Final Delivery InfoVis - Nutrition Analyzer

João Bagorro
IST, 95604
Lisbon, Portugal
joao.bagorro@tecnico.ulisboa.pt

Martim Santos
IST, 95638
Lisbon, Portugal
martim.santos@tecnico.ulisboa.pt

João Maurício
IST, 98530
Lisbon, Portugal
joaogmauricio@tecnico.ulisboa.pt

1. INTRODUCTION

Since 1975, worldwide obesity has nearly tripled. In 2016, more than 1.9 billion adults, 18 years and older, were overweight. Of these over 650 million were obese [1]. Globally, one billion people, including 1 in 5 women and 1 in 7 men, are estimated to be living with obesity by 2030. The fundamental cause of obesity and overweight is an energy imbalance between calories consumed and calories expended [2].

The idea of this project is to give detailed information about various food items. The main objective is to make people aware about the quality of food that they usually consume and encourage them to choose a healthier option. We intend to study the characteristics of a large group of food items which provide information about whether the item is rich in protein, has too much salt, a high calorie density or a high percentage of salt, amongst others.

With this work, we pretend to motivate people and help them put their eating habits into perspective together with persuading them to improve their diet.

In order to pursue our objective, we decided to compact as much useful information as possible in one visualization. There are several physical and online sources that provide nutritional information about almost every specific food item individually however, it proves to be effortful to compare multiple items with each other while grouping them by categories or common characteristics. Our intent is to facilitate comparisons between food items avoiding numerous internet searches to find different nutritional values and information about the products that people consume on a daily basis. With centralized information about multiple products, the user can easily navigate through a diversity of products that he might be interested in.

With this proposed approach and the fact that the data is widely known and familiar to many people, we hope that more people other than us realize the relevance of this theme. A Forbes survey, made in 2018, showed that only a third (31.4%) of 1800 Americans (aged 25-36) said they used the Nutrition Facts label found on most foods and beverage packages “frequently”. As the study mentions, “women, those with higher education and income, and those concerned about their weight used the labels more frequently. The number one food attribute looked at the most was sugar, by almost three-quarters of the respondents, followed by total calories (72.9%), serving size (67.9%) and ingredients (65.8%)”. These statistics emphasize the relevance of the topic presented in this work and highlights our motivation to pursue our objective.

We hope that this visualization helps people put their eating habits into perspective and persuades them to improve their diet along with being aware of the food that they ingest. For that, we decided to work with a dense amount of data that categorizes hundreds of relevant and common food items. We tried to obtain the most complete and flawless group of data amongst that total, understanding what is relevant or not for the objective of this visualization. This would facilitate the understanding of the message we are trying to transmit.

Given the purpose of this work, some questions that we pretend to answer are:

- Do food items with a high protein percentage and low calories per serving generally have a low percentage of saturated fat?
- How does the total fat and sodium influence the cholesterol of an item?
- Is fish generally less caloric than meat with a low fat percentage?
- Does fiber rich starchy food tend to have less cholesterol?
- Is water rich food healthier?
- Does caffeine reduce the presence of B-vitamins, vitamin C and minerals in the food?

These questions were defined back in the first checkpoint of this work because of their solidness, relevancy and the fact that they are good examples of questions that we pretend to answer with our visualization.

2. RELATED WORK

Firstly, we took into consideration some examples from the Information Visualization 2022/2023 Moodle page.

In this page, we discussed some of the examples as a group and learned the basics on how to make good representations of certain types of data as well as how to organize the idioms coherently, in an appealing way for the end-user.

In the section called “Hall of Fame” [3] (projects made by students from previous years) from the course’s Moodle, we found the “Marvel vs DC: Comic Book Characters” visualization by Daniel Serafim, Guilherme Nunes and

Lídia Custódio quite interesting. Since each food item had characteristics (calories, protein, sugars, etc.) like Comic Book characters do (Intelligence, Strength, Speed, etc.), we considered having one of our idioms similar to the one found on this visualization, precisely the Parallel Coordinates plot.

Additionally, in the introduction of this report, we made an allusion to an Overweight and Obesity article published by World Health Organization [1], something that motivated us to search more about the topic of “healthy” vs “unhealthy” food. Consequently, we found a visualization [4] about the characteristics of fast food produced by McDonald’s. For us it made sense that the data was represented in that way, separated by groups with each group having a unique color hue. There is a column for each attribute (y-axis) and an item is a line that crosses each column. Although having a much smaller domain, this visualization can perfectly encapsulate the goals we proposed for our project.

