ABSTRACT
Alzheimer’s disease is the most frequent cause of dementia in the western population. It is a neurodegenerative disorder marked by a cognitive and behavioral impairment that significantly interferes with social and occupational functioning. There are many ways to obtain data from the brain, but it is very difficult to analyze it in text or tabular form because of the complex structure of the brain, having a very large number of regions. Three different connectivity metrics were used, for 114 regions of the brain. A tool using data visualization and statistical analysis was developed, aiming to analyze the brain data and its connectivity, in order to understand what are the relevant metrics and regions associated with Alzheimer’s disease. In the first phase, low-fidelity prototypes were developed aiming to understand the dimensions of the available data. Later, more useful idioms were generated, along with several statistical analysis tests, used to bring more useful insights. The interface contains four different screens, using several information visualizations techniques and additional results from the statistical tests, providing multiple discoveries, through analysis and interaction, of several groups of patients, in different stages of Alzheimer’s disease. To validate the system, usability tests were performed with users who validated the usability and utility of our solution.

Author Keywords
Alzheimer’s disease, Brain Connectivity, Information Visualization

ACM Classification Keywords
H.5.2 Information interfaces and presentation: User Interfaces.

INTRODUCTION
Today, we live in a time where we are collecting data almost about anything, but that data is composed mostly by characters and text, which makes it difficult to understand, explore and analyze, especially when it comes to connectivity data. Our capacity to absorb and store visual information is much more powerful than reading, because our brain process visual content 60,000 times faster than text [2].

The human brain is naturally organized into a complex system whose topological descriptions have been represented as a structural/anatomical connectome of interconnected cortico-cortical axonal pathways and a functional connectome of synchronized interregional neural activity [8]. To compose an anatomical connectome, it is necessary to carve through a spatial terrain, while for the functional one, it's all about capturing fluctuations in activity over time. Given this challenge to depict the huge and complex brain networks, it is important to build easy-to-use and efficient visualizations, which can be used in neuroanatomy, neurodevelopment, cognitive neuroscience and neuropsychology. The image should be loyal to the method and raw material it reflects, but clarity and intuitive design should also be a priority. We see the relevance of this balance when observing the transition that brain connectivity has undergone, going from anatomical to functional to connectional space (Figure 1).

Alzheimer’s disease (AD) is the most frequent cause of dementia in the western population [7]. It is a neurodegenerative disorder marked by cognitive and behavioral impairment that significantly interferes with social and occupational functioning. AD is generally assumed to be caused by degeneration of neurons starting in the hippocampus, later spreading to the temporal and parietal cortex, and finally involving most cortical areas. This leads to believe that AD may have abnormal functional brain connectivity patterns. Currently, definite diagnosis can only be made postmortem, therefore, early and accurate diagnosis of AD is not only challenging, but is crucial in the perspective of future treatments.
RELATED WORK
In the recent years, there has been an increasing number of tools developed on visualization and analysis of brain connectivity and other health-related records. Some of them focus on brain structure and connectivity and others on medical records mostly with temporal data. For this reason, the examined studies were separated into 2 categories: Brain Visualization Tools, and Medical and Statistics Visualization Tools. Regarding the first category, some of the examined studies focus on achieving structural and anatomical views, which are very important to analyze in neurodegenerative diseases because more often than not, the modifications of the brain regions volumes and shapes can be a disease biomarker. Others concentrate on the relationships between brain regions, thus aiming to present connectivity views to visualize the brain network. Other combine this all together but risk to become too overwhelming, cluttered and therefore difficult to analyze and extract valuable information in a short time.

To visualize brain data, two common layout techniques are used: spatial layout techniques, which take into account the anatomy of the brain, and non-spatial layout techniques, which do not. Non-spatial techniques usually include the 2D node-link diagram, along with correlation matrices and connectograms. Spatial techniques use mainly a 3D model of the brain, using also a node-link diagram and sometimes complemented with color layers to distinguish regions of the brain or fiber’s activity. Although spatial visualizations allow the researchers to visualize the brain in its true form with all structural patterns, the produced presentations feel cluttered. To overcome this challenge, the user is given the option to filter which connection they want to see. When interaction is not possible, e.g. in static reports, anatomical planes of the brain, such as sagittal, coronal and axial planes, replace the 3D model. Other information visualization techniques such as dendrograms, distribution graphs, and line and point plots are used to complement the analysis of the data.

