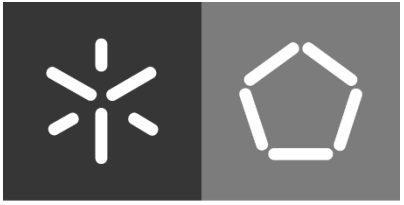
Universidade do Minho
Escola de Engenharia

Carlos José Soares Ferreira da Cunha

Cycling Analytics: Visualization of Cycling Data and Patterns

December 2021

Cycling Analytics: Visualization of Cycling Data and Patterns
Carlos José Cunha
UMinho|2021



Universidade do Minho
Escola de Engenharia

Carlos José Soares Ferreira da Cunha

Cycling Analytics: Visualization of Cycling Data and Patterns

Master's Thesis
Integrated Master's in Engineering and Management of Information Systems

Work Supervised By:

Professor Rui João Peixoto José

December 2021

Dissertation Acknowledgments

University is the final destination in a journey of studying and academic pursuit for a teenager, and whilst it's focus is to teach and learn in the chosen fields of study, the real lessons are the ones taught outside the classroom. From the values learnt, to the experiences had, and the friends made along the way.

The writing of a dissertation marks the end of the journey, and unfortunately this ending step was during the pandemic, that took over the world. Robbing these last years of joyous exploration and growing of a person. Nevertheless, this section is dedicated to thanking everyone that made going through it better.

First and foremost, I would like to thank Professor Rui João Peixoto José, for guiding me through this dissertation, for always being available and showing care for the work done. With him I was able to foster a new love and interest for Analytics that I hope I can carry on for the future. Alongside the P25 team that provided help and support in any endeavors concerning the project.

Secondly, I want to thank my loving family, mainly to my mother for being an everlasting source of strength and motivation. A constant shelter where I could find all the support I needed during the toughest times, that never faltered to anything. To my sister for testing my patience and refining my temperament for better or worse, and I hope you grow to be happy and successful in whatever ways you desire. And to my father, for being larger than life itself, no matter where he is, I will always carry on the pride of being called, his son.

Finally, no journey is meaningful without the friends that were in it, and this journey had plenty of them. They filled these last 5 years with many moments of joy and goofery, never letting me have a moment of peace and tranquil, I want to thank you all. Thank you, Luisa and Ricardo, for being there through thick and thin, and making this journey unforgettable. Without them it's hard to imagine how else things would've been, and I hope the future that awaits you is as bright as your minds and hearts. And thank you Vohel, for teaching what the most valuable things in life are, I hope you found what you were looking for.

A heartfelt **Thank you**, for everything so far.

This work is supported by: European Structural and Investment Funds in the FEDERcomponent, through the Operational Competitiveness and Internationalization

Programme (COMPETE 2020) [Project nº 039334; Funding Reference: POCI-01-0247-FEDER-039334].

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

RESUMO

Analytics tem desempenhado um papel fundamental ao longo da história, na compreensão de informação, simplificando dados complexos e tornando-os numa visualização mais digerível e apelativa. No contexto do ciclismo, *Analytics* obteve o papel pioneiro de gerar visualizações compreensíveis que adicionam valor, à atividade de ciclismo no contexto de *Smart Cities*, e à mobilidade inteligente. Apesar de existirem iniciativas de trazer a atividade de ciclismo para a vanguarda da mobilidade inteligente, esta continua a ser solução relegada, apesar dos inúmeros benefícios que pode trazer. Esta dissertação surge no contexto do projeto *Easy Ride*, uma parceria entre a Bosch Multimédia Portugal e a Universidade do Minho, com o objetivo de fornecer uma plataforma de apoio à exploração e tomada de decisões em torno da Mobilidade Ciclística com a esperança de destacar os benefícios e o valor que o ciclismo traz para a mobilidade sustentada. Para complementar os dados que são recolhidos e tratados pela plataforma *Easy Ride*, será criado um conjunto de visualizações e respetivas técnicas para proporcionar uma compreensão plena dos dados acima mencionados e promover o envolvimento com o projeto com o que este tem para oferecer.

Palavras-chave: *Analytics*; Mobilidade Ciclística; *Smart Cities*; Visualizações.

ABSTRACT

Analytics has played a big role throughout history, in the comprehension of information, by simplifying complex data and turning it into a more digestible and appealing picture. In the context of cycling, analytics serves in the pioneering of comprehensible visualizations that bring value to cycling in Smart Cities and smart mobility. Despite initiatives to bring cycling to the forefront of smart mobility it still tends to be a relegated solution, despite the numerous amounts of benefits it brings. This dissertation emerges in the context of the project Easy Ride, a partnership between Bosch Multimedia Portugal and the University of Minho, with the objective to provide a platform that provides support in the exploration and decision-making surrounding Cycling Mobility with the hopes of highlighting the benefits and value that cycling brings to sustained mobility. To complement the data that is collected and treated by the Easy Ride platform, a set of visualizations and respective techniques will be created to provide a fulfilling understanding of the aforementioned data and promote engagement with the project and what it has to offer.

Keywords: Cycling Analytics; Cycling Mobility; Smart Cities; Visualizations.

INDEX

FIGURE INDEX

TABLE INDEX

LIST OF ABBREVIATIONS

API	*Application Programing Interface*
KPI	*Key Performance Indicator*
IDE	*Integrated Development Environment*
PCT	*Propensity to Cycle Tool*
BFI	Bike Friendliness Index

1 INTRODUCTION

This Dissertation is the graduation project for the Integrated Master's degree in Engineering and Management of Information Systems lectured by the Department of Engineering in University of Minho. It was carried out with the help and supervision of Professor Rui João Peixoto José and its visualization contents were developed on the platform created by Bosch's Easy Ride project team.

1.1 SETTING

Analytics has played a big role throughout history, in the comprehension of information, by simplifying complex data and turning it into a more digestible and appealing picture. Because of the way that the human brain processes information, it is simpler to display charts and graphs to analyze certain data patterns, instead of forcing someone to go through each line of a spreadsheet or report, to understand the overall picture of any environment.

It is very common nowadays, for companies to invest thousands of euros into analytics tools and graphic reports. Because it is the fastest and easiest way of carrying out a message and comprehending an idea.

Understanding the importance of the visualization component, allows for otherwise intangible data, such as the coordinates registered by a sensor that is attached to a bike, to be more perceivable through a map that displays the route that said cyclist is riding. Even if an analytics project is doing an outstanding work in filtering out and correcting data, it still must face the challenge of its users understanding what said data means and what is its perceived value.

The Easy Ride project was created and is being developed alongside Bosch, a worldwide reference in electrical bike equipment that is stationed in Braga, as means to facilitate, and support new measures and upgrades to cycling and its sustainable mobility within cities and surroundings. It intends to achieve its goals of sustainable mobility by supplying law and decision makers with information systems and data, that help understand the current cycling environment and its needs.

This project will take place within a broad and multidisciplinary team that is developing new technological solutions for urban mobility. This team includes skilled members in the areas of Informatics, Electronics, Psychology, Communication, Service Platforms and Data Sciences. This setting, in a context of collaboration with the industry, allows us to emulate a work context with a high potential for real impact of the results produced.

1.2 MOTIVATION

When developing a platform which intends to supply information to a wide audience with different criteria such as the one being developed by the Easy Ride project, there needs to be a whole range of data being gathered and various use cases being created. So that it is possible provide satisfying answers to the questions being made by the users and allowing them to ultimately benefit from the project being developed.

However even with proper data management and organization, it remains a challenge for users to perceive any value from raw numbers and routes since it requires time and effort analyzing them. This problem exponentially increases when talking about intangible and incomprehensible data such as sensor readings, creating a need for an information broker between the data sources and the end user.

This is where cycling analytics and its visualization component come in, by creating and designing different views to accommodate as many of the users' needs as possible. This visualization component will be answering their questions by developing generic graphs and maps that can be queried, allowing for those users to have a more interactive and hands-on approach to data exploration and understanding.

The objective for these generic graphs and maps, will be to supply each category of stored data one query at a time, allowing for an easier consumption and understanding of the presented information without necessarily overloading the userbase with unneeded amounts of unnecessary information. Nevertheless, there is a user base who does not really know what questions to ask or what to necessarily look for. This a problem that can be solved through the creation of more generic visualizations. With the intent to generate more engagement from that user base, with project, so that they will eventually be interested in what the system has to offer and subsequently explore more data on their own.

Finally, it is important that the views that were developed, allow the users to make informed decisions when it comes to cycling mobility. In the hopes that the concept can be more widespread, and city planning can place more value into cyclable roads and the benefit they bring to the population that inhabit in said cities.

1.3 CHALLENGES

There are always challenges when it comes to properly developing the code for a project and successfully deploying it. From establishing the correct channels to import data to properly displaying said data it in a published and hosted website. However, the biggest challenge posing the project being developed is to understand what kinds of visualizations will the userbase truly need, and how can they be helpful in the process of decision making.

Despite being motivated to make a plethora of interesting and interactive displays so that users can participate in the process of data exploration. There needs to be more consideration for the abilities of the users that will utilize the visualization tools and their predisposition to properly interact with them. City officials who will be viewing the presented data will not want to see the exact same data as a cycling enthusiast that practices on the weekends, and these same enthusiasts may not be as familiar with visualization tools as someone who must use them for learning and scoping.

The challenge becomes recognizing the target audience and what they want to see when they open their visualization tool. Properly distinguishing each member of the userbase and adequately identifying what visualizations to provide them, will be the difference between a successful cycling analytics project and a failed one. The end goal is to have every single user achieve the goals that they set to themselves when first opening the visualization tools, without the same effort as they would have when going through their own personal exploration.

1.4 OBJECTIVES

To successfully complete the envisioned project and overcome its set of challenges, some objectives were laid down. They will mostly focus on understanding what the userbase wants and needs so that they can be addressed during development.

Objective 1: Identification of requirements with diverse stakeholders with an interest in cycling mobility data, for example cyclists, municipalities, entities promoting cycling mobility, public transport companies or shared bicycle operators and commerce.

Objective 2: Exploration of technological alternatives for dynamic generation of visualizations hosted on the web. Whilst considering that said alternatives must be open source, so that they can be built upon.

Objective 3: Develop the dynamic and interactive graphs and maps that represent the views of cycling mobility. These views will be made with the information gathered from the two previous objectives, which means they will have to accommodate the needs of the users and stakeholders whilst considering existing technological alternatives. Additionally, they will have to display and solely rely on the data available on the Easy Ride platform, since it will be the most representative of the cycling mobility panorama of Braga.

Objective 4: The developed views will need to be made publicly available and their overall impact will have to be evaluated alongside the stakeholders which were surveyed during the first objective. Once the results of the evaluation are tallied up a report will be written so that a conclusion can be reached about what was achieved.

1.5 DOCUMENT STRUCTURE

The dissertation is structured and divided into eight different sections, that aim to portray and explain the work done throughout it.

Starting with the Introduction where the scope of the project is set, alongside the expectations for it and the expected challenges. Followed by the State-of-the-Art section, that paints a picture of the work done on the field of cycling analytics and related subjects.

The Methodology section presents the research done into selecting a methodology to guide the dissertation and how the selected one was followed and utilized throughout the dissertation.

The Visualizations System Prototype section explains the architecture that shaped the visualizations system and provides a brief explanation of each component of the system and how they operate. It is then followed by The Visualizations section, that explains how the visualizations are operating, from the libraries and languages used, to how they integrate the

web portal. The next section is Reasoning and Lessons from development, which provides context into the decisions made during the shortcomings of development and the lessons that were inferred from those difficulties.

The dissertation ends with two sections, the first being the Evaluation of the Results, in which the objectives that were previously established in the Introduction section are reviewed. Followed by a general evaluation of the project where the results of the project are discussed and how the outcome satisfied the expectations that were set in the beginning. The final section is the Conclusion, that provides a small briefing of the work done in the dissertation, discusses the outcomes and provides insight into the future work.

2 STATE-OF-THE-ART

The State-of-the-Art section presents work on cycling analytics and related subjects. Since this area inherits various other areas of work, each will be explored to paint the bigger picture of what exists and could be used, in the completion of the project. Furthermore, since the visual tools will need to utilize open-source technology, it was decided that Python and its libraries would be the best approach for developing the needed graphs and maps.

2.1 ANALYTICS

Analytics is a study field pertaining to the systematic computational analysis of data or statistics. Which is the focus of this project. Before understanding the use of maps or mapping tools, first there needs to be a clear understanding of what cycling analytics is, by first understanding the term analytics.

According to Cooper, defining the term analytics is a difficult task because of its broad usage in different fields of work. However Cooper still attempts to sum its meaning to; The process of defining the problem and scope of a project and subsequently applying to it statistical models and comparing them to existing or simulated future data (Cooper, 2012).

The definition that Cooper attributes to analytics is solid and applicable in most case scenarios, however as Barneveld mentions, analytics as a term isn't a one size fits all (Barneveld et al., 2012).

Therefore, it is important to understand what area of analytics is being utilized, so that the process of developing a project around it becomes easier and more streamlined. According to Watson, analytics can be used in three major areas, those being; For optimization in the case of mathematical programing, predictive analysis in the case of machine learning models such as decision trees, and descriptive analysis, which pertains to data storage and visualization tools (Watson, 2011). With a term so broad and applicable to different areas, analytics in the context of this project can be classified as descriptive since the focus will be on visualization tools.

As Richard Katz states in the book “The Tower and the Cloud”, analytics can provide a methodology that assists in the exploration of available data (Computing, 2013), which in the context of the project would be the exploration of cycling data and overall bicycle usage.

As such, analytics, in the context of the project, can be described as the; Process of finding and defining the scope of the project and applying statistical models. With the objective of making a descriptive analysis of the available data through visualization tools, while remaining within the scope of cycling and micro mobility.

2.2 VISUALIZATIONS

Visualizations, within the scope of cycling analytics, previously mentioned in the Analytics section, can be attributed to the end goal of producing a descriptive analysis of the studied environment, through visualization tools.

Therefore, to create a descriptive analysis, firstly there needs to exist a clear understanding of what is needed when visualizing data. And subsequently the message that needs to be conveyed, through the created visualization tools.

Starting with the term, Data visualization, it can be described as a quick and easy way to convey information. It can be used in various ways such as; In storytelling, the “connection of dots”, identification of trends, finding outliers, generating audience interest and in the summarization of data in the form of comprehensible representations(Conner et al., 2020).

For the creation of the visualization tools, Golfarelli et al. specifies that the process of building a visualization project must follow a set number of steps. Beginning by declaring the visualization context, then choosing a visualization type amongst a theme, and then binding them into one visualization window.

There are also some requirements according to Furht & Villanustre ,that must be adhered to have a good visualization project. Those being that it must be expressive by demonstrating exactly the information contained in the data, effective by not straining the human capabilities of visualization, and appropriate through the balance of the cost-value ratio of the visualization by having the maximum benefit and the lowest cost possible.

Another aspect of visualizations is its ability to convey a message to the user and for it to learn from the information that it observed. According to Hsiao et al. some visualization

techniques that can be used for that extent are; The nearest neighbor technique, which associates unrelated information belonging to a specific region, the focus technique, that targets a specific piece of information, and the clustering technique, that presents information that is related in some degree.

In the same subject of teaching, the CNN Explainer (Wang et al., 2020), focuses on global visualizations techniques such as global color scales to differentiate good from bad results, text annotations to guide users through more complicated subjects, a control panel to customize visualizations and animations to link related information.