Having both these ideas into consideration we decided to implement the Parallel Coordinates plot since it would be a nice approach for this type of data.

It is worth mentioning that a list with the items currently shown in this visualization was part of the initial idea and prototypes since it would be helpful for the user to have a better perception of the items that are currently visible. The list would enumerate the names of the food items and it would highlight that specific item in the Parallel Coordinates when hovering over an item. This idea, however, was dropped due to the overall space of the visualization only allowing the list to display a small amount thus not fulfilling its original role. The data represented would be incomplete and redundant.

To implement the decided plot, we decided to look for multiple examples online. Since we were using D3.js (a JavaScript library for producing dynamic, interactive data visualizations in web browsers), we went directly to its source and started searching for some examples. We found a website that presented a gallery with hundreds of charts [5]. In this gallery, we found multiple examples of Parallel Coordinates. These plots were capable of highlighting items, filtering and dividing items per groups.

We found, in this same website, an example of a Parallel Coordinates graph that worked with a domain similar to ours and had some features that caught our eye [6]. Since its characteristics met our pretended ones, we decided to base our model on this example, adjusting it to our extra desired features while removing the unnecessary ones, such as adding hoverability to the lines.

This visualization was not enough since we intended to let the user choose categories and compare them not only between each other but also with a reference value. We wanted the user to be able to highlight each specific item and compare it based on its group but also individually. We understood that we would need more information or even more idioms to complement this visualization. Some linked idioms and filtering options would improve our work.

Since we wanted each item to have a category and a subcategory with each category and subcategory aggregating multiple items, we searched for idioms that represented this domain in a clear way. After an exhaustive search and discussion between options, we decided that a Treemap would be the best option. A Treemap is a diagram representing hierarchical data in the form of nested rectangles, the area of each of them corresponding to its numerical value.

To implement this idiom, we went back to the D3.js Graph Gallery [5], now searching for Treemaps. Once more, in this gallery there were many examples relevant for our design and implementation of this idiom. We intended to have a customized idiom with the item categories and their subcategories present in each category.

Finally, in order to enable the user to compare each item with a reference value, we designed multiple Jitter plots. These plots are useful for representing hundreds of items, dividing them by similar characteristics. Then, it is possible to understand the dispersion of the items having one attribute into consideration.

To implement the Jitter plots, we searched for examples already made by others online and we came across two great examples [7][8]. Both these examples refer to Violin Plots, an interesting type of plot that is slightly more complex that the Jitter plots that we pretended to implement. Since our data would have multiple categories and therefore the x-axis would have multiple labels, the violin plots would be confusing and not as appealing as Jitter plots.

With this being our final visualization, the user would not be limited and would be capable of answering not only the proposed questions but also many more of the user’s interest in a simple and obvious way.

3. THE DATA

We looked up different datasets that had a diversity of food items and its characteristics. Our desired dataset would not only have the most popular attributes for each item (like calories, protein, carbohydrates, etc.) but also others that we found relevant and significant. After analyzing multiple options present on the internet, we came across this dataset on Kaggle, an online community platform for data scientists and machine learning enthusiasts.

The dataset had all the nutritional values for around 8.8k items (food products). It is a table with a total of 77 columns, most of them containing a single value and its unit of measurement, which represents the quantity of each component.

We did some modifications in this dataset in order to have the data described in a way that was in line with our goal.

Although very detailed, accurate and with the desired characteristics, this dataset still contained some columns that are not as interesting given our goal with this visualization. Consequently, we decided to consider only the most relevant and known columns as well as the ones that we decided to include because of their importance in answering the questions we propose in the first section.

Having our dataset with only the pretended attributes for each item, now we had to deal with the missing values. To tackle some of the missing values, we considered mathematically computing them but since it was a negligible percentage of our dataset (around 6%) we decided to just remove them.

For outliers, we decided to keep every item since there are in fact items who have extremely high quantities of certain attributes (e.g. oils are composed essentially of fat, some of them around 99%).