Regarding the second category, to visualize the EHR records, the standard approach consists of visualizing data over time. One of the commonly used layout techniques is the timeline, where the focus is on finding patterns that can help with the diagnosis, and on the analysis of the data, which would be very difficult to do with the data in text format. Apart from the timeline visualizations, other techniques are used, such as radial and sequential displays, chord diagrams, treemap charts, and sunbursts. Most of the studies examined in this section feature some kind of interactivity in their tools, which usually follow the Information Visualization Mantra: overview first, zoom and filter, details on demand [3]. Along with the visualizations, statistical methods are used to process the data and complement the goals of the visualization, be it for finding patterns algorithms or analyzing groups of patients to understand where new patients can fit and therefore predict events. Statistics also play a major role in some of these tools by simplifying the very large data sets and extract the most important information.

However, there is still a need for tools that combine the two aspects of the above categories, visualizations and statistical analysis. As our tool focuses on data of patients with mild cognitive impairment and AD, we aim to address this shortcoming by creating visualizations to analyze functional brain data in an intuitive and simple way, while also using statistical methods to complement the visualizations, in order to better understand patterns, changes and distributions of data among several groups.

REQUIREMENTS ANALYSIS
For the reason that this work was carried out in partnership with Instituto de Biofisica e Engenharia Biomdlica (IBEB), the first step was to understand the scope of the project and therefore define a set of requirements for the solution we wanted to build. We considered that talking with professionals in the field of neuroscience research was the best way to define the requirements.

Apart from attaining knowledge about the brain, current work was discussed in meeting with professional from IBEB, to learn about and understand what kind of work exists, and what is missing or could be improved. Finally, we defined the following system requirements:

- The system should use the AD data provided by IBEB
- It should be possible to select the metric to analyze
- Only one metric must be analyzed at each time
- It should be possible to analyze one or more brain regions at the same time
- It must be possible to analyze different patient states at the same time
- It must be possible to analyze the groups of patients by state
- The system must have visualizations which allow analyzing the variance of the groups of patients
- The system must have visualizations which allow the analysis of the significance of the data’s values among the groups of patients
- The system must have visualizations through which it can be concluded if the values of 2 distinct regions are independent
- The system must use statistical tests to extract additional information from the data

ARCHITECTURE
One of the most important first steps in software design is the definition of the architecture. The description of an application’s structure and how it will function will influence every step of the development from then onwards. It defines the problems we might encounter in the implementation phase and makes it much easier to make decisions and manage all sorts of change.

After deciding to use R as our main language, we started to research how to bring what we were building in R. That is when we decided to implement our solution using Shiny, for all the advantages it would it brings. As a result, the communication between the client and the server is completely handled by Shiny, removing this aspect from the concerns of our implementation. With Shiny it is possible to host standalone apps on a webpage and even build dashboards, which
meets our needs to create this solution. Shiny apps can also be extended with CSS themes, htmlwidgets, and JavaScript actions.

The data processing was handled entirely by R.

From then on, we researched the best tools to create the UI components we wanted to include in our application, like graphs, tables and control inputs. We relied on Shiny and additional R packages to accomplish this, by using HTML, CSS, and JavaScript to create those elements and to provide our solution with interactivity. The complete architecture of our application can be seen in Figure 2.

**TOOLS**

At first, we were considering using d3.js, a JavaScript library for data visualization [6] to build our solution. However, the initial analysis of the available data was performed using R statistical language [5].

The R language can be expanded by installing packages available from online repositories, such as The CRAN. In addition to this, there exist an increasing number of external visualization libraries integrated with R. Using those, we were able to add interactivity to the visualizations, since it was a crucial feature of the visualizations we wanted to create for our solution.

Another advantage of using those libraries was that it allowed us to do all this without leaving the R development environment. RStudio [4] is a free and open-source IDE for R. RStudio Desktop version 1.1.453 with R 3.4.4 was used to develop all the projects code.

**BRAINVIS**

The user interface of the functional prototype follows a dashboard structure, including a sidebar and a body. The sidebar includes several menus which lead to different pages when clicked. In addition, it includes various filtering options: metric selection, which is common to all pages, and other filters and settings particular to the selected page.

The available menus are Distribution analysis and Statistical analysis. The latter includes three sub-menus: Normality Test, Compare mean and median and Compare correlation. In total, there are four menus leading to different pages. Each page includes some sort of visualization and, in the case of the Statistical analysis pages, they also include data tables.