Asides from techniques that facilitate user perception of the data, there needs to be the appropriate graphs and maps to visualize said data in the easiest way possible. The FT Visual Vocabulary project (*Chart-Doctor/Visual-Vocabulary at Master · Ft-Interactive/Chart-Doctor · GitHub*, n.d.) focuses on improving chart literacy by pointing the biggest strengths each chart has to offer.

When it comes to map visualizations, Behnisch et al. emphasizes that topological representation with the minimum essential metric information is the best course of action for designing map views, and that the best special representation of traffic prediction or human activities come from utilizing existing roads and drawing in axial lines on top.

Researching the techniques behind what compose well-made and quality visualizations, provided insight on useful ways to create quality visualizations, that don't depend on the limitations that the creation tools may have.

The main techniques in creating good visualizations are amongst many; Finding relations in cycling data that can paint the overall picture of a city, sticking to one consistent visualization type like maps, that are easy to interpret and understand. While focusing on a region of data and highlighting the key components that want to be seen.

2.3 SMART CYCLING

The Smart cycling field of study is very large when it comes to the number of solutions and technological innovations that it covers. Meaning that is not exclusive to gadgets or technological advancements made on bikes. Therefore, tremendous value can be found for designing visualizations because they also play a major role for the future of smart cycling.

Understanding that role is paramount in the design of visualizations so that they can accomplish their goal in the advancement of smart cycling.

Smart Cycling innovations have led to the transformation of cycling practices and the revision of the meaning of cycling (te Brömmelstroet et al., 2020). However these innovations have yet to be systematically reviewed and appreciated which in turn means that their potential role in mobility systems has not been achieved (Nikolaeva et al., 2019).

Despite the current condition of smart cycling, Nikolaeva also states that, it is important to set eyes into the future, because smart cities will be driven by technology which if implemented properly has the power to introduce behavioral change.

Both authors mention how politicized smart cycling is and emphasize the need to convince policy makers to adhere to initiatives. Behrendt mentions how the European Commission acknowledges the health benefits, however it still is the most marginalized mode of transports in Smart Cities (Behrendt, 2016).

In technical terms Smart Cycling derives from the term “Smart Mobility”, and according to Docherty, Smart Mobility is the encapsulation of various phenomenon such as the shift from ownership of vehicles to “usership” which means their servicing. Other phenomenon consist of the market redefinition of the word mobility, going in tandem with the servicing of vehicles and the rise of sharing companies, and the third and fourth consist on greater convenience for the end user, shifting its role in the transportation system (Docherty et al., 2018).

With the increase of servicing businesses in the transportation sector the next step towards the envisioned state of smart mobility would be the development of a general network system instead of a series of separated vehicle services (Elliott & Urry, 2010).

Therefore, Smart Cycling can be summed up to a series of transformations in cycling mobility and practices. That intend to introduce a behavioral change in both population and governance, so that a healthier infrastructure of movement and traveling can take place in today’s cities.

Meaning that the role of Cycling Analytics in the scope of Smart cycling, is that of a support to the ideas and initiatives to change and upgrade smart mobility. Focusing on

providing visual aid for raising awareness and conveying information that result from these initiatives for change.

2.4 CYCLING MAPS

Cycling maps represent the spectrum of possible map visualization techniques that can be applied to a map visualization layer. When studying cycling maps the first step is recognizing the available map types and how they are used.

There are some widely used map types for plotting traveled routes, first one being street-maps, which covers an area around a city and is used to detail route shapes. On the forefront of street maps there's OpenStreetMap, a crowdsourced project which aims to have a fully detailed map that's free of use and license (Weber, 2008).

Other map type that is quite relevant, is the heatmap, which utilizes a map layout like street maps but instead of plotting routes, it plots areas according to their relevance (Zhao et al., 2021). According to Zhao, these maps perform a better job of displaying points of interest when comparing to drawn routes, as can be seen in Figure 1(Zhao et al., 2021):

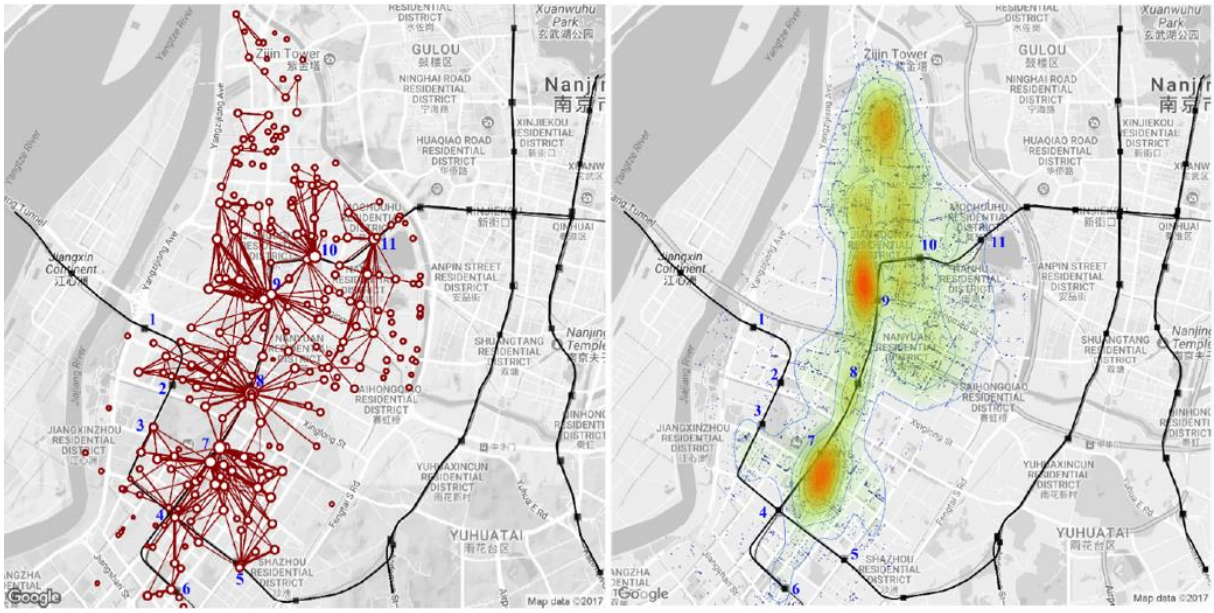


Figure 1 Comparison of both map types

Heatmaps prove to be less cluttered and provide better insight into more active areas of cities. However, there is a third type of map visualizations that merge the point of interest highlighting potential of heatmaps while retaining the drawn routes of street maps. These are called Tube maps and are typically used to portray underground railroads. These

maps utilize, symbols as cognitive aides for recall, clear title legends for an overall understanding, a visible and a scaled time axis to display their routes. These are generally regarded as helpful for target reaching because of the amount of detail fit into one image and structured information which can be zoomed in. (Aslak et al., 2005; Burkhard & Meier, 2005) as shown in Figure 2(Aslak et al., 2005)

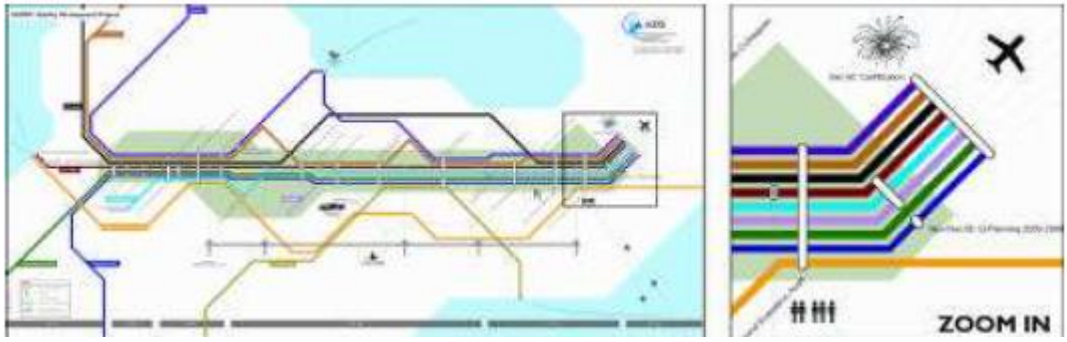


Figure 2 Depiction of a Tube map

Regarding Map views that present successful innovations, Lee et al. presents a map configuration that is akin to street maps with the twist that those routes are color coded by their density creating a heatmap hybrid. S. Chen et al. presents a map layer that is more leaning towards a heatmap, however it represents unstructured pairings of origin-destination routes for an easier perception of movement tendencies and points of interest.

All in all, map types are an essential component to consider in the spectrum of visualization tools. They serve as the main attraction for the appeal of the visualizations and their choice affects many of the following decisions that involve customizing the map.

Despite the existence of many different types of maps, not all can be represented through the most used tools. Whilst some are very valuable and visually appealing, their implementation ability will largely depend on the technologies available for use.

2.5 CYCLING ANALYTICS

Cycling analytics is the culmination of all the previously mentioned fields of study, where the representation techniques meet the statistical and conceptual sides of cycling mobility. The development of the visualization tools will depend on what has been previously done on this field, and what it currently lacks, to progress it, into hopefully, a better state.

When approaching the concept of Cycling Analytics, one would assume that it would be about applying the concept of analytics to an environment where the studied data would be related to cycling. However as previously stated in the field of Analytics, because of the many interpretations the word can have, subsequently, the field of cycling analytics also covers a wide share of conceptually different projects and studies.

Beecham et al. sums up his visual analytics prototype for bike sharing, with a very accurate portrayal of what cycling analytics is. That being that it should enable space-time patterns of cycling behavior to be associated with the customers' usage characteristics.

With the definition of Beecham in mind, there is several works being developed that match the description of Cycling Analytics. The first one being BOOST, a tool that determines the gross potential for cycling in a region or city in Portugal. By comparing the existing population per square kilometer, to the available activities for the population in the area such as jobs, or other day to day activities that demand commute.

The ratio that is obtained from the comparison is displayed on a ranking table and an in-depth heatmap, that covers each area of the city. The ratio demonstrates the potential for a said region or city to create a sustainable cycling infrastructure. The project intends to make itself available through the Web so that decision makers can evaluate their position and consider the city's overall potential for cycling mobility (Potential, 2020).

Another innovative cycling analytics project is the Propensity to Cycle Tool (PCT), which is an open-source, R based tool, that utilizes OpenStreetMap for displaying maps in the United Kingdom and Whales.

It was made to provide insight into the network of routes that compose each city, by providing data related to them, mainly focusing on its cycling potential, which is calculated using the trip distance of the route, and the overall number of hills and inclinations that make it up. (Lovelace et al., 2017).

The Bike Friendliness Index (BFI) proposes a heatmap developed on Mapbox and OpenStreetMap, that displays the biking friendliness of cities. Taking into consideration the existing infrastructure that the cities provide to cycling and urban mobility.

It focuses on five different metrics that are weighted unequally, those being the average slope of the roads, the density of buildings within the city, the infrastructure, current political

compromise, and average bike usage, to create the afore mentioned index (Figueiredo, A.P. e Vale, 2018).

Another biking index that presents a different set of metrics, was proposed by Arellana, suggests utilizing both Observable and non-Observable characteristics of road infrastructure for the classification of the bike friendliness of cities.

The criteria are the presence of bicycle infrastructure, the climate of the city, the cost of trips made by motor vehicles, the comfort and attractiveness of the roads and the overall traffic safety of the city (Arellana et al., 2020).

In the context of applications, CityVerse is a dashboard that tries to integrate different aspects of urban sustainability including bicycle journey data. It utilizes a web-based Django framework and focuses on several different visualization techniques to present its cycling data, from a time series analysis, a correlation analysis with other urban data and hierarchical clustering for route bundling so that data can be presented in a cleaner way (Gledson et al., 2019).

All these projects constitute major strides made in the field of cycling analytics and provide useful tools to integrate in more visually focused projects such as these. The utilization of more complex indexes can provide the visualizations additional ways to display more intricate data, that is usually very hard to present.

The inclusion of indexes such BFI and PCT, would help to convey a more detailed message of change and awareness to the officials in charge that can instigate change in the infrastructure and consequential cyclability of cities.

2.6 PYTHON AND ITS LIBRARIES

The chosen programing language for developing the visualization tools was Python, and therefore all the research put into finding suitable libraries, was done within the scope of Python. Whilst taking into consideration how they could complete the established requirements for the project.

The Table 1 provides an apt view of many Python libraries that are widely acknowledged. It will feature their names alongside their latest supported version of python, a brief description of their capabilities and what field do they operate in.

Name	*Description*	*Field*
Numpy *Python >=3.10*	Numpy is a widely known library for python, that provides support for large, multi-dimensional arrays and matrices as well as other more complex and high-level mathematical functions.	Analytics
Plotly *Python >=3.9*	Plotly is a graphical library that focuses on generating interactive and visually appealing graphics. It is mostly used alongside other analysis tools, to create custom made dashboards, strengthening visual aspect of the dashboard.	Analytics
Geoplotlib *Pyglet >=1.2.4*	Geoplotlib is a python API that generates geographical visualizations though OpenStreetMap. Given proper data such as coordinates, the API automatically creates a dot map with each dot corresponding to the coordinate it represents.	Visualization
Geopandas *Python >=3.7*	Geopandas is the Pandas API for plotting maps. It is considerably more complex and harder to use then other plotting libraries, however it is also much more forgiving on the data that it receives and has much wider applications.	Visualization
Folium *Python >=3.5*	Folium is a library used for visualizing geospatial data. It is also a wrapper for Leaflet.js which is an open-source JavaScript library for plotting interactive maps. Its biggest asset is the ability to personalize and customize its maps, which are very detailed and interactable.	Visualization
Bokeh *Python >=3.7*	Bokeh is a library that focuses on interactive data display and visualization. What differentiates bokeh from other visualization libraries, is the fact that it uses HTML and java script to render its graphics, allowing the creation and development of web-based dashboards.	Visualization
Seaborn *Python >=3.6*	Seaborn is not so much a library by itself as it is an upgrade to matplotlib, it is an API that aesthetically improves matplotlib allowing for more aesthetically pleasing graphs and charts.	Visualization

IPyleaflet *Python >=3.5*	IPyleaflet is another Leaflet.js based library for creating interactive maps. The biggest advantage is that its fully supported by IPywidgets, enabling more interactive maps.	Visualization
IPywidgets *Python >=3.5*	IPywidgets offers an interactive set of HTML widgets for the Python kernel. These can be linked to actions, enabling the creation of fully interactable applications.	Analytics
Streamlit *Python >=3.6*	Streamlit is an engine for creating data apps, available as a library in python, with fully disclosed source code. Allowing for the use of widgets, graph scripting and web deployment.	Deployment
Voilà *Python >=3.6*	Voilà is an add-on that converts Jupyter notebooks, into standalone applications with a fully dedicated kernel, to allow for intractability with the applications.	Deployment

Table 1 Python Libraries Comparison

3 METHODOLOGY

The Methodology section of the dissertation provides context into the research made into finding a suitable methodology that would help in the completion of the project. It also follows the chosen methodology, completing its steps sensibly, without compromising its development.

A methodology structures the workflow required for the completion of the dissertation, which is critical in having an organized and well thought out project. There are many established and useful frameworks to conduct a visualization project, and the objective is to have an overall flow that the project can follow. Having guidelines to work through these essential steps goes a long way in developing a successful project with minimal step backs.

While it is possible to complete a development project without adhering to a specific methodology, due to the practical a straightforward nature of developing, using an established methodology gives credence to the project and helps to determine an overall thought process and approach for development.