By doing these procedures, we noticed that there were many food items with nearly the same composition and name, while others were very specific and negligible towards the end goal of this project, like sauce from a very specific brand, so we decided to remove them.

All the approaches referenced above were done using the pandas Python Data Analysis Library [9].

Two of the most relevant additions were the two new columns that classify each item regarding its category and subcategory. The category was defined by hand after defining all the possible categories. Because of the density and specificity of some of the item´s names, this step took us a good amount of time and effort. The subcategory column, also called "type", was obtained by splitting the food item's name by the first comma since every name has a pattern. The remainder was characterized as a description of the item.

Previously there was not a great way of grouping items of the same category to visualize them as a group, this modification proved to be important.

In addition to our main dataset, we decided to include a reference table that has the recommended daily intakes of the calories, macronutrients, sodium, potassium, and sugar for an adult, in order to permit comparisons.

Regarding the derived measures, various columns were added after cross-referencing the values from the dataset with the values from the reference table, after calculating the percentages of each component per item.

Using these derived measures, we decided to compute some measures comparing the amount of a certain attribute present in an item with the recommended daily intake of said attribute. These measures are useful to have a direct contrast between different types of products, in particular those who are labeled as "healthy" or "unhealthy". We also added percentages of some attributes for each item.

It is important to note that some values of the dataset (e.g. calories) are already derived data (can be calculated taking in consideration the grams of protein, fat and carbohydrates).

After cleaning the dataset, everything was perceptible and there was nothing that we wanted to find and didn't. As expected, all the data fields were float values (ratio variables) except the names, types, descriptions, and categories.

In terms of scalability, the paradigm adopted by us permits the addition of more items to the dataset if the items have the specification of the value for the needed attributes.

Furthermore, we also needed to derive a file from the original data table, with the categories, subcategories, and their quantities, to be read by the Treemap idiom.

Summing up, the final dataset is simpler and easier to read compared to the initial one. It aims to correctly answer the proposed questions as well as others that make sense in this domain.

4. VISUALIZATION

4.1 Overall Description

On the top of the interface, we have the header of the page with the title of our project, a counter and progress bar showing the number of the items currently rendered and visible and a header for the Treemap.

Underneath, we have our first idiom: the Parallel Coordinates plot that shows all the items that respect its filters. The items are encoded by lines that cross every attribute column.

On the right, there is the Treemap. The user can observe 10 rectangles with sub rectangles representing each Category and its Types. The size of the rectangles is proportional to

the quantity of items in that Category/Type. The types are grouped per category and have the same color hue.

Lastly, at the bottom left, we have six Jitter plots, each one comparing the items of the 10 Categories by a specific attribute. Every plot shows the recommended daily intake of that attribute per serving, i.e. 100g, using a dashed orange line.

In the initial state of the visualization, all the items are shown. The Parallel Coordinates shows 1640 food items that can be filtered according to the 15 nutritional attributes.

