VI Project - City Ranking

Pedro Monteiro, 97484
Information Visualization
Computer Science MSc
DETI, Aveiro University
Aveiro, Portugal
pmapm@ua.pt

Eduardo Fernandes 98512
Information Visualization
Computer Science MSc
DETI, Aveiro University
Aveiro, Portugal
eduardofernandes@ua.pt

***Abstract*—In this report it is possible to observe different ways to represent data related to quality of life in different cities around the world. It also allows some interactivity between the application and the user. All data utilized comes from a kaggle dataset [1].**

***Keywords*—data visualization, cities dataset, bar chart, line chart, radar chart, map chart, lollipop chart**

I. Introduction

This article comes within the scope of the first VI [2] project, where the objective is to apply different ways of visualizing the data, which are appropriate to the type of data that is intended to be visualized.

All the code developed can be found in our github repository [3].

II. Motivation and Objectives

Before traveling or moving house, everyone has the concern to check the destination city as well as the living conditions practiced there. If a person wants to see some information about a certain city, they probably would search on google to see what are the main aspects of that city. This means that there is not an available platform or a software where such aspects are condensed in one place. To solve this problem, we introduce *City Ranking*, where users can find multiple informations regarding a city. Our solution aims to provide multiple views to show all city aspects.

Mainly, our application was developed thinking about two main objectives, those being able to check different aspects about a certain city and also being able to compare at least two different cities, in order to make a more direct comparison between each one. The visual representations implemented, try to fulfil this necessities.

III. Users and Questions

As our application is focused to be used by users in any age, it should probably be designed in a way that all users can understand all the data displayed.

This generates a good question, that is *"What is the best way to design our application in a way that is easy to be used by everybody?"*, in other terms, the best way to design a user-friendly application.

In order to represent the users in our platform, we created two personas, each one described in the sections below.

A. First Persona Case

This persona name is *João Martins* and hes has 20 years old. His goal is to find a city to travel where the prices and crime rate are low. This leads us to several questions that this type of user might want to be answered by our representations.

One of our first thoughts was to being able to display the most data that we can, so that the user can check several aspects on the various cities. However, this foments other question that is, *"How are we going to display information about more then 200 cities?"*.

Supposing that the user is able to see all this information, *"How can the user find the most suitable city based on all that information?"*. Also, assuming that the user chooses two cities, *"How can the user decide which city to choose?"*.

B. Second Persona Case

This persona name is *Marta Silva* and she has 35 years old. She will travel to Istanbul, so her goal is to see what is the quality of life in there.

Although, it does not seem to generate a lot of questions when thinking of a solution for this problem, it turns out not to be as straightforward as we thought.

The main question may be *"What is the most adequate visual representation to display information about a certain city?"*.

falar acerca de facilitar a experiencia do utilizador, fazendo uma aplicaçao user-friendly; aplicação single-page; minimalista(simples)...

IV. Dataset

As previously mentioned, the dataset is from Kaggle [1]. This dataset contains information about 216 cities around the globe, and it consist of:

- **City:** the city name
- **Purchase Power:** value that is a comparison between the average cost of living with the average local wage
- **Health Care:** value that shows how citizens feel about their access to healthcare, and its quality
- **Pollution:** a score of how polluted people find a city, includes air, water and noise pollution
- **Crime Rating:** the lower the score the safer people feel in this city

- **Quality of Life:** a balance of healthcare, pollution, purchase power, crime rate to give an overall quality of life score
- **Movehub Rating:** a combination of all scores for an overall rating for a city

It is also possible to find another file that gives us information about the prices of some products, measured in GBP for each city:

- Cappuccino
- Cinema
- Wine
- Gasoline

V. Low Fidelity Prototype

This earlier prototype was done with paper and pencil, because in our point of view it is much more simple. It was created in order to try to answer some of the questions presented above.