Finding a methodology that fits is imperative, however it shouldn't be followed religiously if it brings consequences in the development of the project. Using an inadequate methodology for the sake of using, can cause more harm than good, and may lead to it being ignored all together during development.

3.1 METHODOLOGY RESEARCH AND CONTEXT

The selection of the research methodology for this work involved the analysis of various alternatives with the intent of finding a model that is adequate for the development of the project. The search was mainly done on Software development life cycle (SDLC) methodologies since they are the most adequate for this project.

One of the oldest and most established methodologies is the Waterfall Model(Petersen et al., 2009). It is composed by a series of sequential steps that need to be followed in an orderly fashion beginning with the requirement analysis phase, and then going through high level design, coding, testing and finally ending on maintenance.

What makes the Waterfall Model such a widespread model is the fact that, by starting with analysis and design step, most design flaws end up being found and discussed before development. However, it doesn't mean that it is foolproof, since most technical errors can only be found during development, despite that, it still tends to save a substantial amount of time by not going back to previous phases or redesigning during development.

The waterfall model isn't impervious to issues and as such faces some design flaws that contradict the nature of more experimental projects such as this one. The major one being its high inflexibility, which derives from the mandatory understanding of all the requirements in the beginning of the project and the high amount of documentation that needs to be done before any practical development, heavily restricting on the fly solutions and alternate routes during development.

As a result of the model's inflexibility and closed workflow, the Client has a very poor understanding of the planned solution. The issue results from no intermediate development steps before the final solution. Causing the Client to be unable to preview the solution before its completed, possibly resulting in unmet needs and a resulting unsatisfaction with the presented outcome.

A model that tries to fix the biggest flaw in the waterfall model is the Spiral Model(Boehm, 1988). It operates in bursts of development cycles and goes through them as much as needed, whilst always weighing in the risk it takes to complete the project.

The combination of bottom up and top-down approaches of the Spiral Model, results in a more flexible design phase that can tackle and work around the technical issues that occur during the prototyping phase. Without sacrificing the proper planning that came with the Waterfall model. As a result, this approach manages to minimize risk during development, by having technical experimentation and subsequently presenting it to clients, which helps in receiving feedback and reviewing requirements.

The Spiral Model operates through four core phases, those being: Planning, which heavily involves the client so that the requirements can be thoroughly established, Risk Analysis, which englobes the prototyping phase and serves to weigh in the risks that the project has, and the possible workarounds. Development, where the software is created and tested, to provide the potential final product. Finally, the Evaluation phase gives the Client the

opportunity to evaluate the delivery, linking it to the Planning phase, so that new cycles can be created and form the spiral which will tie in with future iterations.

Even though the Spiral model prides itself in safety due to the high amount of risk analysis, it comes at the cost of massive investments in both time and cost, to properly execute it. From needing experts in the field of risk assessment, to the high amounts of documentation required for even smaller projects.

In the context of the current project, it would be suboptimal to adapt the Spiral Model, since more time and resources would be spent planning instead of actual development, leading to an unnecessary slowdown of the project.

The last model to be studied before settling on a choice was the Iterative and Incremental Model(Larman & Basili, 2003). It takes the processes of the Waterfall model and completes them in an iterative fashion. The objective is to produce deliverables with each iteration, to display progress to clients and receive adequate feedback before the final product is presented, which was one of the major weaknesses for the Waterfall model.

With this model the priority is to construct a partial implementation of the system with the basic requirements, and incrementally add the remaining parts until it becomes functionable and according to the specifications planned.

The Iterative Model strongpoint is the ability to develop high risk functions first. To assess their implementation ability, which diminishes the risks taken when developing in one large cycle. Resulting in constant customer feedback through the presentation of deliverables, in-between cycles, which results in higher customer satisfaction and product improvements as development progresses.

However, this is not a perfect model since its execution requires a very good understanding of planning and design, so that development produces early versions of the system. That are fully functional and able to be expanded on throughout the project. At first the Iterative Model appears to be a very flexible model. However, the reality is that it does not allow for backtracking during the project, due to the nature of its incremental expansibility, making it in turn, a very constrictive model in nature.

Whilst reviewing these models it became apparent that they all provide some major advantages in small project development. However, they are held back by some major

constraints that make them very hard to work within this context. The most notable ones are the large amounts of planning and design beforehand, and the documentation that needs to be produced. Both are very hard to properly accomplish due to the experimental nature of a thesis project. Which has large amounts of solution experimentation and product redesign throughout it, before coming up with a desirable solution.

3.2 THE METHODOLOGY OF CHOICE: THE A-Z MODEL

The AZ-Model(Akbar et al., 2017) aims to merge the benefits of the process focused models such as the Waterfall model, with the easily adoptable and change accepting methodologies such as Scrum. It's a model that operates in three different stages, each with additional subphases, those being the Costumer Involvement phase, the Development Phase, and the Product Release Phase. Following each of them in order, so that communication and requirements are gathered before development, ensuring that in the end the product is released according to customer specifications. The Figure 3 is a representation of all the subphases that compose the model.

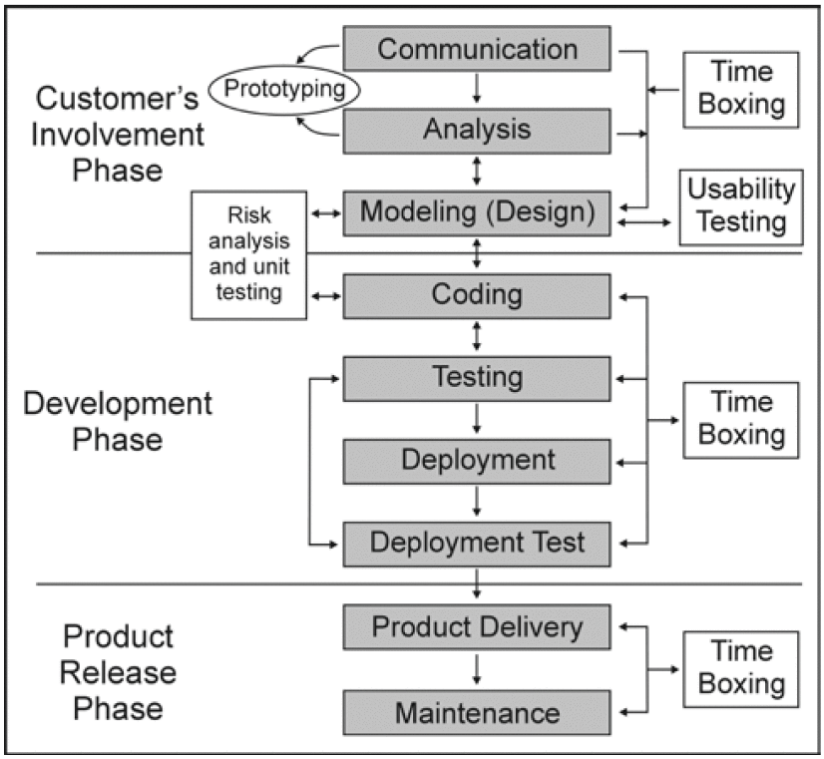


Figure 3 The A-Z Model

The first phase is the Customer Involvement Phase, in which the client is the centerpiece. This phase is further divided into three sub-phases. They mainly focus on

information gathering, mainly focusing on the client's needs, objectives, and requirements for the project.

The first sub-phase is Communication, in which contact is established with the client. During this phase is where all the important information is gathered, as previously mentioned these are the requirements, needs, objectives and the expected final product.

It is then followed up by the Analysis phase, where the requirements are curated and analyzed, this phase works in tandem with the communication phase since during the talks. Initial solutions can be prototyped and discussed to further help in the upcoming Modeling/Design phase.

The Modeling phase follows up Analysis and iterates upon what is produced during it. It's crucial to develop a solution that pleases the clients. According to the methodology, the way to produce an adequate design for the clients, is through constant risk analysis and testing, of initial coding prototypes. By testing out many possible solutions, and presenting them for feedback, it's easier to produce a robust plan that won't suffer changes later in development.

A key phase during the Costumer Involvement phase, is Usability testing. As the name implies, testing which systems will feel the best for the end user is paramount in the satisfaction of the client. Many times, the most developed and robust solutions end up being unwanted by the client. Therefore, to avoid major commitments in the development of wasted infrastructures and code, creating initial interactable protypes for the clients, before the development cycle, can avoid the issue of developing undesirable solutions.

The Second Phase is the Development phase, as the name implies, during it, the product will be developed and can only be deemed finished when something presentable can be tested and ready to be deployed.

The Development phase divides itself into four sub-phases that compose the basic steps towards a successful product development. It starts with Coding; This sub-phase is still tied with the previous modeling phase. Since to produce prototypes, one naturally needs to code them. However, once a design is settled upon, the phase continues, as it still needs to complete and perfect the chosen prototype.

Next subphase is Testing, it's closely related to coding as both are an iterative process, since once new developments are made, they need to be tested. Long periods of coding often result in more difficult, longer, and unnecessary periods of debugging. However, testing isn't only limited to the functionality of the program, testing the layouts, usability and navigation of the product is also essential. Some flaws may pass unnoticed by debugging and only through using the product extensively can they be exposed.

The next two phases are Deployment and Deployment Testing. Deployment is an essential part of development, and if it's not thought of during design and coding, it can result in situations where a successfully coded program or application has no compatibility with deployment platforms, and consequently becomes unusable. Therefore, some deployment testing needs to be done during design and conceptualization to avoid this pitfall.

The final major phase is the Product Release Phase, composed of only two sub phases. The first one is Product delivery, as the name implies, the focus of the phase is to release the product to the client for utilization. It's a formal step since the product was completed during the deployment phase, and so the only thing left to do is to properly deliver it. By this time no major changes are made, since the project was closely aligned to the client needs that were defined back in the Modeling phase.

The final sub-phase is Maintenance, set up as the last phase of the project due to its nature of continuously being able to provide support to the delivered product. It can be viewed as the longest phase since its end is only scheduled to when the product is no longer deemed viable to be used. From *bugfixing* to troubleshooting and even usage guidance, any task that pertains to the lifecycle of the product and the client satisfaction its Maintenance's responsibility.

3.3 UTILIZING THE A-Z MODEL

After choosing the methodology that best suits the project, the next step is to then determine how each phase will be utilized in the completion of the project. Not all phases need to be strictly followed as the model is simply a guidance tool in helping the completion of the project.

Following the methodology of choice, the A-Z Model, the first step in development was to communicate with the client, which in this case is Bosch, and access the needs and expectations for the visualizations.

3.3.1 CUSTOMER INVOLVEMENT PHASE

When talking to the Client, the first thing to be established is that the visualizations will compose the frontend of the platform, as well as final process in the display of the data that the EasyRide Platform is gathering.

3.3.1.1 COMMUNICATION AND ANALYSIS

The first thing to determine during communications with the stakeholders, were the requirements for the project. In the topic of available data, it was discussed that the data for the project was from OSM (Open Street Maps) in the format of Geojson. The data was mainly pertaining to route data, where each route contained aggregated information of the bike count that went through each route alongside the average speed at which people cycled in the route, and the slope of the route.

The questions posed were mainly directed to the Client and pertained to what the end user wanted to see when observing the maps containing the available data. Having data and knowing what it means is important, but without an end goal and an overarching idea of what is interesting and needs to be seen, the information contained in the routes has no use.

The answers obtained were sufficient to answer the question of what the target audience was. The main objective was to display the data to decision makers and people interested in change. Therefore, the visualizations had to focus on the data that was collected by individuals, which was then aggregated into comprehensible datasets that pertained to each region in Portugal. And finally, those datasets had to be displayed in a map that could represent each route and allow for further exploration.

3.3.1.2 DESIGN

Figma was used to design sketches of the final product. Figma is a web based, vector graphics editor with plenty of add-ons and community tools for designing prototypes. Using Figma, several maps were sketched based on the examples found on python libraries that were previously researched for the development phase, presented in Table 1.

Both the initial page and subsequent visualization pages were designed, however they were not meant to be strictly followed, as some design liberty needs to be given for the development phase. The main point was to sketch out and highlight the most important design philosophies.

The designs were based on two main criteria: The first is how will the data be grouped and shown on each visualization; And the second is how will the visualizations be useful for the end user. Both criteria are heavily intertwined, as having a comprehensible and useful visualization is closely associated with how the data is chosen to be presented. The first step in designing these visualizations was to approach the conveying of the information. Through coloring, highlighting and noise reduction, the user would be more engaged with exploring snippets of information, instead of looking at a cluster of data.

Coloring allows for immediate recognition of the target data amidst each route, without needing to click on each individual one to search what It's looking for. Highlighting works in tandem with noise reduction to remove visual clutter from the screen. By using less detailed backgrounds and layers, alongside providing layer filtering, a visual environment is created where the user will want to explore instead of being intimidated by incomprehensible and hard to grasp visualizations.

Once the visualization techniques were established the next step was identifying what kind of data was going to be shown. Dumping all data into the visualization isn't a good approach and interferes in the clarity trying to be achieved. Therefore, the visualizations had focus on specific regions, which in the case of Portugal they pertain to its Districts. Region based data helps the user situate itself and improves performance on interactable visualizations, effectively solving two problems with one solution.

The designed sketches of the visualizations were created based on the selected library for developing the maps and future views, *IPyleaflet*. They were modeled based on tech demos that already existed and the appropriate widgets were added to complement the maps. The widgets were designed based on the *IPywidgets* library, which has full compatibility with the previously mentioned map library. This enables fully interactable applications that give more freedom to the user if it so chooses.

The Figure 4 represents the initially prototyped map view; It lacked any coloring as that would be an inefficient use of time for what was essentially an initial demo of what the visualizations would become the widgets on top of the map represent the selectable data that can be filtered.