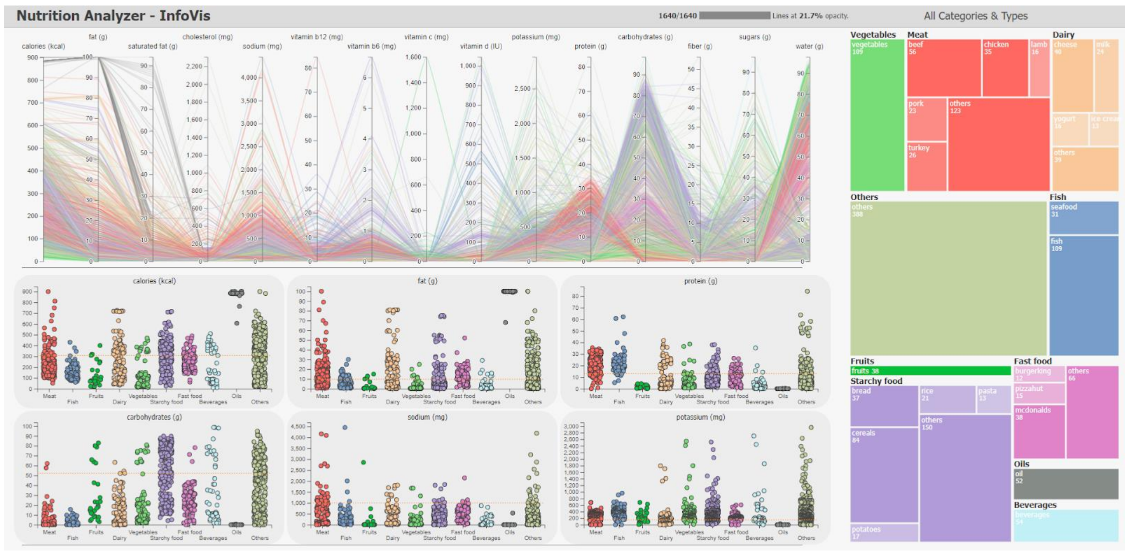

Figure 1. Final Project layout

Each horizontal line mark represents one item and its color hue encodes the items category. The y-axis for each attribute has a scale using the units presented in the title of the axis. Each item crosses that axis on the value corresponding to its attribute value for that specific attribute. Regarding the filters, each one of the attributes y-axis can be filtered by positioning the cursor above the axis and dragging to create a vertical rectangle that represents the filter. By doing that, not only are the lines in the plot filtered but the items/points shown in the Jitters.

The Treemap shows the composition of the dataset by dividing it per categories. Each category is represented with a rectangle mark and its channel, size, represents their affluence in the dataset. Each category is encoded by its unique color hue. When the user hovers over a rectangle it shows a gray border. The user can press a rectangle that corresponds to a Category and that selection propagates to all the Jitter plots highlighting that category columns and blurring the non-selected ones. The user can also select multiple categories in order to filter the ones that he pretends to analyze. When a Category is selected a darker border appears on its rectangle to mark the selection.

For the six Jitter plots, we chose to represent one plot for each of these attributes: calories, fat, protein, carbohydrates, sodium, and potassium since all of them have a reference value on the reference table. Each plot has all the Categories encoded in the x-axis and shows the distribution of the items given their quantity of that same attribute in the y-axis. Each Category is encoded by its unique color hue, the same as its color in the other two idioms. When the user hovers over a circle of selected Category, the item is highlighted in every Jitter plot and its name is shown near the cursor. If the Category is not selected, it shows a hint asking the user to select the Category first. The user can also select or unselect a Category on the Jitter plots by clicking on it. These changes propagate through all the idioms.

All of this can be represented for example by 15 bar charts, each on representing each attribute, but if the user wants to compare different categories or even types of food, it would turn out confusing and having 1640 bars per chart, the dataset would not be used in its maximum utility.

4.2 Rationale

In our prototype, the idioms have and can link between themselves. It was a clear and easy way to understand the domain and in a way that gives the user the freedom to do all types of comparisons between types or categories of food items.

As referred above, we used three different idioms: Parallel Coordinates, a Treemap, and six Jitter Plots. There are links between each other. As we explained in the overall description, every idiom has its functionality, all of them show data and are not only used to filter measures.

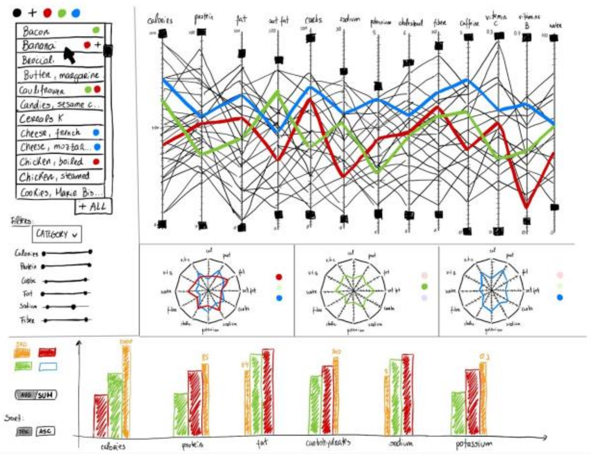

Figure 2. Initial Layout Idea

Initially, as shown in Figure 2, instead of the Treemap, we thought of having three Star plots to give more detailed information about the selected groups of foods. These groups of foods were formed by the user in a list next to the Parallel coordinates. For every Star plot, there was a button for each group (with its color) that could be pressed to show the group characteristics that specify the star plot. It was possible to compare multiple groups in the same plot as well as compare them in different star plots. But further on we gave up on that idea because this process would be confusing and redundant since it allowed making the same comparison in the three different Star plots.