Initially, we started to develop our application trying to solve the *"What is the best way to design our application in a way that is easy to be used by everybody?"* question. Like some previous projects, we started by defining that the software should probably be a single-page application. This measure was taken to provide to the users a more easy and direct way to interact with the website. For the user to navigate it was created a side-bar that works as a control center for the program so that he user is able to change between several visual representations from here.

In order to answer the question*"How are we going to display information about more then 200 cities?"* we thought that a Heat Map combined with a Map Chart would be the best approach to solve this issue. Cities would be categorized by country and then different countries would have a different color according to intensity of the variable being analysed. For the question *"How can the user find the most suitable city based on all that information?"* we thought that a simple bar chart with the cities would solve this problem, but after some user feedback we realized that it may not be sufficient. For this fact, it was implemented a new feature in the Heat Map, that is when a user clicks a country, all the cities available are displayed with their respective value for the current variable being studied. After that, we chose to keep the bar chart in order to allow a diversified view for the user.

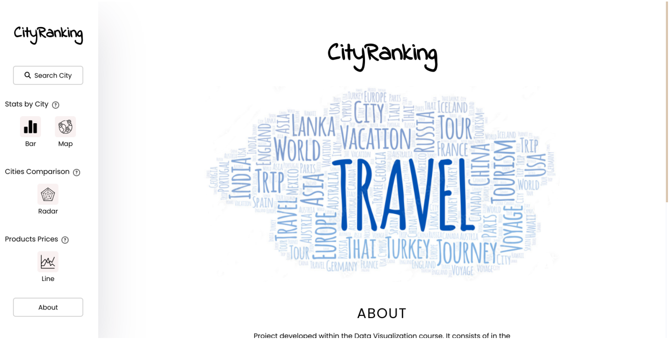
The last question answered in the low fidelity prototype was *"How can the user decide which city to choose?"*. Until then, the application did not allow making direct comparisons between the information of each city and, in order to answer this question, a view was created that contains a Radar Chart. The only question that we didn't answer in the this prototype was *"What is the most adequate visual representation to display information about a certain city?"*. As a result, during the development we were forced to implement a visual representation for this inquest however, in our opinion the result is good as the others.

VI. Functional Prototype

The developed solution has only one page, with different views.

A. Home View

This is just a simple view to show the title of the project, as well as a little about the context of the project and also the team that developed it.



Project developed within the Data Visualization course. It consists of in the

Fig. 1. Home View.

B. Search City View

In this section it is possible to search for a city, and if that city is present in the dataset, the data about it will appear in a lollipop chart.

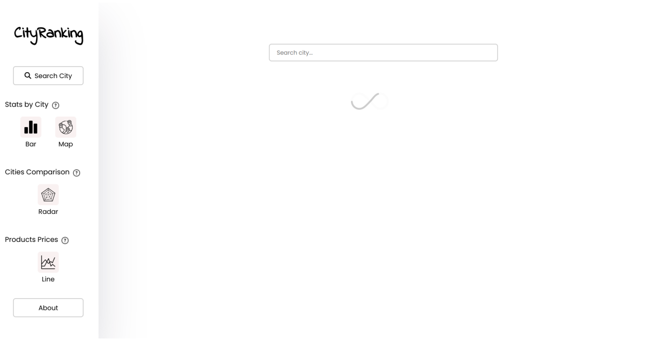


Fig. 2. Search City View.

As you can see in Figure 2, there is some feedback to let user know that he can enter a city in the search bar.

When starting to write the name of a city, an autocomplete appears that allows user to understand the existing cities in the dataset, and being able to simply select the one he wants to search for, instead of having to type everything.