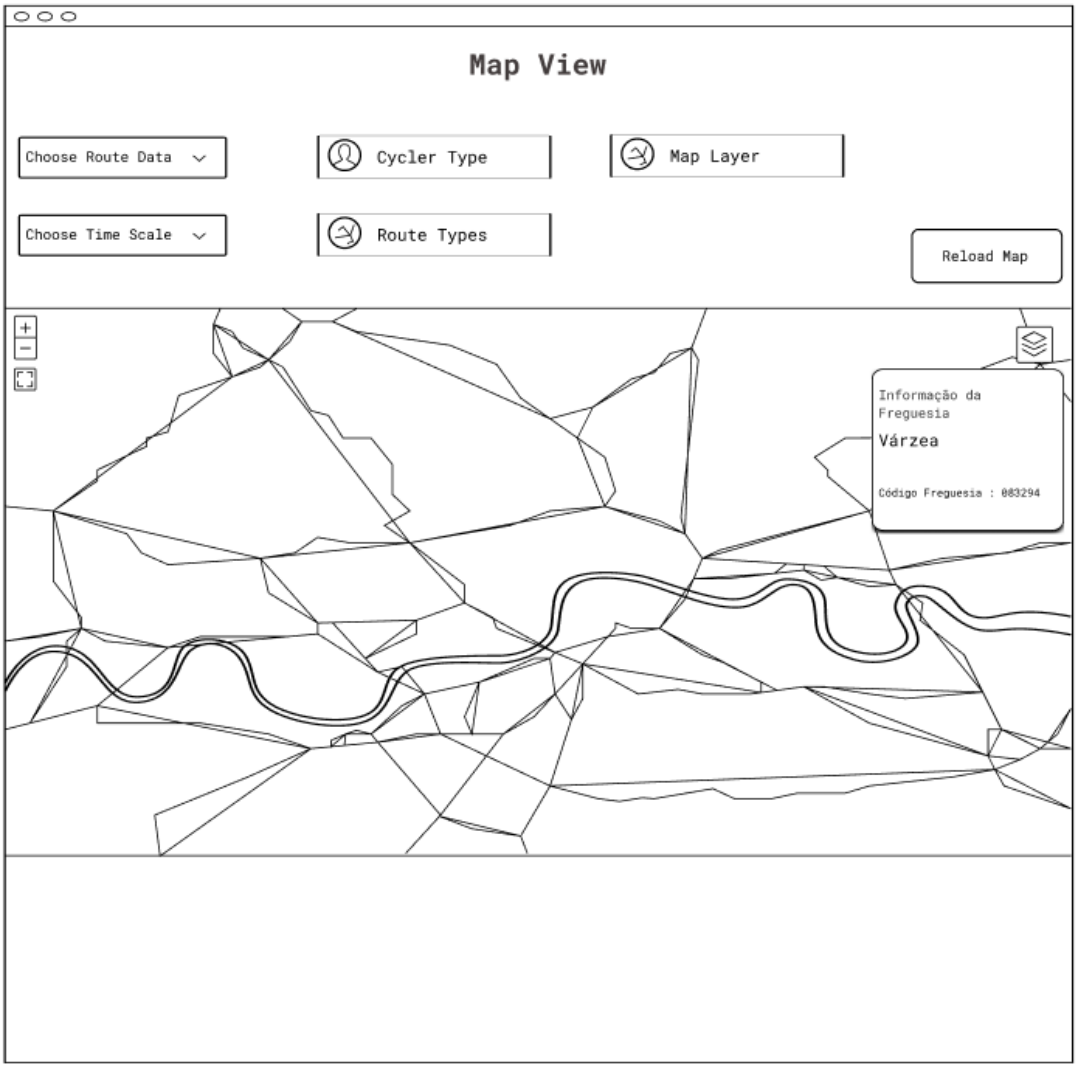


Figure 4 Initial design of the Map visualizations

Alongside the map view, two additional prototypes were created. The first one pertained to the welcome page of the platform, and the second one was a graph view, with complementary data, that is meant to provide statistical backing to the observed visualizations.

The front page, shown in Figure 5, was designed to be enticing to the public, with the objective of displaying general information. Hopefully through it, users would be engaged to try other views, and consequently adhere to the platform. The information that could be of interest, would be related to the usage of the platform, and the bike activity in the city. By

displaying bike activity in a statistical fashion, the user would feel more compelled to explore bike related information through the map views. Those would then go into detail on the bike usage throughout the district and offer more exploratory options.

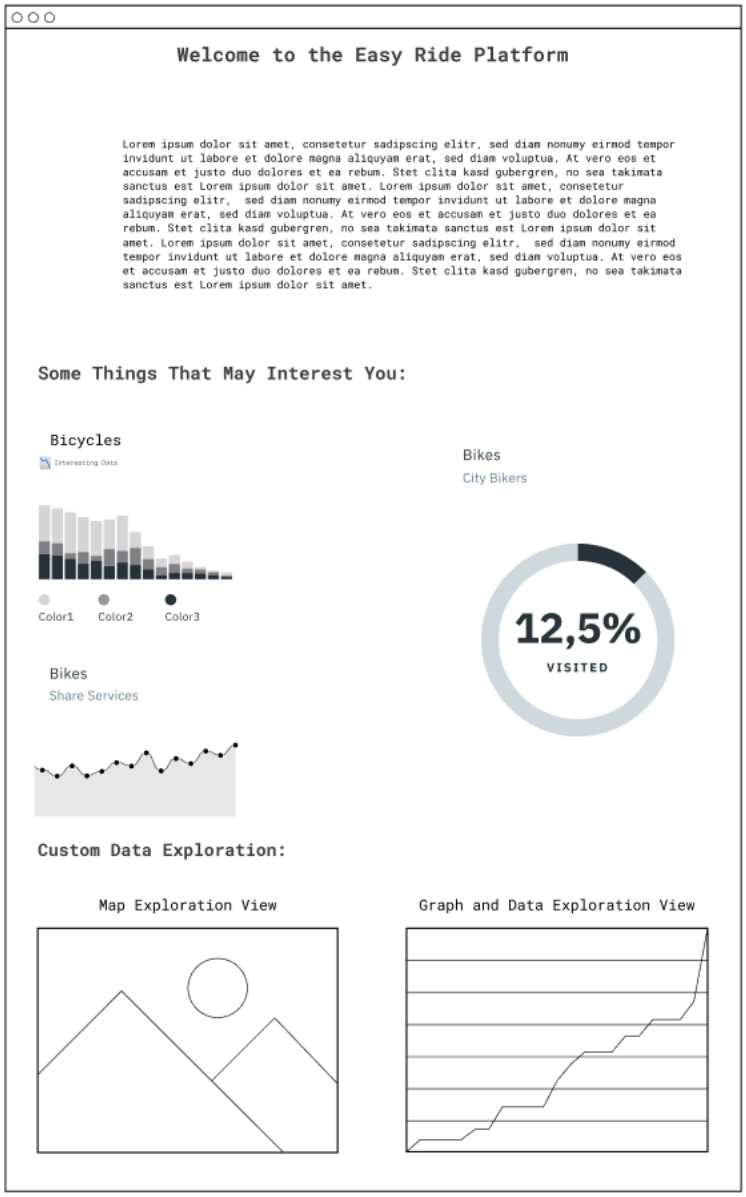


Figure 5 The Web Portal welcome page

Much like the front page, the graphical views were designed to be focused on displaying bike activity and grouped statistical data. The intent was as said previously, to provide data that could serve as backup in what was being shown by the map views.

Alongside being complementary data, the graphical views, shown in Figure 6, would also provide other relevant types of data that were supplied by the platform. That data was deemed relevant, however was not adequate for display in the map views and therefore would be relegated to this page. It would still be pertaining to the types of data presented in the maps such as routes and zones and was presented in graphical fashion.

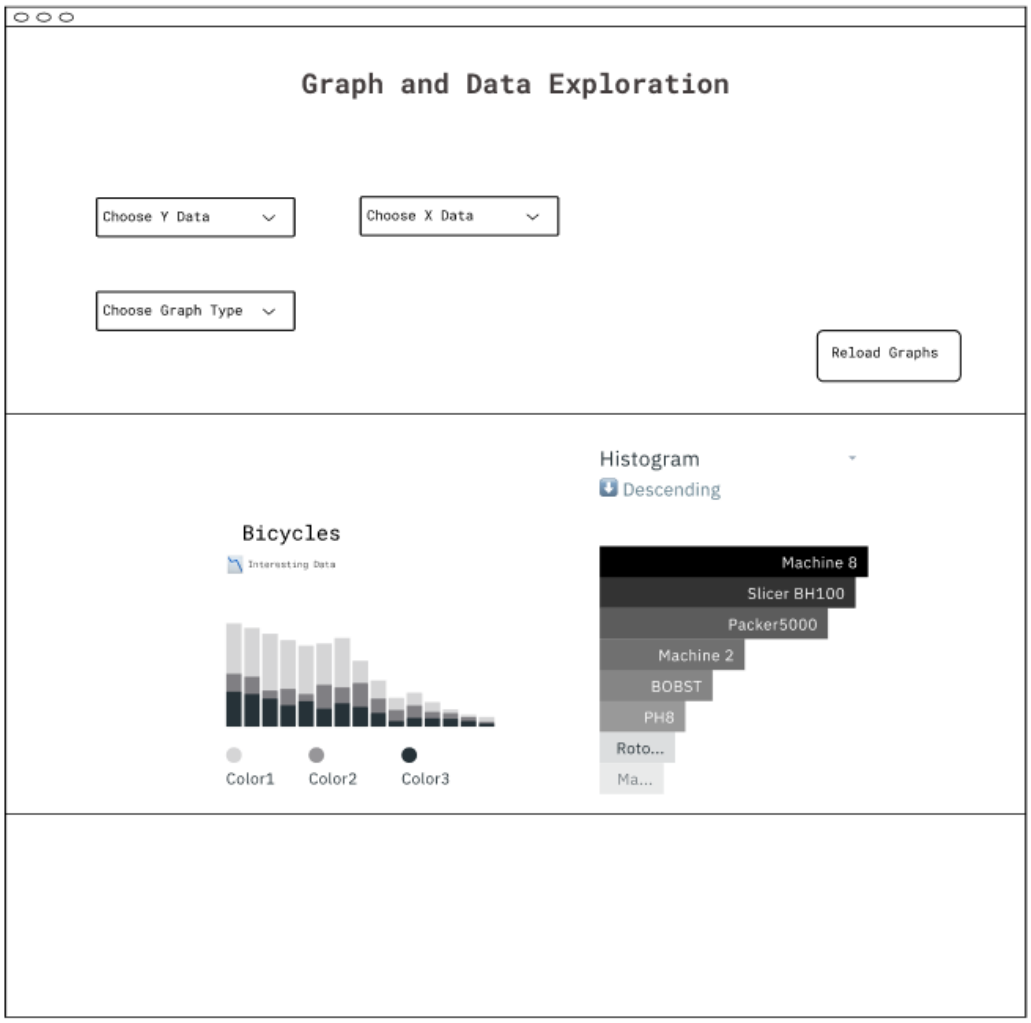


Figure 6 Graphical Views

3.3.2 Development Phase

The development phase was very long, comparatively to the other phases of the project, and detailing the whole process would be out of the scope of the dissertation. However, details relating to the functioning of the visualizations system, and the lessons learned in developing it, can be found in The Visualizations System Prototype and The Visualizations sections of the dissertation. As a footnote, the testing phases of both

development and deployment were done in tandem with them, being repeated each time new features were added to the visualizations.

3.3.3 PRODUCT RELEASE PHASE

The Release of the product is the final phase in the conclusion of the project and marks its transition to the maintenance phase. Because the project is a part of Bosch's Easy Ride set of projects, it was set to end with a presentation open to other Bosch contributors. Therefore, the release of the visualizations was firstly seen by a public that wasn't the initially targeted one, those being the decision makers and other responsible for the changes in infrastructure. Nevertheless, it helped in gathering initial feedback.

The Maintenance of the project is an uncertain reality since this dissertation won't focus on its continuous support. However, measures have been made in the development of the code, so that future contributors that may be interested in contributing to the development of the maps can do so without beginning from scratch.

4 THE VISUALIZATIONS SYSTEM PROTOTYPE

The visualization system prototype section explains how the visualizations system functions, by first explaining the system architecture and then delving into the visualizations, and how they operate.

4.1 SYSTEM ARCHITECTURE

The system architecture of the visualizations prototype revolved around four major components, those being the analytics server, the development environment, the server, and the web portal. Each played a significant role in their fruition, and they composed the system architecture as can be seen in Figure 7.

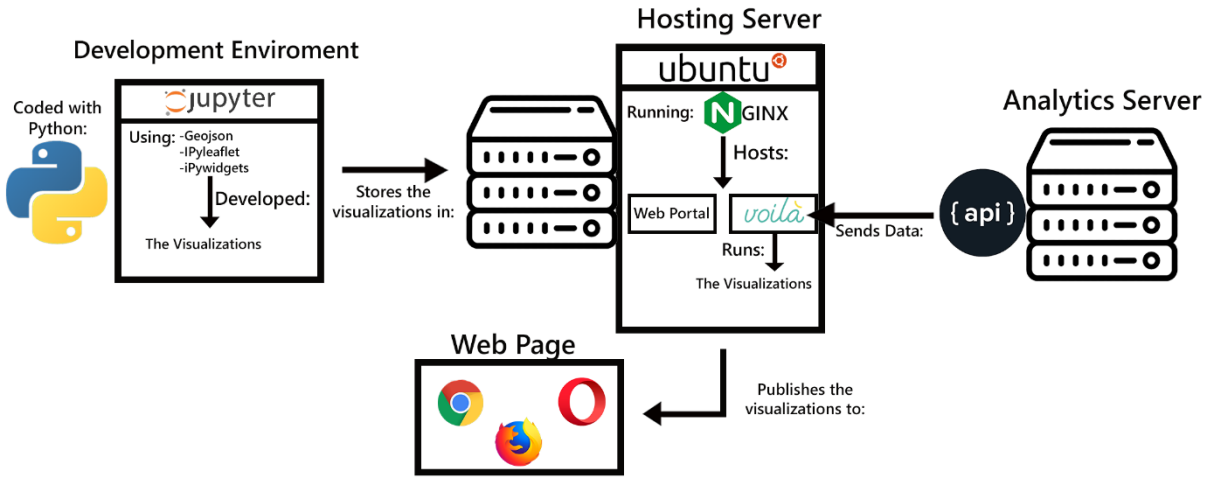


Figure 7 The System's Architecture Design

The system's components worked in sequence, beginning with the analytics server that was responsible for collecting data and then redirecting it to the Development Environment. Which then utilized that data to develop the visualizations, which were interactable maps with various features. These were then stored on the hosting server, that provided the means to run the maps. Finally, the visualizations were imported into the web portal and then presented in the Web Page.

4.1.1 ANALYTICS SERVER

The beginning of the cycle to develop the visualizations prototype is the Analytics Server. This server's responsibility was to gather geographical data through multiple related

tools. Organizing it into common geographical structures and subsequently storing in a dedicated database.

The data structures that offered were relative to whole routes and the segments that compose them. These were then made available to use through the API, in the Geojson format. The API offered the ability to select the data range and area, for further customization ability when designing the maps.

The data that was supplied by the API data ranges from; the slope levels of the routes, their bike count, and the average speed at which cyclists travel through them. The intention was to expand this range of data, as more data sources are integrated into the platform and its API.

4.1.2 THE DEVELOPMENT ENVIRONMENT

The development environment as the name indicates, is where the visualizations were developed. Utilizing Jupyter as coding environment and Python as the language for coding. Both of which are necessary since Python is the language in which the libraries operate, and Jupyter because it's a requirement for Voilà, a hosting tool.

The process to develop the maps began by receiving the data that the analytics server sent through the API link. The data was then read and interpreted with the help of the library, Geojson, and was then prepared and arranged to be utilized by the maps.

The maps were created with the IPyleaflet library which utilizes OpenStreetMap, a free to use geographical database, as the template for base layer of the maps. Afterwards, the maps utilized data that was imported with Geojson to populate routes, zones, and markers into the maps.

Afterwards the library IPywidgets provided complementary tools for the comprehension of the maps, such as map labels and information boxes that complement the routes with data relating to them. As well as control tools that managed the map's zoom levels, full screen mode and current visible layers. The combination of these libraries resulted in fully interactable and explorable maps. Which were then further customized, according to the needs of representing different realities of data. This customization often resulted in the coloring of routes, based on what data type is trying to be highlighted. All the coloring wass

done according to KPIs provided by the platform, which gave increased fidelity to the grading of the routes.

Coloring plays a big part in the understanding of visualizations and therefore different techniques were applied to the visualizations depending on the type of data that was being represented. For instance, the slope of a route is objective and therefore a range of different colors that represent its difficulty is appropriate. Whereas bike count, is more subjective since it is depending on the amount of bike traffic that a city usually experiences, and therefore, color gradients offer a general perspective on the bike passages of each route, in relation to the total amount of traffic that exists.

4.1.3 HOSTING SERVER

Once the visualizations were completed, they were exported and saved on the platform's server, where they can be executed afterwards. Because they were utilizing data that is being sent through an API, they don't need to suffer any coding changes, unless they are for bug fixing or the addition of new features.