We also had, in line with the possibility of forming groups, a Multiple Bar chart, with each attribute being the sum of all food items selected in each group. Then the graph would compare the created group´s attributes between each other and their recommended daily intake. We could also sort the plot in ascending (ASC) or descending (DESC) order. We had the option to see the average values of that group instead of the sum. There was a possibility of selection between which groups would appear in the chart. But as explained before, the whole selection of the group process could turn out to have unnecessary steps that could be sorted out in a simpler way.

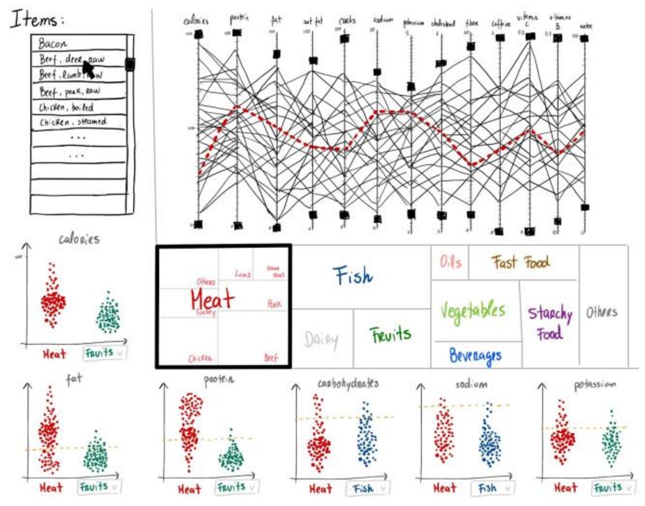

Figure 3 Final Layout Sketch

After a discussion about what could be changed, we came up with this layout idea that would turn out to be close to the final prototype.

Instead of allowing the user to create custom food groups the Treemap divides the dataset per categories while also conveying the size of each of them to the viewer in an easier to perceive way. Each category is represented with a rectangle mark and its channel, size, represents their affluence in the dataset. The user can press a rectangle that corresponds to a category and the selection propagates through all the plots. Each category is encoded by a color hue, repeated in every idiom. When a Category is selected a border appears on its rectangle and it starts showing the Types in that Category. Also, when a Category is selected, instead of showing all the items, the Parallel coordinates plot shows only the items in the Category. As for the Jitter plot, the left value on the x-axis (the fixed category) changes to the selected category.

Finally for the six Jitter plots, we chose to draw six plots, one plot for each attribute that has a recommended daily intake (calories, fat, protein, carbohydrates, sodium and potassium). Each plot has two categories encoded in the x-axis and shows the distribution of the items. Each item has its percentage value of that attribute (excluding the calories attribute in which the quantity of that attribute itself is shown) encoded in the y-axis. For every plot, the Category represented at the left is "fixed", that is, it is the selected category in the Treemap. The right one has a dropdown menu that lets the user select a Category to compare with the fixed one. Both these "columns" show all the items if no categories are selected. The recommended percentage of each attribute per day (DRD %) is represented by an orange line in the plots, except for the calories since it's much higher than the calories of each individual item and has no relevance. Also, by clicking in a point from the "fixed" Category in any Jitter plot its Type is selected and this filter propagates to every other plot i.e. Parallel coordinates, Tree Map and the other Jitter plots. When items are filtered in the Parallel coordinates, the sizes of the rectangles for the selected category (Types) can change if the proportions change.

After these two layout sketches, we put together the first real version of the prototype, similar to Figure 4.

Even though it was a simple and cohesive idea, we tried to make the user have to take as few steps as possible to make any kind of comparisons and have as much freedom as possible.

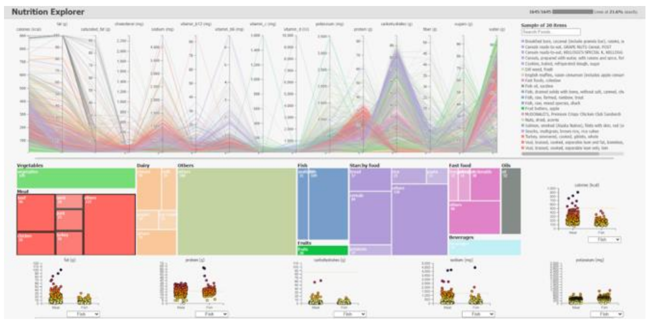

Figure 4. First Prototype

Looking at the final prototype (Figure 1) the list and the dropdown menus for the jitters were removed. Apart from the Treemap and the Parallel coordinates, the six jitter plots now have all the 10 categories in the x axis and can be compared directly. Every plot shows the recommended daily intake of that attribute per serving, i.e. 100g, using a dashed orange line.