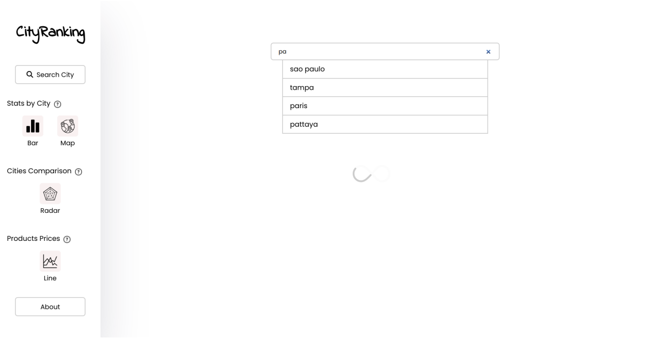


Fig. 3. Search City View Autocomplete.

Through figure 3, is possible to see that autocomplete was implemented in a way that it's not necessary to write the cities with capital letters, that is, the autocomplete is not case sensitive.

When selecting a city, a lollipop chart appears with the data related to that city.

This page basically seeks to answer the question ”How can the user find and see the information about a specific city?”. In order to answer this question, a graph of this type was used, being possible to quickly and easily check the values practiced in the selected city, for example, the quality of life practiced, or even the levels of crime or pollution in that city. It's another way of representing the data of a city without using a bar chart, which, as will be discussed, was used in another view.

By hovering with the mouse on one of the graph points, it's possible to observe the real value, as represented in the figure below.

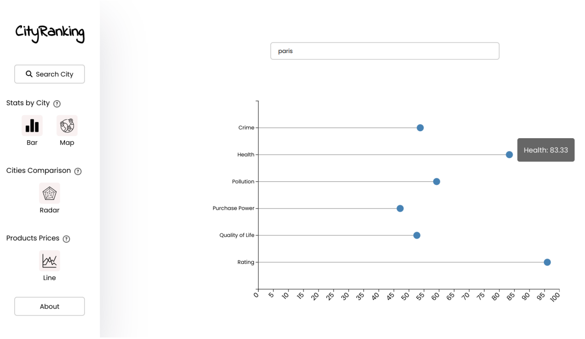


Fig. 4. Lollipop Chart.

C. Cities Comparison View

”How can the user decide which city to choose?”, this was the basic and most crucial question for the construction of this view.

As you can easily see, when looking for a city to live in, or even to spend some time, whether on vacation or temporarily for other reasons, there is a need to make comparisons between different cities.

In this section, comparison between two different cities is allowed, making it easy to see which city has the best and worst values in terms of quality of life.

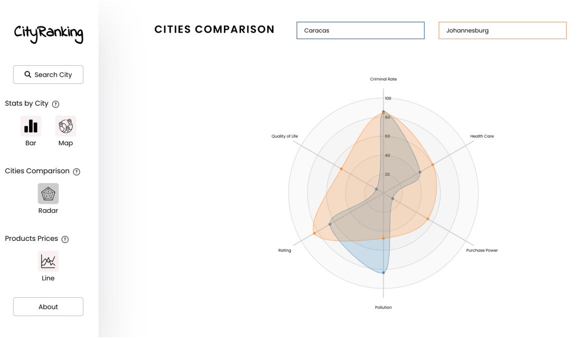


Fig. 5. Radar Chart.

The main component in this section is the radar chart. In the implementation of this chart, a d3 library was used [4], which made the implementation of this chart simpler and faster.

The data contained in it can be changed through the inputs at the top of the page. Similar to other pages, autocomplete has been implemented in these inputs so that a city can be selected more intuitively. It is possible to distinguish the cities through the color blue and the color orange.

In the graph it's easy to see and quite automatic when a city has a better quality of life. Looking at figure 5, you can see that Caracas has a higher level of pollution, the crime rate is quite high and consequently, the quality of life in this city has a very low value.

On the other hand, Johannesburg, represented by the color orange, has higher values for health care and purchase power, with lower pollution levels, which makes it a city with a better quality of life when compared to Caracas.

D. Products Price View

In addition to data related to the quality of life in cities, there is also another file that allows to observe the prices of some products, although the dataset contains few data.