The hosting of the visualizations is done through Nginx and Voilà, and they both played a major role in their accessibility. Nginx redirected and managed all requests done to the server and its files and subsequently was also responsible for executing and hosting an instance of Voilà. Voilà's role was to run and provide a dedicated kernel for each visualization, converting them into a web application and therefore accessible without any prerequisites and installations.

The visualizations could all be accessed through the server once deployed, and the deployment of new ones didn't compromise the running status of the server. This was because the server was only focused on running an instance of Voilà, and it was that instance's job to ensure that all files within it could be executed. This model allowed the platform to have multiple contributors to the making of the visualizations and reduced the workload for the maintenance of the server.

Combining both Nginx with Voilà enabled Voilà to be publicly available through Nginx, and consequently the visualizations that were ran through Voilà are also made public in the form of applications. These applications were then referable through a link that only changed depending on the name of the file that was originally executed.

4.1.4 WEB PAGE

The web portal was hosted by Nginx and is situated in the same server that was hosting the visualizations. However, it had a different hosting process then them, meaning that the portal and the visualizations weren't directly connected. To integrate the visualizations into the web portal, the only thing needed was to import the links of the visualizations generated by the server, into an HTML iframe contained within the web portal.

Iframe enabled the embedment of other html content into the web portal's pages, meaning that the maps were loaded displayed in the boxes allocated by the iframe. Once loaded these maps were fully interactable and could even function offline, supporting use cases where connections were unstable. As mentioned previously, there was no setup required, and the only condition for running these apps was an internet connection.

The web portal was then converted into the web page when published by the server, becoming accessible by anyone through its dedicated link.

4.2 THE VISUALIZATIONS

The visualizations were the centerpiece of the project. These were mainly categorized as analytics maps and were developed and produced, with the use of Python, in Jupyter. Utilizing the libraries, Geojson, IPyleaflet, and IPywidgets to create and draw the maps.

To contextualize all the terms that are used as components of the maps, the Figure 8 is provided as a means of visually referencing those components:

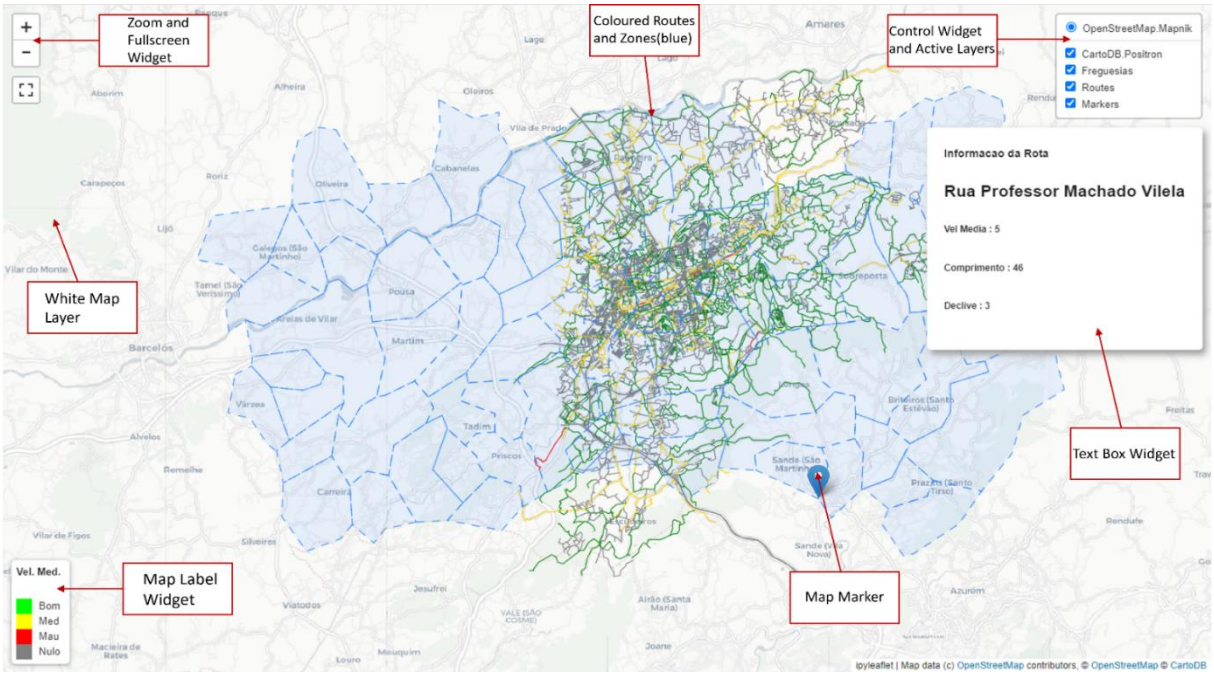


Figure 8 Elements of the Map Visualization

The main reason for utilizing Python in the development of the maps was because it's a very flexible and lenient programing language, versus the counterparts. Which was highlighted in the process of receiving and adapting data that was consumed by the maps.

The decision to use Jupyter as the development environment for Python boggled down to the fact that it's a requirement for Voilà, which was the library used for the deployment of the maps. However, a big advantage of Jupyter was that it was a very simple environment to use. With plenty of features that helped testing and bug fixing, meaning that it had a plethora of upsides for its usage.

Four visualizations were developed during this project, and they were all very similar in their conception so they will be explained together. Beginning with the importation of the libraries that were necessary for the development of the visualizations.

The first step in the functioning of the maps, was to import the data that was being supplied by the platform. The data was relative to the district of Braga and was initially received through static files, for ease of development and testing. However, by the end of the project, a fully operational data API was used. The API's link contained coordinates that

allowed for data filtering, by specifying their start and endpoint, enabling the coverage of different regions.

To ensure the reliability of the data source, a try catch was implemented during its loading, ensuring that the code wouldn't continue, until having its data available, since the API sometimes was prone to failing when first loading it.

The data was loaded with Geojson and subsequently converted to that format, since it was mandatory for the functioning of the maps. Then began the process of data treatment. This was a necessary step after loading it from the API, because the data columns did not have proper value rounding or were not aggregated enough, which was not ideal for the map displays.

The Geojson data structure operated off four major components. The Feature Collection, which was the main level that separated route collections, meaning that each region had a dedicated Feature Collection. Then inside each Feature Collection, was a Feature, which was relative to each component of a region. For example, each Feature in a route collection, was relative to a single route. Finally, inside each of these Features there was two data collections, the first being the Properties, which stored all the data's characteristics, and the Geometry collection which stored the data shape, and all the coordinates of the route.

Once the data preparation was over, the next step was configuring the map. Map configuration as a process was relative to its initial status before loading any type of data. Meaning that, the base layer for the map was chosen alongside its starting position, alongside some usability options, and then the layout size was configured. Afterwards the final step in configuring the map was to add the control tool and layer. This tool allowed for the toggle of any layer added to the map, enhancing user experience by giving the option to turn any layer on and off.

The next step was to display the route, zone, and marker data. Displaying these different map components through IPyleaflet required a single function, that operated on any Geojson data. It was through the Geometry field inside the Feature Collection, that the function knew what kind of shape to display. Routes were displayed through the "PollyLine" property, whilst zones were displayed by being "MultiPolygon", and finally markers needed to

have the “Point” property. In the case that no geometric property was given, the library would default to the “PollyLine” shape.

To compliment the display of the Geojson data, a set of tools was created to enhance intractability and map comprehension. The first tool to be added was the Text Box Widget, which operated in tandem with the display function. It operated and was invoked through either on click or on hover actions for the routes and zones. Its job was to display the features of the segments that were selected, in an HTML white box, inheriting the properties inside the segment’s feature. It required data to be specified for presentation due to the way it used to intake said data, meaning that it was not fully automatic.

The next tool to be added was the coloring tool, which was a function that was manually developed to color routes based on criteria that the Easy Ride platform provided. While there was a native property for the color of the routes, there was no innate function that changed their color. To color the routes the first step was to sift through the route’s properties to match the color that represents the KPIs that the platform provided, to the value of property that was sifted. The coloring function was then summoned inside the main display function of the routes, to change the color property of the route that was being displayed. This process repeated every time a route was displayed on the map, until all were displayed.

The final tool to be added to the map was the Map Label Widget, which provided context to the colors utilized for the routes. At the time of writing, it was quite limited in what could be written inside it, and that was the reason why it presented small, and simple text to serve as the color descriptions.

The final step in creating the maps was to add the finished map to a virtual box function provided by IPywidgets. The virtual box function enabled the map to be displayed inside a container that virtualized the map. This same box was the reason why the additional tools were visible, since the widgets that were presented, all derived from the IPywidgets library, that was integrated into IPyleaflet.

The Figure 9 and Figure 10 are the completed versions of both the average speed and slope maps. Both utilized the coloring functions normally and compared the route data to the KPI values supplied by the platform. The Slope map had a wider color range to accommodate

to the diferent levels of difficulty of slopes, as specified by the KPI. Both maps have the regions layer disabled and are utilizing a white layer so that the coloring is more perceivable.

The

Table 2, Table 3 and provide context on how the KPI's are being used to color the maps. By showing how the values of the properties are being compared and the respective color that

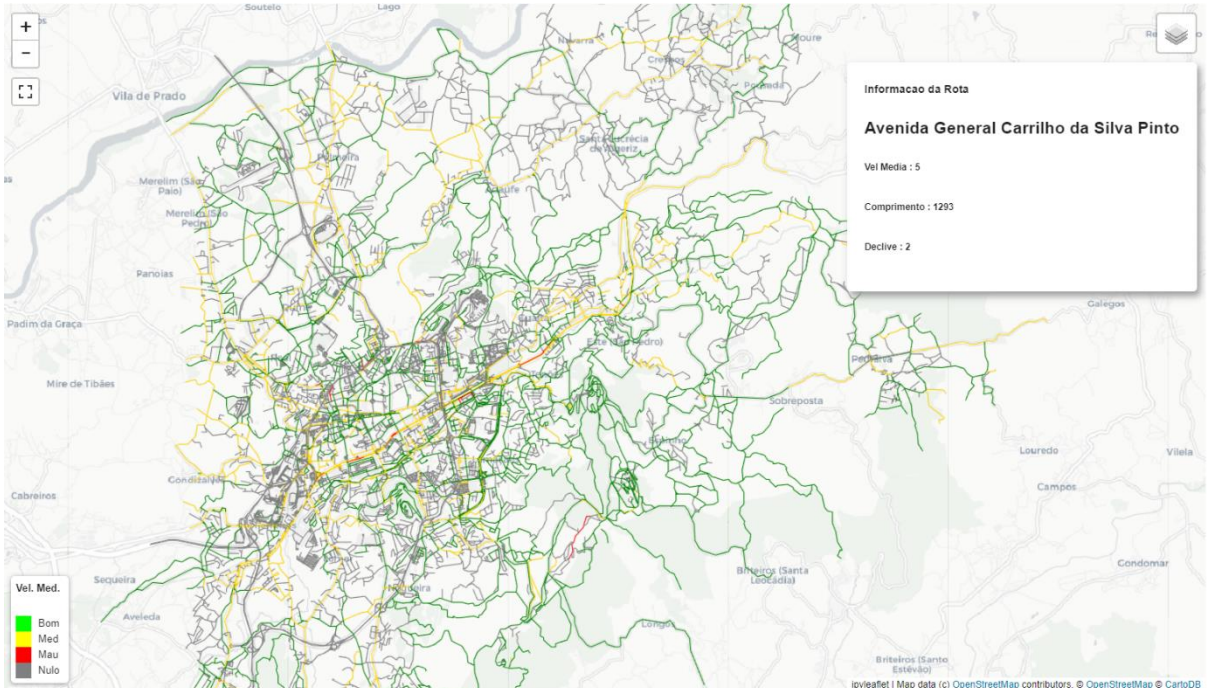


Figure 9 The Average Speed Map

is being attributed to them.

Table 2 Average Speed Colouring Table

Value Ranges	***Categorical Name***	***Colour***
<=3 km/h	Slow	Green(#5DFF00)
3-6 km/h	Medium	Gold Yellow(#FFD500)
>=6 km/h	Fast	Red(#FF3A00)
?	Unknown	Grey(#808080)

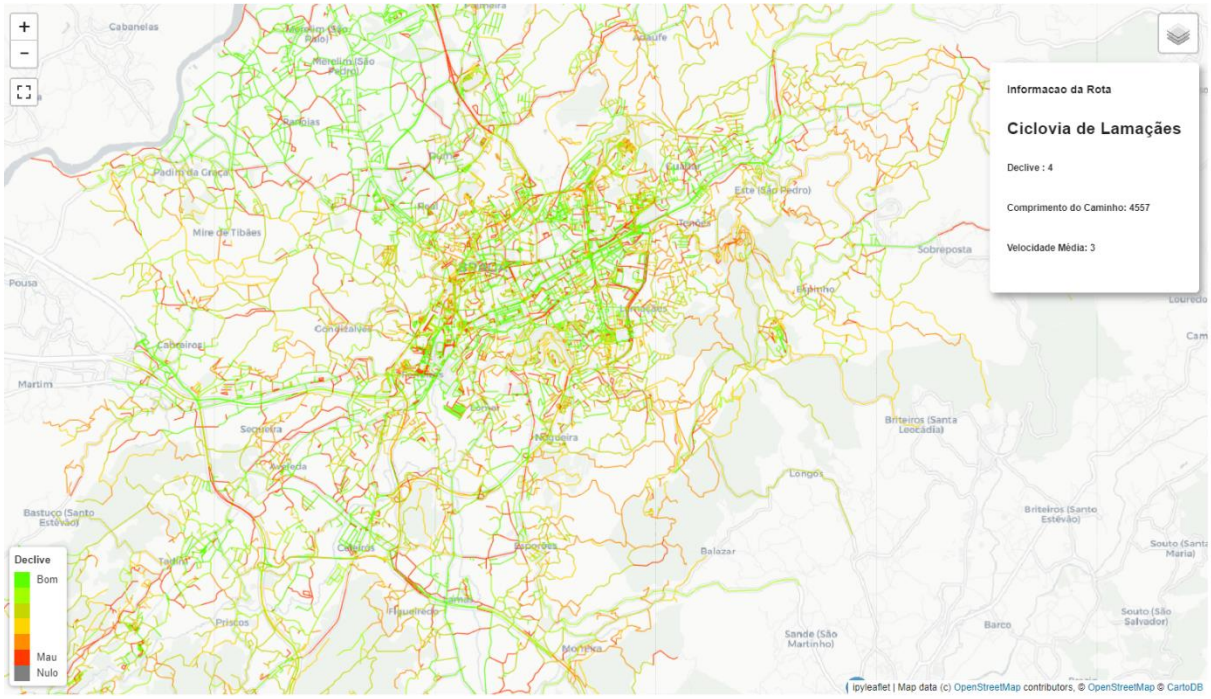


Figure 10 The Slope Map

Table 3 Slope Colouring Table

Value Ranges	***Categorical Name***	***Colour***
0-3 %	Plain	Green (#5DFF00)
3-5 %	Light	Lime Green (#A2FF00)
5-8 %	Medium	Olive Green (#C6D600)
8-10 %	Demanding	Gold Yellow (#FFD500)
10-20 %	Terrible	Orange (#FF8F00)
>20 %	Impossible	Red (#FF3A00)
?	Unknown	Grey (#808080)

The bike count visualization differs in the way it colors it's routes by not utilizing KPIs. It utilized a new function, before coloring, to create the bike usage percentiles. This new function was created using NumPy, a python library that provides additional math operations.