Each Category has its unique color hue, the same in every idiom. When the user hovers over a circle of selected Category, the item is highlighted in every Jitter plot and its name is shown near the cursor. If the Category is not selected, it shows a hint asking the user to select the Category first. The user can also select or unselect a Category on the Jitter plots by clicking on it. These changes propagate through all the idioms, including the Treemap. In the initial state of these idioms, all the categories are unblurred in the Jitter plots and unselected in the Treemap, i.e, there are no default selections, and the idioms show all the items.

4.3 Custom Visualization

We started by choosing a Strip plot to represent our data. This idiom is a single-axis scatter plot that is used to visualize the distribution of numerous individual one-dimensional values. These values were plotted as dots along the axis, with many of them overlapping. Because of this, it was very hard to perceive the items in the graph. We then discovered that a possibility to fix this issue was to make it similar to a Jitter plot. This plot is better to visualize the distribution of many individual one-dimension values, as is our case.

Inspired by this, we expanded the old idiom, creating columns for every category and randomizing the position of each point along the width of every column. To further customize this idiom and allow us to answer some of our questions, we added a column that references the attributes recommended daily intake per serving of 100g in each of the plots. This addition allows us to observe how each item compares to its recommended daily intake of every attribute, a feature that allows us to make judgements about the quality and healthiness of food items. We also wanted the user to have a more comprehensive and clear way to see the categories and food items themselves, corresponding to the 6 most relevant attributes. By hovering through the points in the jitter, it shows the food item name and clicking it can filter the Treemap and the Parallel coordinates by its category. Also, each category has its own unique color representation on every idiom, which facilitates the understanding and the importance of categories and the dataset itself.

4.4 Demonstrate the Potential

Our Prototype has the comprehensive idioms to answer any question the user has about this topic and the dataset itself.

Considering the initial state of the prototype in Figure 1.

Considering the questions we proposed initially, the user pretends to obtain the answer to the question ”Does fiber rich starchy food tend to have less cholesterol?”. For that, the user needs to select the Starchy Food category in the Treemap or in one of the Jitter plots. This selection highlights this category in the Jitter plots and filters the items in the Parallel Coordinates.

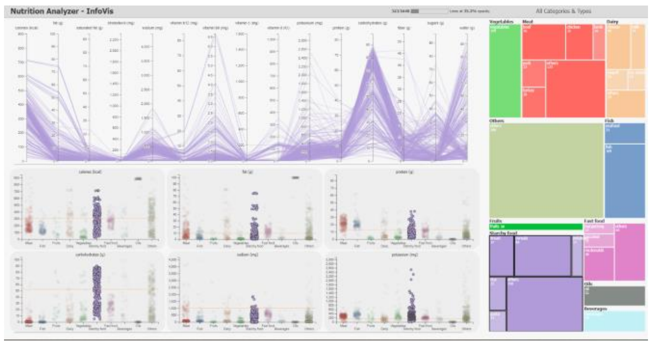

Figure 5. Selecting Starchy Food Category

Then create a filter in the fiber’s column of the Parallel coordinates by clicking and dragging the cursor over the axis. The filter can be resized as well as moved by the user to obtain more/less items. A good example would be a filter between 25g and the maximum.