In this view, a line chart was implemented, where the main question to be answered is ”The price of product X is higher in which city?”.

The main workflow in this section is to insert a city in the input, and automatically the prices of available products from that city are placed in the graph. It is possible to add up to 8 cities, and the axes resize as soon as higher values are added. By clicking on the name of the city, in the upper left corner of the graph, it is possible to hide the line, so that sometimes, if necessary, you can have a better view of the products.

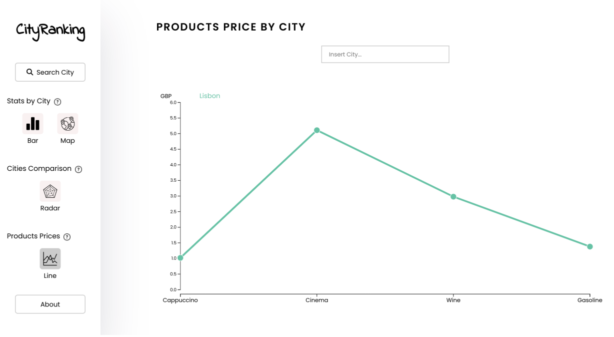


Fig. 6. Line Chart with One City.

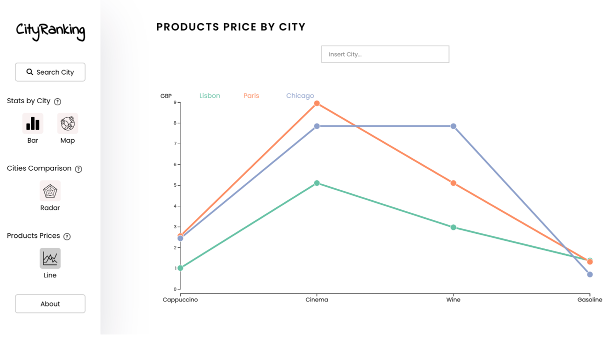


Fig. 7. Line Chart with Three Cities.

Looking at figure 7, it can be seen that Paris, represented in the graph by the color orange, is the city where the cinema has the highest price, when compared to Lisbon and Chicago. Looking at wine, it's more expensive in Chicago, with blue color, and cheaper in Lisbon, with green color, when these 3 cities are compared.

Starting from the question asked initially, it's then possible to verify that is easily answered through the use of this type of graph.

One of the limitations we dealt with was the fact that we were unable to implement a way to allow the user to remove a city, for example, in case it was added by mistake, which is a feature that can be useful sometimes.

E. Stats by City Views

This section is related to the views that display all the statistics about the cities. As it was already mentioned in this paper, in order to answer the question *"How are we going to display information about more then 200 cities?"* two views were created.
In both views the user can select the dependent variable from the upper right drop-down.

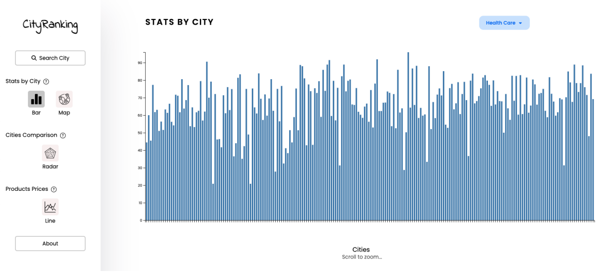


Fig. 8. Bar Chart.

1) Bar Chart: A Bar Chart was implemented to list all cities available from the dataset.

More than 200 cities appears to be to much to put in this type of chart, however the main idea is to let the user explore all the information that is displayed to him. For that reason, it is possible to zoom in and also to move the graph horizontally. Bellow each bar, is the name of the city and when the cursor is hover one of them, it shows the accurate value of the dependent variable.