It gathered all the bike count data from the routes and summed it, subsequently applying the percentile function in NumPy. It created five quadrants for comparison, but more could be added for higher detail. The higher the percentile of the quadrant, the more pronounced the color gradient would be, starting with dark red for higher values and ending in orange for lower values.

Figure 11 is the outcome of the bike count visualization. Because it utilized percentiles for the display of information, the data pertaining to the number of bikes is not displayed, since it would defeat the purpose of relativity that is trying to be achieved. For demonstration purposes, this visualization has the region layer disabled and utilizes a white layer for more perceivability.

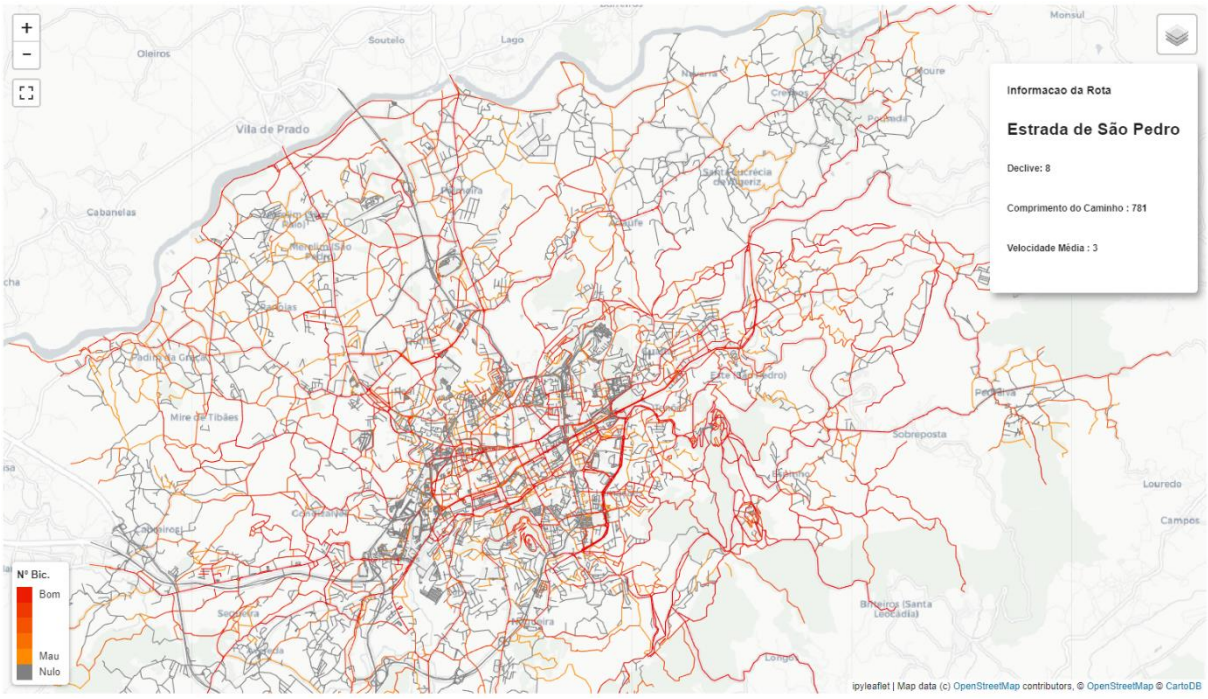


Figure 11 Bike Count Map

Table 4 Bike Count Colouring Table

Value Ranges	***Categorical Name***	***Colour***
Percentile 10	Very Low	Orange (#FF8B01)
Percentile 20	Low	Flush Orange (#FA6F01)

Percentile 40	Medium	Reddish Orange (#F55301)
Percentile 60	High	Bright Red (#F03801)
Percentile 80	Very High	Red (#FF3A00)
Percentile 90	Extremely High	Dark Red (#E60001)
?	Unknown	Grey (#808080)

The fourth visualization that was created, was very experimental, and provided the ability to compare route data through tables as can be seen in Figure 12:

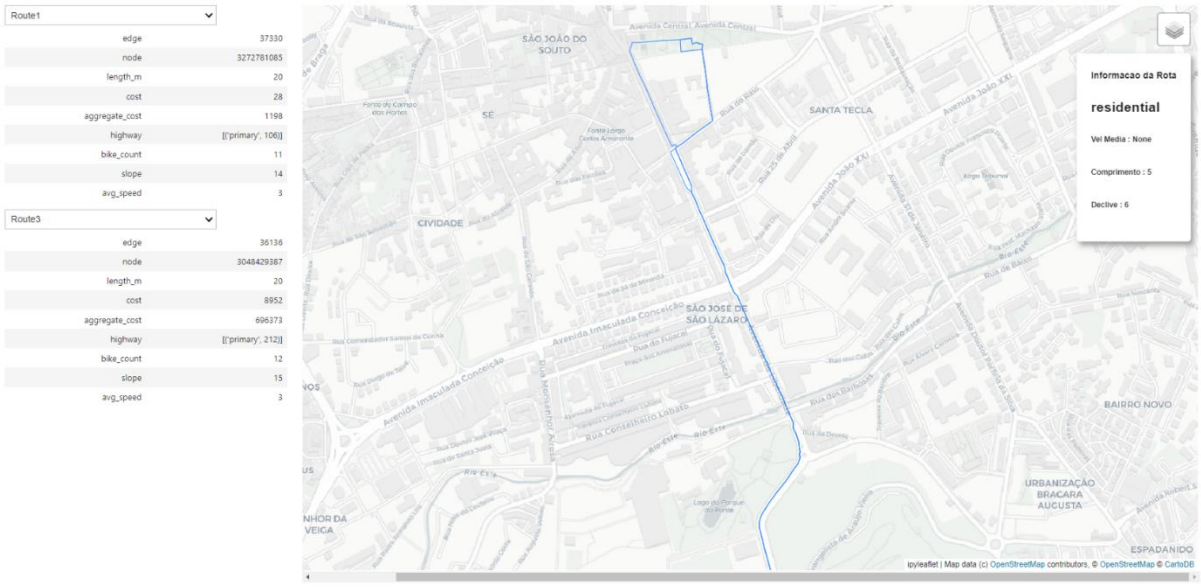


Figure 12 Route Comparison Visualization

For the development of this visualization, some aesthetic sacrifices had to be made, for it to be functional. The first one was that the fields that came from the database had to retain their code names, meaning that the beautification they had on the Text Box Widget was no longer possible. Furthermore, the tables in which the information was displayed were not aesthetically pleasing, due to them being created through HTML.

The major difference between this visualization and the others that were presented, asides from the table widget, was the increased need to create a function that searched through the data. And found each field in a property of a route without originally knowing the properties that it had. Afterwards, the function averaged out the values of the properties of

each segment composing a full route. In the case of string type data, it found the most common word and its count. The function repeated until all properties were averaged. At the end of each iteration the property and its average value were stored inside a list. In the case of string type data, the most common word was calculated through a Counter function.

The result of the function was a list, that only contained the average values of the properties of the route. To then associate the route with its corresponding values, a dictionary emulating a json structure was created. A function coordinating the whole process would then associate the route name to its list of properties. Because this whole process was done in iterations, the routes were named in the order in which they were summoned by the function. Therefore, the first route was called “Route0” because zero was the first number in a cycle. Subsequent route names were incremented by one, until completed.

The next step was to create the comparison table. The table was generated in simple HTML and utilized the newly generated dictionary to display the route data. The table creation function created the initial table and afterwards through a cycle, concatenated each row, that contained the values of a single property. The cycle ended when all properties of the route were displayed in the table. Afterwards, for the table to be displayable alongside the map, it needed to be converted into an IPywidgets object, by utilizing a function that had to intake the table and outputted it as a widget.

The creation of the dropdown boxes was done with IPywidgets, where the values that were selectable, corresponded to the names of the routes available. To be able to dynamically change the information that was seen on the table, a function was created that read the changes done on the dropdown box and outputted the value of the name of the route, to the table function.

With the conclusion of the development of the visualizations, the next step was to deploy them. To deploy the visualizations the first step was to deploy them on the hosting server. The hosting server oversaw the hosting of all the requirements for running the visualizations, from the python kernel to the deployment library Voilà.

The decision to utilize Voilà came from its ability to convert any code developed in the Jupyter IDE into a presentable standalone visualization. Its biggest advantage was the fact that it was very simple to setup and utilize only needing the initial installation setup. Voilà, is a

library that converts Jupyter notebooks into standalone web applications with their dedicated kernels. The process of conversion required the developed notebooks to be ran with the Voilà prefix and additional key arguments, such as the desired hosting port, the specification of the Tornado engine and content settings. During the process of conversion Voilà locally hosted the visualizations on the specified port and attributes them a dedicated link for accessing them.

To make this link accessible by other machines the server utilized a tool that provided a Proxy Pass, which converted the locally hosted link into a public link for usage. The tool utilized by the server was Nginx, which is a web server that provides the ability to reverse proxy. The link that was generated was relative to the instance of Voilà and changed based on the name of the notebook that was being executed.

Voilà's ability to run the visualizations as standalone applications also meant that they could be integrated into any html page with the use of an "iframe". An iframe is an inline frame, that nests a browsing content. The result was an html page that is embedded into the original page. This interaction was what allowed the visualizations to be deployed into the web portal.

The deployment of the hosted visualizations into the web portal was done through the generated links that the server provided, and their embedment into the iframes that were in the web portal. The result was an iframe that summoned the visualization through the hosted link, which would cause Voilà to run an instance of the visualization that could be interacted with. For the links to be invoked by the iframe, an additional parameter that whitelisted the web portal needed to be included before running Voilà, to bypass HTTPS laws that forbade the misuse of content inside iframes.

The major upside of the hosting process used was that only a single instance of Voilà needed to be hosted by the server. By running it without including any file parameters, Voilà could access any notebook inside its directory, meaning that there was no need to separately run each view, allowing for constant support and deployment of new views without restarting and restructuring the server or the web portal.

The choice to deploy the visualizations through Voilà and Nginx came from their documented compatibility. Many options were tried before hand to accommodate different use cases and requirements, but they all ended up failing.

The Figure 13, Figure 14 and Figure 15 are the visualizations published on the web page. They can be freely accessed through the link; http://easyride.dsi.uminho.pt/.

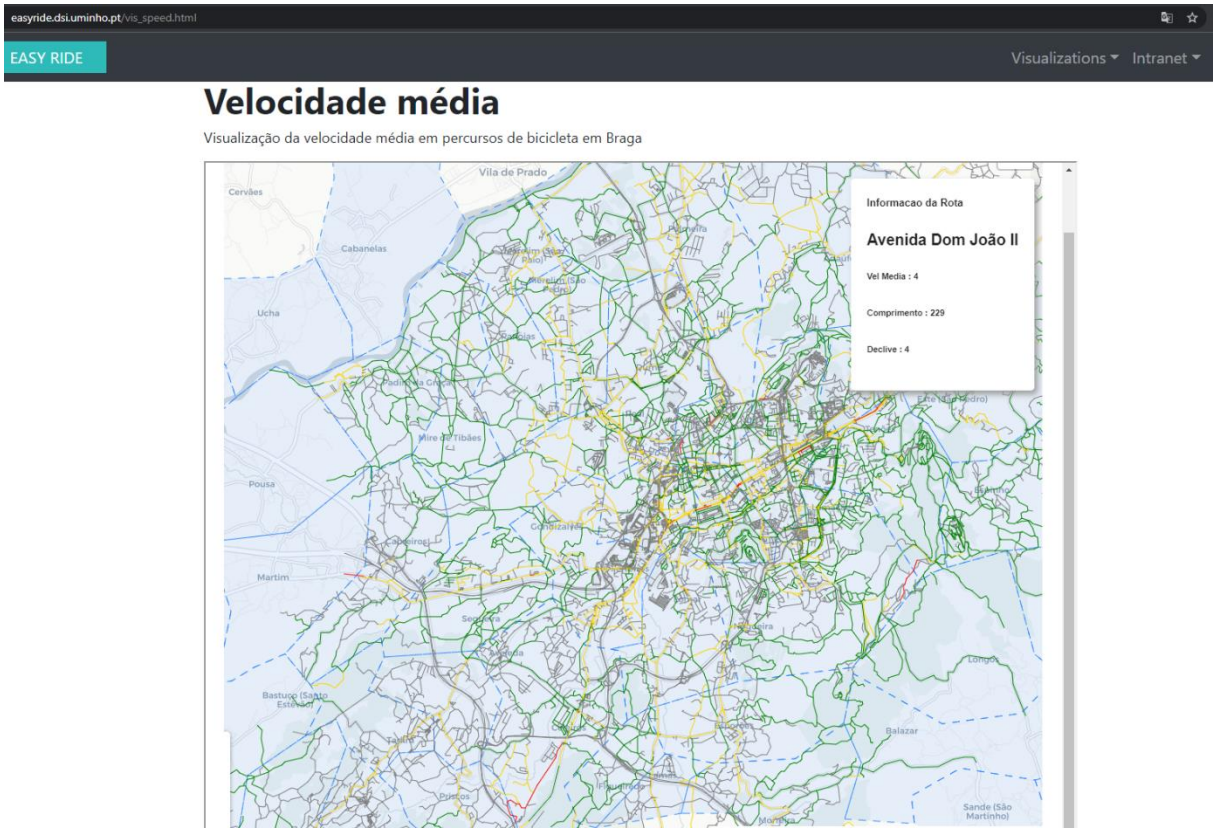


Figure 13 Slope Map Page

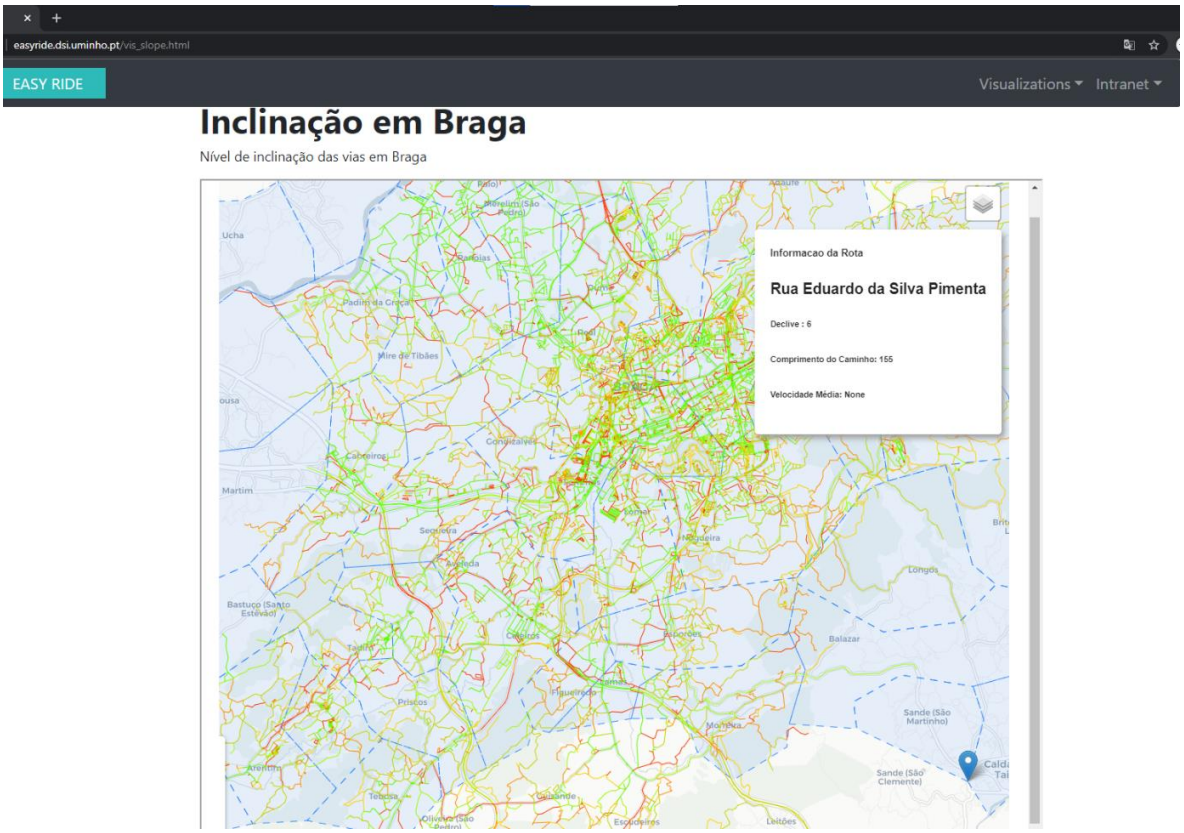


Figure 15 Average Speed Map Page

Figure 14 Bike Count Map Page

This concludes the explanation on how the visualizations operate and function and their deployment onto the web page. More in-depth explanations on the decisions made during the development and the lessons learnt, can be found in the Development Lessons section.