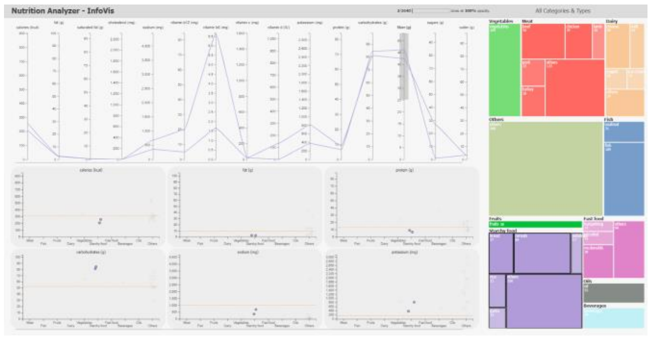

Figure 6. Filtering Amount of Fiber

With the filtered items, we can observe that the items with high levels of fiber tend to have almost no cholesterol by looking at the intersections with the cholesterol column in the Parallel Coordinates plot. It is worth mentioning that if we wanted to know about the other attributes, not only could we inspect their columns in the Parallel Coordinates plot but also have in consideration the Jitter plots. Given a specific set of filters, the user can compare the selected items with the recommended daily doses per serving and also have an idea of how the other items behave (dots with less opacity in the Jitter plots).

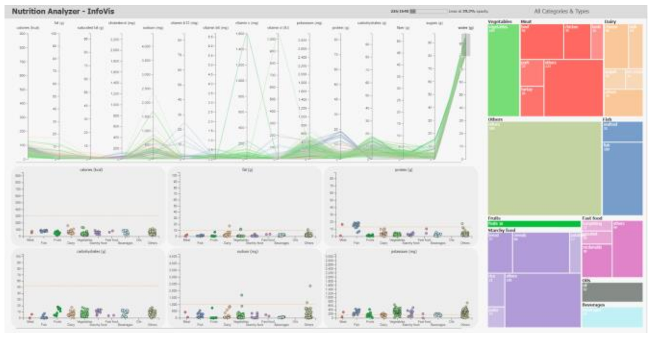

Figure 7. Is water rich food healthier?

For example, if the user wants to know if water rich food is healthier, he has to adjust the bounds on the water attribute in the y axis to higher values (80g-100g) and compare the remaining values in the Jitter plots with the daily intake which is the orange dashed line. Taking the user only 1 step/interaction to answer this question.

As you can see, in most of the cases, the food items with a great amount of water have a low value in calories, fats, proteins, carbohydrates and sodium when compared to the respective daily intakes. Although these foods normally have a high potassium value compared to its daily intake, the user can conclude that water rich food is indeed healthy.

A fun fact that we discovered by testing all the possibilities and comparisons, we selected the categories Vegetables and Fruits in the Treemap and adjusted the bounds to higher values (200 kcal - 500 kcal) in the calorie y axis in the Parallel Coordinates.

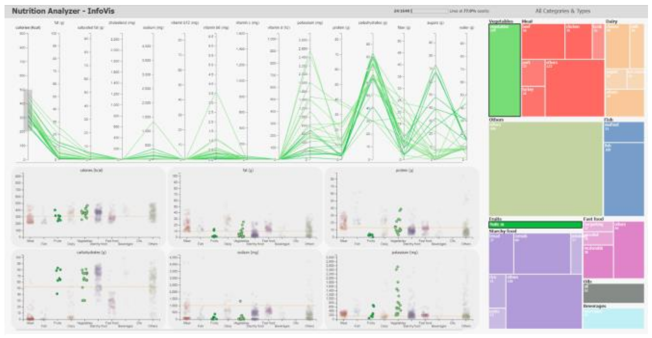

Figure 8. Vegetables being unhealthier than fruits

We concluded that, in the majority of the cases, vegetables are unhealthier than fruits. Only by exploring even more combinations and filtrations could we find a lot more facts that were unknown.

5. IMPLEMENTATION DETAILS

To deal with our dataset, we first had to convert most values to floats. After having every value in the correct format, we started using them in every visualization.

The most challenging idiom was the Parallel Coordinates plot. We decided to base our model on the ones presented in the D3.js Graph Gallery [5] adapting it to our additional features and improving some of their aspects. In the initial state of this plot, every item is represented, and the float value of every attribute is used to encode the position of the intersection between its line and the attributes column. When a filter is created, moved or resized it is added to a map of filters. A brush event then is called. This brush event starts by cleaning the plot and then takes into consideration every item that respects the applied filters and redraws those lines. These new filtered items are saved in a variable. This new variable is used to filter the items present in the Jitter plots. Only the items that respect the filters are presented in the Jitter plots. If an item is added to a Jitter plot after a filtering event, it shows an animation of it going up to its position.