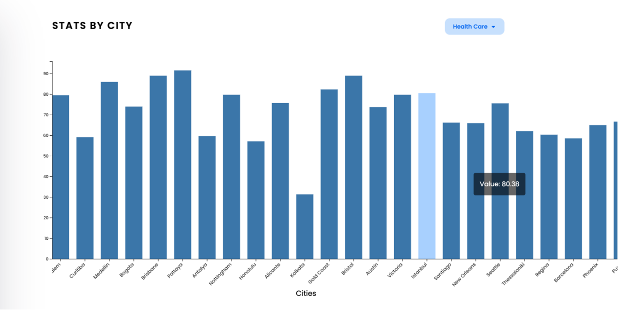


Fig. 9. Bar Chart zoomed in.

2) Heat Map: Through a heat map where the world map is represented, it was possible to show the data coming from the dataset.

This map plots the data according to a color scale, with the darkest color being in locations/countries where cities contain the highest values. The scale that represents the color can be found in the lower left corner of the map, as exemplified in figure 10.

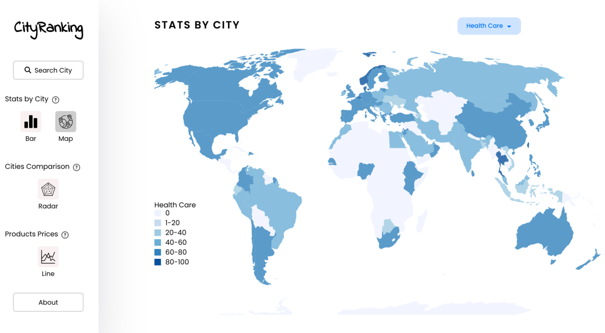
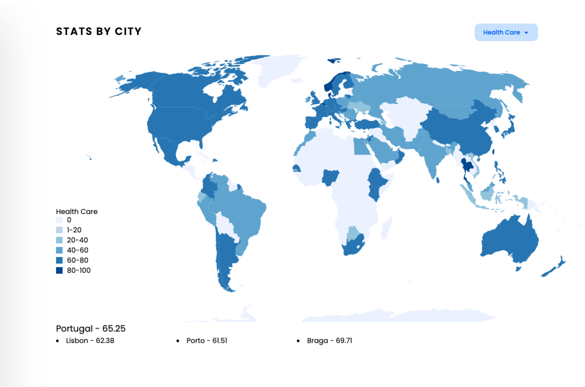


Fig. 10. Heat Map.

It should be noted that it's possible to filter what is being shown on the map, through the dropdown at the top of the section. When changing, is possible to verify that the colors are also changed according to the new values. Also, when hovering over a country the user receives some feedback.

In addition, when the user clicks on a specific country, the system shows the data of cities that are available in the dataset, as can be seen in figure 11. Just highlight the fact that it shows the average of the values for the cities belonging to the country, next to the country name.



Portugal - 65.25

- Lisbon - 62.38
- Porto - 61.51
- Braga - 69.71

Fig. 11. Heat Map Onclick.

VII. Conclusion

At the end of the project we were satisfied with the final result, having a functional prototype that can handle the scenarios designed for our personas, as well as respond to the needs of the users.

It should be noted that having made a prototype on paper and discussing it with some colleagues helped a lot to understand how we would build our application.

The heuristic evaluation also allowed us to find some flaws in the application that could be easily corrected in future work.

VIII. Appendix

The application has been deployed and is accessible at https://vi-city-ranking.netlify.app/.

Project Contribution Percentage:

Pedro Monteiro: 50%

Eduardo Fernandes: 50%

Acknowledgment

We would like to thank Professor Paulo Dias, regent of VI course, for being available to answer questions and also to Professor Beatriz for being present at our presentation.

References

[1] Dataset used https://www.kaggle.com/datasets/blitzr/movehub-city-rankings?select=movehubqualityoflife.csv
[2] Information Visualization https://www.ua.pt/pt/uc/6495
[3] Github Repository https://github.com/pedromonteiro01/city-ranking
[4] Github Repository https://www.npmjs.com/package/react-d3-radar