5 DEVELOPMENT LESSONS

This section describes the main lessons learnt from the development of the visualization system. Focusing on the main points of difficulty experienced throughout the project.

5.1 PYTHON VS JAVASCRIPT

The debate between using leaflet through JavaScript or Python is one that is important to discuss. On the surface, JavaScript seems to be the obvious winner. The reason was that leaflet in JavaScript had years of user experience and resources that helped in initial development. Alongside countless add-ons for map customization and functionality that made easier to integrate the maps on a website. On the other hand, Leaflet on python had strict use cases, provided no add-ons, and the ones that it did, were integrated through the library. It also needed extra tools for implementation and lacked support from the community.

Despite many advantages to the usage of Leaflet on JavaScript, the language chosen for coding was Python, and the reason was as follows; Leaflet in JavaScript required a very solid design approach for development, and a set-in stone data structure from the source. Both of which were lacking in an experimental project that aimed to produce a prototype. On the other hand, Python was much more malleable and lenient. The data that was received, can be first curated, and adapted using functions before being consumed by the maps. Not only that but the language was much easier to use and provided more support for tracking down errors through IDE's like Jupyter Lab, something that JavaScript heavily lacked.

As a final note on the debate, python, in most major projects, served as a prototyping language and a proof of concept. For more permanent deployment solutions, it's best to use JavaScript, especially when given more time and resources.

5.2 IMPLEMENTING THE WIDGETS AND TOOLS

The biggest issues began when implementing additional tools, such as the map label and the text box display to the map. Those issues came from the lack of information about the integration of said tools with the maps. IPyleaflet was designed with the objective to

integrate IPywidgets, therefore, the details and use cases of the mentioned tools had to be found on the IPywidgets library.

It required allot of research to find examples and use cases where the widgets were used alongside the maps, and yet, they didn't work properly. The label widget was easier to implement, however, it was bug ridden, having numerous edge cases where it didn't work as intended. The biggest issue was that if the words inside the label went over, a seemingly arbitrary character limit, the widget would try to adjust the contents inside, instead of expanding in size to accommodate the contents. This resulted in warped text and mismatches with the aligned colors, which rendered the widget useless, and was the reason behind only displaying very simple and unsatisfactory text in the label such as "good" or "bad".

The textbox widget had opposite issues to the ones faced with the label widget. Whilst the label widget was mentioned and supported by official documentation, the textbox widget had no mentions of it in any documentation. However, after deep research, one use case was found in a notebook example where it displayed an example in code. Fortunately, most key features of the widget could be found in the example, and the rest were figured out through trial and error.

The text box was relatively easy to setup and customize thanks to the contents inside being in html format. It came with support for click and hover functions that provided real time display of the information in the routes and zones, through either hovering or clicking on them. Unfortunately, the ease of use that came from the html format was also the biggest wall in fully automizing the box for the display of any data source. The reason being that for both aesthetic and functional purposes, all the headers inside the textbox had to be hard coded. Another problem that compounded the issues was the stringent demand, by the function that generating the textbox, to only accept string data. Making it extremely difficult to develop functions that would automatically generate the contents that filled the box, because they had to be given to the function in the form of variables.

Finally, both the map label and text box widgets shared a problem that derived from the way IPyleaflet manages layers. A common complaint during development and use testing was that the text box layer occupied a large portion of the map and couldn't be hidden. This problem couldn't be solved and existed because two issues that were working in tandem. The first issue was that; The visibility property in any layer that was tagged as a "control layer"

couldn’t be changed either internally or externally through a function. This was a reported issue, and the developers were still working on it by the end of the project. The apparent solution would be to change the property to a “map layer” so that the layers could be manipulated the same way as the routes and zones were. Unfortunately, that resulted in an error, meaning that there was no solution for fixing the ability to hide those layers.

5.3 THE COLORING FUNCTION

The limitations that existed in the coloring function were intended. These consisted in fixed parameters such as hardcoded pathing to the properties of the route, the colors, and the value comparisons. Since the objective was to color the map based on KPIs that the project provided, it wouldn’t make sense to develop a fully interactable coloring system, since it would be outside the scope of utilizing the KPIs.

The issues that came from the limitations of the coloring function were that the data sources couldn’t structurally change. Otherwise, the coloring function would cease to work and consequently so would the maps, due to the errors generated. Nonetheless, these issues would only arise if the data source would continue to suffer major changes, which by the end of the prototype it didn’t.

5.4 MAP MARKERS

Markers were conceptualized to integrate the maps as points of interest however they were minimally used in the final project, as no marker data was provided. However, the technology was made beforehand, therefore they remained in the final prototype as a placeholder and proof of capability.

Markers were supposed to be added one by one, from what was shown in the documentation of IPyleaflet. Nevertheless, IPyleaflet provided a grouping function for layers meaning that multiple layers could be combined into one. This was the basis for the marker code that utilized a cycle that would create all “Point” data found in a feature collection of the Geojson data, which would then group all the markers into a single layer.

The marker creation cycle also had a custom text popup function that merged the marker properties, such as its location and type. Creating a single text popup rectangle that

displayed a sentence portraying the marker's description. Enabling full marker support development for the future. Once again, these features suffered the same limitations as the text box. However, creating the marker grouping layer, led to another issue where they would not hide from the map once they were toggled on the control layer. This issue was deemed unworthy of being fixed, due to being relatively minor and markers not having any data interest due to the lack of it.

5.5 The Development of the Route Comparison Visualization

During the development of the route comparison visualization, the issues began with the data structure of the routes not being compatible with the functions used by IPyleaflet to display routes. Therefore, a function was created to intake the data given by the API converting it into a format usable by IPyleaflet.

The biggest issue to arise during the development of the visualizations was that the routes' structure was composed by segments. Therefore, when trying to print its properties onto a table, only the properties of the first segment would be displayed. The difficulty and complexity stood in creating a function that searched and averaged the values of the properties for each segment.

Creating the function to solve the issue of displaying only a segment in the tables, started with creating multiple small functions. The first one store the names of the properties of the routes into a list, that helped other functions find and operate based on the properties of the routes that were found. After they were known, the next function went through each separate route, to retrieve the value of each property of all the segments, and then sum them. Afterwards the resulting value was divided by the number of existing segments and the average for a property would be obtained.

The next problem was creating the tables, resulting from a compounding of issues from incompatibilities between IPywidgets and the VirtualBox. IPywidgets had a specific set of intakes it accepted, meaning that utilizing more appealing tables from other libraries was out of the question due to not being displayable in the virtual boxes. Furthermore, the tables that

were developed had to be interactable with the dropdown widget, that selected the route for comparison.

The first attempt at creating tables was with the html widget that was used for the text box widget. However, it failed because there was no way to append new data rows to it, due to only accepting string type inputs. The fix was to separately create an html table and append each row with the properties and values of the routes. Afterwards this table was converted through an IPywidgets function into an IPywidgets output. Resulting in displayable tables inside the VirtualBox. However, with this approach, the function that linked the tables with the dropdown widget stopped working. Manipulating the VirtualBox in which the tables were in didn't work. And changing the VirtualBox as a display method would result in the maps not being displayable, which removed the initial purpose of the visualizations.

This visualization wasn't the focus of the platform therefore no more time could be allocated to its development meaning that it had to remain with the presented issues until further fixing was attempted.

5.6 THE DEPLOYMENT OF THE MAPS

The choice to deploy the visualizations with Voilà was made due to it sharing the same contributors as IPyleaflet. Meaning that they were fully compatible and therefore the maps were fully viewable without suffering any issues related to the way they were hosted.

The next planned step of deployment was to utilize MyBinder, which is an online service that provides hosting and sharing of executable environments such as Jupyter, from online repositories like GitHub. It has no additional costs and serves as a public service for people to execute their environments. The only requirements were: Having executable code developed in an environment, the configuration file to mount all pre-requisites and that those files were hosted in a Git repository. Yet the plan didn't come to fruition as that would imply that part of the prototype would be depending on external factors, which couldn't be relied upon during the development of a project.

To avoid dependencies of external services, the project acquired a server with a clean install of CentOS for all hosting needs. This server was utilized not only for the hosting of Voilà but also the API that provided the data for the visualizations. The Voilà documentation only

referred to Ubuntu as an example of private server hosting however Voilà was supported by CentOS as well.

The main goal was to run the visualizations through Apache, which is a an open-source HTTP web server for delivering web content. It redirected and managed communication between clients and the servers, while having the big advantage of redirecting Ips and therefore avoiding the exposure of the backend that results from hosting directly from the port. This is where unexpected problems with deployment began as the maps were unopenable with the Apache implementation. The issue wasn't apparent at first, however because of Apache's ability of redirecting access to the files, Voilà couldn't utilize its own resources properly. The error occurred when Voilà tried to access JavaScript resources, it would infinitely scour through file paths attempting to find said resources until it crashed.

Replicating the issue using only port forwarding techniques was unsuccessful. Which meant that the problem was either in the code or it came from Apache itself. After multiple code configurations with the intent of fixing the problem, no solution was found. This meant that additional Apache configurations had to be done to fix the issue. Many fixes were attempted to get Apache to run Voilà without problems, however none were successful, therefore a decision was made to move to Nginx.

Nginx is very similar to Apache as it is also open-source and is HTTP based, and in addition its coma with innate reverse proxy meaning it was already configured for proxy passing which is used for converting locally hosted links into public ones. Nginx was the featured HTTP based server in the documentation of Voilà which meant that it was expected to have higher compatibility. However, the same issue that plagued the Apache approach was found. No matter what pathing Voilà took, it always resulted in endless looping until crashing.

The final attempted fix was switching to Ubuntu, with the hopes that by removing all external factors that may interfere with compatibility, that it would work. Since the documentation of Voilà mentioned Ubuntu alongside Nginx as the private hosting solution. And the fix did work, the views were now deployable. However, when executing Voilà inside of Nginx's configurations, some errors occurred.

By the end of the deployment phase, it was decided that it was too cumbersome to pursue a full redirection with Voilà, and that the visualizations would simply be running on a

separate port. It was a temporary solution for the prototype and the hope was that some improvements on the compatibility of Voilà and the HTTP based servers would be made by the developers.

6 EVALUATION OF THE RESULTS.

With the conclusion of the prototype, the next step was to evaluate its initial success and determine where it failed and succeeded. Because of the nature of visualizations, some aspects of it were hard to evaluate due to the subjective nature of appearance and esthetic. Nonetheless, the effectiveness of these visualizations can still be evaluated by their ability to convey the information that the project aimed to deliver.

6.1 REVIEW OF THE ESTABLISHED OBJECTIVES

The first step in the evaluation of the project was a revision of the initial objectives established in the Objectives section of the Introduction and observe how well they were realized.

6.1.1 OBJECTIVE 1

The first objective was relative to the gathering of requisites by interviewing stakeholders and other entities that may be interested in utilizing the visualization protype or are invested in the success of it.

This objective suffered some tribulations in its completion simply due to the experimental nature of developing the prototype. While there were stakeholders that were interviewed and there was information gathered on how these visualizations were supposed to look like by the end of the project, not all of it was done in the beginning. As the visualizations were being developed, they were subsequently shown to stakeholders for testing and approval. This resulted in incoming feedback, and therefore new discussions emerged that impacted the visualizations and instilled changes based on the criticism received.

When taking everything into consideration, it was possible to ascertain that the objective was accomplished adequately, despite not following exactly what was initially established. The core necessity of developing the maps according to the specifications of the entities interested in the visualizations was accomplished. Meaning that the objective served its main purpose when it was initially created.

6.1.2 OBJECTIVE 2

The second objective was pertaining to the exploration of technologies for the creation of the visualizations, with the requirement of them being open source.

This objective had to be accomplished obligatorily before any work had been done on the maps, contrarily to the first objective. The main reason being that any work done on a specific technology, before analyzing the whole array of available technologies, could result in wasted development time. From having to upgrade to a more robust and sophisticated solution that other technologies provide.

The objective was accomplished according to what was established. All technologies were explored beforehand, mainly because none were pre-established and therefore a thorough investigation of the landscape of available tools for developing maps was done. The search led to the eventual decision to utilize Leaflet, which was a popular and supported library for developing maps in OpenStreetMap. The decision to then utilize python came from the leeway that it gave when beginning to develop experimental projects.

The technological search had some failures, relatively to the lack of investigation for deployment of the maps. Whilst JavaScript is easy to deploy on any website due to having an engine by default on all major web browsers. Python needed to use convoluted deployment techniques. Not enough research was done into implementing Voilà on a server, which caused several issues while setting up the server.

Overall, despite the shortcomings in researching the deployment of the maps, the objective was completed successfully. However, a lesson was learnt into also researching the feasibility of deploying the chosen technologies, before beginning their development.

6.1.3 OBJECTIVE 3

The third objective pertained to the development of the visualizations and the assurance that all the pre-requisites that were established are properly accomplished. These were mainly the needs of the stakeholders and the choice of technologies to be used.

The development of visualizations was the most important phase of the project since the whole objective was to produce them. Therefore, the expected outcome, are well put together visualizations that convey information that is clear to understand and detailed

enough that invite into further exploration of the data. Alongside with them properly working and offering the promised features. And to that extent the completion of the objective partially succeeded. The reason being that the maps were the focus of development they were delivered successfully, however they weren't robustly implemented and lacked some features.

The maps worked properly despite needing some initial seconds of loading, they were mostly dynamic requiring very little maintenance, and the parts that weren't dynamic, were so due to either design choices or limitations in the development tools that were used. Most of the requirements and requests of the stakeholders were also implemented. The ones that couldn't be implemented resulted from the afore mentioned limitations in the development tools, such as limitations in the labeling of the map, or the lack of ability to hide the text box.

Despite accomplishing the main goal, the prototype failed to deliver on the complementary graphics that were presented in the design phase. These weren't high on the priority list for the stakeholders, even so, they were promised, and none were delivered. The reason being that there was no supporting data for the creation of these graphs alongside the fact that the choice of deployment through Python and Voilà meant that there were additional constraints to the fruition of these graphs.