To optimize the drawing of the lines and inform the user about the status of the brush event, we used some additional functions to delay, animate and provide feedback to the user. For instance, the progress bar and counter at the top of the page are responsible for giving feedback about the total of items rendered and visible throughout the brushing event. All the update functions present a delay for the user to see the newly added/removed values.

Regarding the selected categories, their names are also stored in a variable when the user clicks on the Treemap or on the Jitter plot category. This variable is used to filter the domain used in the Parallel Coordinates, showing only the items from the selected categories, again, by calling the brushing event.

When a category is selected, only the items of that category are shown in the Parallel Coordinates plot, however, in the Jitter plots, all the items that respect the existing filters appear. The ones from the selected categories are presented with full opacity/highlighted while the others have their opacity reduced, reinforcing that they are not selected. If the user hovers over an item in the Jitter plots, if the item is highlighted, it shows its name. Otherwise, it gives an insight asking for the user to select its category.

For the Treemap, we had some difficulties processing the data in order to further improve our idiom. This idiom requires a data file in a specific format in order to create a hierarchy for the categories and types. To update the idiom in real time, we needed to be constantly adjusting the values of the existing types to redraw the rectangles.

We did not use any algorithm to efficiently filter the data since the current model seemed to be working smoothly without them. Since our focus was not to parse the data with efficiency, we decided to focus on improving our idioms as well as creating new useful features.

6. CONCLUSION & FUTURE WORK

With this project, we´ve learned all the implications of making a cohesive and good visualization. From picking a dataset with potential, to processing that group of data in order to reach an easier and perceptible visualization, to brainstorming ideas of layout and extensions of so many idioms, to implementation seeing in practice what would work with the dataset. All of these small steps came to be useful to reach a good final prototype.

Although we didn't show in this report all the answers to our questions, we did indeed have a response to all of them. The user can reach it in a fast and efficient way using no more than 2/3 steps.

Initially we had only fewer days to choose the dataset. Although it's an important topic, in this dataset there is no variety in the attributes' types, as referred before, four of them are nominal and the rest is ratio. We also faced many difficulties trying to add nominal attributes to our dataset, such as the food categories and types, having to do most of it manually which was not optimal and took a lot of work. Fundamentally we would've picked a more interesting and diverse dataset, while also thinking more clearly about difficulties during implementation.

If we had more time and more resources to develop our solution, we would change our Treemap to be dynamic and zoom in when a category is selected showing the types of the food items. Also when the user filtered the upper and lower bounds in the Parallel Coordinates, the Treemap would only show the categories/types of the food items present in the Parallel Coordinates.

It is also worth noticing that the dataset we chose is referring to foods that can be found in North America so the items and attributes can be quite different from the ones we are used to in Portugal and Europe. One further expansion to this project can include a revamp of our dataset to include food items that are common in Portuguese supermarkets.

REFERENCES

1. One Billion People Globally Estimated to be Living with Obesity by 2030.
   Retrieved November 06, 2022 from
   https://www.worldobesity.org/resources/resource-library/world-obesity-atlas-2022
2. Obesity and overweight.
   Retrieved November 06, 2022 from
   https://www.who.int/news-room/fact-sheets/detail/obesity-and-overweight
3. Hall of Fame, Information Visualization 2022/23.
   Retrieved November 06, 2022 from
   https://pcm.rnl.tecnico.ulisboa.pt/moodle/mod/page/view.php?id=3055
4. Percentage of Recommended Daily Intake.
   McDonald's. Age group 4-8.
   Retrieved November 07, 2022 from
   https://visualizationdesign.wordpress.com/2015/05/31/group-10/
5. The D3.js Graph Gallery
   Retrieved November 07, 2022 from
   https://d3-graph-gallery.com/index.html
6. Parallel with full option. D3.js Graph Gallery.
   Retrieved November 07, 2022 from
   http://bl.ocks.org/syntagmatic/3150059
7. Violin plot with jitter in d3.js.
   Retrieved November 07, 2022 from
   https://d3-graph-gallery.com/graph/violin_jitter.html
8. Gradient Jitter Violin Plot.
   Retrieved November 07, 2022 from
   https://bl.ocks.org/motttey/339c1a5ec41c2c3b5da8
9. pandas - Python Data Analysis Library.
   Retrieved November 07, 2022 from
   https://pandas.pydata.org/docs/