
17. Is your data temporal?
18. Is your data geospatial?
19. Do you want to emphasize the trend line?
20. Do you want to emphasize the shape of the distribution?
21. Do you want to emphasize each data point?
22. Do you want to show [a specific number] of variables?
23. Do you want to show [a specific number of dimensions?
24. Do you want to show a specific number of time periods?
25. Do you want to show [a specific number] of categories?
26. Do you want to show [a specific number] of data series?

B.Snapshot of a section of NEViM


Is your main task to compare?
No
Yes
Do you want to compare change over time?
No
Do you want to compare quantities?
No
Do you want to compare proportions?
No
Do you want to compare something else?
No
Table
Yes
Small Multiples
Yes
Do you want to compare proportions over time?
Yes
Do you want to show many time periods?
Yes
Do you want to emphasise the trend?
No
Stacked Area Chart
Yes
Stacked Line Chart
No
Stacked Bar Chart
No
Do you want to show hierarchy?
Yes
Treemap
No
Do you want to show more than 7 categories?
Yes
Bar Chart
No
Pie Chart
Yes
Do you want to show exactly one category?
No
Do you want to show many categories?
Yes
Small Multiples
No
Do you want to show more than one data series within each category?
No
Bar Chart
Yes
Clustered Bar Chart
Yes
Do you want to show more than one variable?
Yes
Radar Plot
No
Do you want to show more than one data series within each category?
No
Bar Chart
Yes
Clustered Bar Chart
Yes
Do you want to show exactly 2 time periods?
Yes
Slopegraph
Do you want to emphasise the overall trend?
Bar Chart
Yes
Do you want to emphasise data point?
Line Chart
Yes
Dot and line chart

APPENDIX III.

A. Example classification of a Pie Chart

1. Is your main task to compare over time? No
2. Is your main task to compare quantities? No
3. Is your main task to compare proportions? Yes
4. Is your main task to compare something else? No
5. Is your main task to analyze trends? No
6. Is your main task to analyze correlations? No
7. Is your main task to analyze distribution? No
8. Is your main task to analyze patterns? No
9. Is your main task to analyze clusters? No
10. Is your main task to show composition? Yes
11. Is your main task to show distribution? No
12. Is your main task to show relationships between variables? No
13. Is your main task to show connections? No
14. Is your main task to show locations? No
15. Is your data statistical? Yes
16. Is your data temporal? No
17. Is your data geospatial? No
18. Do you want to show hierarchy? No
19. Do you want to emphasize the trend line? No
20. Do you want to emphasize the shape of the distribution? No
21. Do you want to emphasize each data point? No.
22. Do you want to show a specific number variables? Yes, 2.
23. Do you want to show a specific number of dimensions? Yes, 2.
24. Do you want to show a specific number of time periods? Yes,1.
25. Do you want to show a specific number of categories? Yes, less than 7.
26. Do you want to show a specific number of data series? Yes, 1.

B. Classification of a Sankey Diagram

1. Is your main task to compare over time? No
2. Is your main task to compare quantities? No
3. Is your main task to compare proportions? Yes
4. Is your main task to compare something else? No
5. Is your main task to analyze trends? No
6. Is your main task to analyze correlations? No
7. Is your main task to analyze distribution? No
8. Is your main task to analyze patterns? No
9. Is your main task to analyze clusters? No
10. Is your main task to show composition? Yes
11. Is your main task to show distribution? No
12. Is your main task to show relationships between variables? No
13. Is your main task to show connections? No
14. Is your main task to show locations? No
15. Is your data statistical? Yes
16. Is your data temporal? No
17. Is your data geospatial? No
18. Do you want to show hierarchy? Yes

19. Do you want to emphasize the trend line? No
20. Do you want to emphasize the shape of the distribution? No
21. Do you want to emphasize each data point? No.
22. Do you want to show [number] variables? Yes.
23. Do you want to show [number] dimensions? Yes, 2.
24. Do you want to show [number] time periods? Yes, 1.
25. Do you want to show [number] categories? Yes.
26. Do you want to show [number] data series? Yes, 1.