With these factors in mind, the lesson to be learnt was that there shouldn't be an overpromise to the contents to deliver. It's best to focus on the main objective first and if the conditions allow for the development of additional features, then those should be designed afterwards and properly thought out before assuming the responsibilities to create them.

6.1.4 OBJECTIVE 4

The fourth and last objective was relative to providing public access to the visualizations, and subsequently receiving user feedback. With the intent to make a serious review of the project's success.

Relatively to the publicization of the views, whilst they were made publicly available, that doesn't equate to them being published and ready to be used. The reasoning being that the process of publishing the visualizations through Voilà, made it so that they suffered in their presentation to the target audience, and therefore they can't be seen as a final product of the project.

The key difference between publicly available and published, is the main factor as to why many opinions weren't tallied about the success of the visualizations. Only the stakeholders that were in charge, and interested in the success of the visualizations, provided meaningful feedback, and contributed to changes made to them. As mentioned in the evaluation of the first objective.

The initial purpose when designing the objective was to ensure that there would be a focus on properly sharing the visualizations once they were completed. However, due to the experimental nature of the project and the choice of utilizing python for the prototyping of the visualizations, this objective could not be completed successfully.

The lesson to be taken away was that, for a project aimed to be accessible by the public and to be utilized in a meaningful way, there was not room for convoluted ways to deploy a project, such as the ones that were used through Voilà. It's best to start development with more robust tools such as JavaScript, that were already popular and proven to be effective. Then risking the scenario where the implementation leads into poor user experiences, that subsequently tarnish the potential ability of the project.

Therefore, the fourth objective wasn't completed successfully. Consequence of the choices made to deploy the project. Resulting in the lack of public feedback, inhibiting the ability to properly evaluate with external feedback.

6.2 General evaluation of the Project

When analyzing the project as a whole and observing the results that were obtained, the general verdict was that it was a success considering the overall work done and the shortcomings had. The reasoning was that the area of micro mobility and cycling was still unexplored for decision making, and therefore developing solutions and contributions becomes harder with the lack of available resources. These difficulties experienced from development were also lessons to be taken away after the project's conclusion.

Creating the visualizations gave insight into how maps operate and that the smallest details like coloring and line thickness can have into the legibility of them. However, the biggest lesson in the field of development was that it's best to utilize tried and proven technologies instead of opting for experimental ones, even if they seem like a better option.

The choice to not develop in JavaScript had major consequences in the deployment of the maps that resulted in numerous headaches during the process of hosting the maps.

Whilst python offered more flexibility when utilizing Leaflet in the first stages of development, as progress was made, it became more and more restrictive. Because the Python version of the Leaflet library was an adaptation of the of the original one created in JavaScript. Resulting in broken and missing features that constricted the decisions made to the customizations of the maps. The problems continued towards the deployment phase where many workarounds had to be made to properly deploy Voilà and subsequently the visualizations. Whereas developing through JavaScript would require minimal work for deploying the visualizations into the web portal.

Despite many criticisms posed to Voilà, the objective is not to discredit it, as it was a very good tool for deploying python projects of any kid that are developed in Jupyter. However, the failure was in not understanding that its uses are more adequate for private or internal applications, rather than something that needs to be made publicly available and utilized frequently.

The visualizations progress the field of cycling and mobility solutions in a conceptual level yet fail to be easily replicated, thus lacking impact in the general landscape of these fields. Even when utilizing open-source libraries and easy to learn coding languages, the fact remains that the visualizations presented carry too many deployment constrictions to be considered a viable option for integration in solutions that aim to have an active and large userbase.

All in all, the progress made through the development of the visualizations can't be deemed as a failure. Due to offering a legitimate way of presenting routes and other geographical data, without requiring any setup. Proving that through freely available tools, there is the possibility for anyone to develop and create cycling tools. That can aid their community and decision makers into creating a healthier and more aware environment for cycling. With hopes instigating changes and improvements into the cyclability of cities and its riders.

7 CONCLUSION

This section marks the end of the dissertations and briefly presents all the work done, alongside the findings that were obtained. Highlighting the outcome of the project in conjunction with the difficulties and challenges that were had to produce it.

The conclusion is often divided into three sections, the Synopsis, Discussion, and Future Work. Each aiming to provide brief context into the overall work put into the project, what was obtained, and the continuation.

7.1 SYNOPSIS

This dissertation had as an objective to contribute to the field of cycling analytics through the creation of visualization tools. With the aim of aiding in decision making so that the cycling infrastructure of cities can be improved.

The goal of contributing to the field of cycling analytics was achieved, by using maps that were populated with data related to routes and other geographical structures. Which had associated usage information such as the average speed in which they were circulated through. The creation of these maps was made possible with the help of Python and its supporting libraries, chosen to support the usage OpenStreetMap, so that the visualizations could be fully open-source and contributed to by anyone.

The development of these visualizations began with the exploration of all related work that could aid in the discovery of techniques and concepts for the creation of the maps and the presentation of data that was contained in them. Afterwards several technologies were tried and tested, with the aim of finding a set of python libraries that could aid in development of the maps.

The remaining dissertation focused on explaining and discussing the process and work involved in the development of maps. By extensively detailing the usage of the selected coding libraries; Geojson, IPyleaflet and IPywidgets. Ending by detailing the deployment process that involved the usage of Voilà, and how the maps were eventually displayed on the web portal.

7.2 DISCUSSION

With the project concluded and the impact of the visualizations analyzed, it could be deemed that the main objectives of the project were completed successfully. Save the exception of the development of complementary graphs that were meant to aid in the understanding and perception of maps.

Even so, the development of the project allowed for a better understanding of the study field of cycling analytics, and the surrounding concerns and focus points. Reaching a higher level of maturity in the development of tools and proper usage of visualization techniques for conveying information.

The dissertation aimed to provide visualization tools to aid in decision making and in that, maps were produced that provided a general scope of the city in study and highlighted the current state of the roads in the city. These tools were also capable of being fully adaptable, allowing for the input of different data sources, hopefully motivating in the integration of various cities other than Braga, so that they can also participate in the process of improving cycling and micro mobilities within them.

The main lessons learned were in the process of deployment, where Python showed some limitations, mainly in the field of user experience. Therefore, justifying the usage of languages with less features like JavaScript, that offer more robust integrations for HTML.

Other lesson that came from the development were to not rely on the usage of libraries that have allot of levels of abstraction such as IPyleaflet, since they eventually run into limitations that are hard to bypass.

The evaluation process lacked empirical data to support the overall success of the project since it did not dispose of much time to be presented and made publicly available. However, the feedback from stakeholders was overall positive and the hopes are that as data availability becomes higher so will the maps become more detailed and provide a better understanding of the scope of the city.

7.3 FUTURE WORK

The aim with future work is to work on the failings that the project had. Those being the deployment process and the limitations in the tools used to develop the maps. Which lead to the maps lacking features that could've helped in portray a bigger and more detailed picture in the state of cycling of cities.

Therefore, the aim for future work is to attempt to create and develop the visualizations in JavaScript, comparing them to the versions in Python, and analyze the impact that a language switch up can have in development of the maps. Once the preferred development tools are chosen, the goal is to expand upon the number of features and data that are presented in the maps, with the aim of providing more data and for it to be visually pleasing.

The final concern is to work on better ways to deploy the maps, so that eventually the platform can be contributed to and by many different users, leading to the creation of a crowd-sourced analytics platform that will change the cycling landscape.

BIBLIOGRAPHY

Akbar, M. A., Sang, J., Khan, A. A., Fazal-E-Amin, Nasrullah, Shafiq, M., Hussain, S., Hu, H., Elahi, M., & Xiang, H. (2017). Improving the quality of software development process by introducing a new methodology-Az-model. *IEEE Access*, *6*, 4811–4823. https://doi.org/10.1109/ACCESS.2017.2787981

Arellana, J., Saltarín, M., Larrañaga, A. M., González, V. I., & Henao, C. A. (2020). Developing an urban bikeability index for different types of cyclists as a tool to prioritise bicycle infrastructure investments. *Transportation Research Part A: Policy and Practice*, *139*(July), 310–334. https://doi.org/10.1016/j.tra.2020.07.010

Aslak, R., J, M. T., Aslak Burkhard, R., Meier, M., Rodgers, P., Thomas Jelle Smis, M., & Stott, J. (2005). *Kent Academic Repository Full text document (pdf) Enquiries Citation for published version Document Version UNSPECIFIED Knowledge Visualization: A Comparative Study between Project Tube Maps and Gantt Charts 1 Introduction: The Power of Visual Metaphors i*. 388–395.

Barneveld, A. Van, Arnold, K. E., & Campbell, J. P. (2012). Analytics in Higher Education: Establishing a Common Language. ELI Paper 1. *ELI Paper*, *2012*(1), 11.

Beecham, R., Wood, J., & Bowerman, A. (2012). A visual analytics approach to understanding cycling behaviour. *IEEE Conference on Visual Analytics Science and Technology 2012, VAST 2012 - Proceedings*, 207–208. https://doi.org/10.1109/VAST.2012.6400550

Behnisch, M., Hecht, R., Herold, H., & Jiang, B. (2019). Urban big data analytics and morphology. *Environment and Planning B: Urban Analytics and City Science*, *46*(7), 1203–1205. https://doi.org/10.1177/2399808319870016

Behrendt, F. (2016). Why cycling matters for Smart Cities. Internet of Bicycles for Intelligent Transport. *Journal of Transport Geography*. https://doi.org/10.1016/j.jtrangeo.2016.08.018

Boehm, B. W. (1988). A Spiral Model of Software Development and Enhancement. *Computer*, *21*(5), 61–72. https://doi.org/10.1109/2.59

Burkhard, R. A., & Meier, M. (2005). Tube map visualization: Evaluation of a novel knowledge visualization application for the transfer of knowledge in long-term projects. *Journal of Universal Computer Science*, *11*(4), 473–494. https://doi.org/10.3217/jucs-011-04-0473

chart-doctor/visual-vocabulary at master · ft-interactive/chart-doctor · GitHub. (n.d.).

Chen, S., Guo, C., Yuan, X., Zhang, J., & Zhang, X. L. (2015). MovementFinder: Visual analytics of origin-destination patterns from geo-tagged social media. *2014 IEEE Conference on Visual Analytics Science and Technology, VAST 2014 - Proceedings*, 239–240. https://doi.org/10.1109/VAST.2014.7042509

Computing, C. (2013). Book Review: The tower and the cloud: Higher education in the age of cloud computing. In *Christian Education Journal: Research on Educational Ministry* (Vol. 10, Issue 1). https://doi.org/10.1177/073989131301000117

Conner, C., Samuel, J., Kretinin, A., Samuel, Y., & Nadeau, L. (2020). A picture for the words! Textual visualization in big data analytics. *ArXiv*. https://doi.org/10.13140/RG.2.2.25351.83360

Cooper, A. (2012). What is "Analytics"? Definition and Essential Characteristics. *CETIS Analytics Series*, *1*(5), 1–10.

Docherty, I., Marsden, G., & Anable, J. (2018). The governance of smart mobility. *Transportation Research Part A: Policy and Practice*, *115*(October 2017), 114–125. https://doi.org/10.1016/j.tra.2017.09.012

Elliott, A., & Urry, J. (2010). Mobile Lives. In *Mobile Lives*. https://doi.org/10.4324/9780203887042

Figueiredo, A.P. e Vale, D. S. (2018). *BikeFriendlyIndex – Um indíce para avaliação da amigabilidade de um concelho para a utilização da bicicleta enquanto modo de transporte urbano*.

Furht, B., & Villanustre, F. (2016). Big data technologies and applications. In *Big Data Technologies and Applications* (Issue September 2018). https://doi.org/10.1007/978-3-319-44550-2

Gledson, A., Dhafari, T. B., Paton, N., & Keane, J. (2019). A Smart City Dashboard for Combining and Analysing Multi-source Data Streams. *Proceedings - 20th International Conference on High Performance Computing and Communications, 16th International Conference on Smart City and 4th International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2018*, 1366–1373. https://doi.org/10.1109/HPCC/SmartCity/DSS.2018.00226

Golfarelli, M., Rizzi, S. A., Golfarelli, M., & Rizzi, S. (2020). *This is the final peer-reviewed accepted manuscript of : Rights / License : A Model-Driven Approach to Automate Data*

Visualization in Big Data Analytics. *19*, 24–47.

Hsiao, I. Y. T., Lan, Y., Kao, C., Li, P., Journal, S., April, N., Hsiao, I. Y. T., Lan, Y. J., Kao, C. L., & Li, P. (2017). *International Forum of Educational Technology & Society Visualization Analytics for Second Language Vocabulary Learning in Virtual Worlds Published by : International Forum of Educational Technology & Society Visualization Analytics for Second Language Vo*. *20*(2), 161–175.

Larman, C., & Basili, V. R. (2003). Iterative and incremental development: A brief history. *Computer*, *36*(6), 47–56. https://doi.org/10.1109/MC.2003.1204375

Lee, D., Kim, J., & Hahn, M. (2014). Density Map Visualization for Overlapping Bicycle Trajectories. *International Journal of Control and Automation*, *7*(3), 327–332. https://doi.org/10.14257/ijca.2014.7.3.31

Lovelace, R., Goodman, A., Aldred, R., Berkoff, N., Abbas, A., & Woodcock, J. (2017). The propensity to cycle tool: An open source online system for sustainable transport planning. *Journal of Transport and Land Use*, *10*(1), 505–528. https://doi.org/10.5198/jtlu.2016.862

Nikolaeva, A., te Brömmelstroet, M., Raven, R., & Ranson, J. (2019). Smart cycling futures: Charting a new terrain and moving towards a research agenda. *Journal of Transport Geography*, *79*. https://doi.org/10.1016/j.jtrangeo.2019.102486

Petersen, K., Wohlin, C., & Baca, D. (2009). The Waterfall Model in Large-Scale Development. In F. Bomarius, M. Oivo, P. Jaring, & P. Abrahamsson (Eds.), *Product-Focused Software Process Improvement* (pp. 386–400). Springer Berlin Heidelberg.

Potential, G. (2020). *Gross Potential for Cycling Cycling potential ranking*. 13–16.

te Brömmelstroet, M., Nikolaeva, A., Nello-Deakin, S., van Waes, A., Farla, J., Popkema, M., van Wesemael, P., Liu, G., Raven, R., de Vor, F., & Bruno, M. (2020). Researching cycling innovations: The contested nature of understanding and shaping smart cycling futures. *Transportation Research Interdisciplinary Perspectives*, *8*. https://doi.org/10.1016/j.trip.2020.100247

Wang, Z. J., Turko, R., Shaikh, O., Park, H., Das, N., Hohman, F., Kahng, M., & Chau, D. H. (2020). CNN Explainer: Learning Convolutional Neural Networks with Interactive Visualization. *IEEE Transactions on Visualization and Computer Graphics*, *2626*(c). https://doi.org/10.1109/TVCG.2020.3030418

Watson, H. J. (2011). Business Analytics Insight: Hype or Here to Stay? *Business Intelligence*

Journal, *16*(1), 4–8.

Weber, P. (2008). User-Generated street Maps. *October*, 12–18.

Zhao, D., Ong, G. P., Wang, W., & Zhou, W. (2021). Estimating public bicycle trip characteristics with consideration of built environment data. *Sustainability (Switzerland)*, *13*(2), 1–13. https://doi.org/10.3390/su13020500