**Distribution analysis**

The Distribution analysis screen features three types of visualizations: histogram, density plot and box plot. It is possible to see one of the visualizations at a time, and the selection is made in a group of three buttons.

In addition, there is also a numeric input to define the number of bins of the histogram (Figure 3 B), and a slider input to adjust the bandwidth of the density plot (Figure 4 A). These two inputs can only be seen when the corresponding visualization is selected and their purpose is described further below. The sidebar also features a dropdown to select the brain region we want to analyze, and it is accessible in any of the three visualizations (Figure 3 A).

Regarding the visualizations, as illustrated in Figure 3 and Figure 4, the histogram and density plot contain multiple containers: one for the visualization which contains all the four patient states, and four for the individual visualizations of each state. These two visualizations show the distribution of the values in each patient state. All these visualizations have the brain region on the x-axis and the corresponding values on the y-axis. The boxplot, on the other hand, features the states of the patients on the x-axis, producing up to four distinct boxplots, one for each state, hence it is redundant to have four more visualizations for each state.

All the generated visualizations have several interactive actions available to them, such as tooltips, zoom in and out, clicks on the legend to select and deselect groups, through the use of the plotly’s modebar (Figure 3 C).

We decided to use the histogram, the density plot and the box plot to visualize the distributions.

**Figure 3. Histogram plot**

The first one represents the distributions of the values by separating them into bins and counting the number of observations in each bin (Figure 3). Our first prototype had two types of frequency plots: frequency and relative frequency. The former shows the count of the observations in each bin, and the latter shows the proportion of the observations in each bin, expressed in percentages.

The histogram features a tooltip when hovering on a bar - it shows the count of the observations of the corresponding state group and the name of the region (Figure 3).
The density plot (Figure 4) is a variation of the histogram described above, as it also represents the distribution of the values but in its turn, it uses kernel smoothing to plot them, allowing for smoother distributions by smoothing out the noise. One advantage over the histogram is that it allows determining the shape of the distribution, which can be more challenging using the histogram because the density plot is not affected by the number of bins used. Because in this plot we want to compare the distributions of the values across the multiple groups, we used filled density plots, which allows us to visualize what are the regions where the values overlay. We also added a rug plot to the density plot. With this, we can see every value plotted on the x-axis.

In addition, we allow the user to adjust the bandwidth of the density plot. We include a slider input in the sidebar (Figure 4 A), which starts with the value 1 (means the default bandwidth is used). A bigger value results in a wider bandwidth and consequently in a smoother visualization, while a lower value results in a narrower bandwidth and therefore in a density curve with many spikes.

This plot features a tooltip on hover, both on the density plot area and on the rug plot, showing the corresponding value and the name of the region (Figure 4).

The boxplot (Figure 5), similarly to the plots described above, also shows the distribution of the values, but it does it through various calculated key values such as the median, the minimum, the maximum, lower and upper fences, and the first and third quartiles. An advantage of this plot when compared to the histogram and the density plot is that it takes less space and do not overlap when comparing groups. Another interesting feature of the boxplot, which is missing in the other two visualizations referenced above, is that it allows us to discover outliers immediately.

The boxplot features a tooltip on hover (Figure 5), showing the key values referenced above, at their corresponding position. We also added all the data points on top of the boxplot and they too have a tooltip on hover.

**Normality Test**

The Normality Test screen is the first sub-menu of the Statistical Analysis menu and it has two containers (Figure 6). The one on the left contains an interactive table that contains four columns: one for the region and the other four for the patients’ states. The rows contain the region name in the first column and the p-values of the Shapiro-Wilks test of the corresponding group in the other four columns. This test is used to test the normality of a set of values.

The container on the right holds a placeholder text when no row is selected, indicating that the user can interact with the table. It is possible to click on the table rows and after this, a Q-Q plot is generated in that container. This plot draws the correlation between the values of the selected region and the normal distribution. In this case, it draws the values of the four groups of patients states, so we can see all the points of each group (colored in the same color scale as the previous visualizations) and a reference line based on the data quantiles, for each group. The table also allows ordering by clicking on the rows name, which can be useful to select regions with greater p-values of a certain state, for example. There is also a search input which can be used to search for a region in the table.

In addition, it is possible to select multiple rows of the table. When this is done, the selected rows are painted in blue and the plots of the selected regions are generated side-by-side, and, if more rows are selected, the corresponding plots are drawn down below (Figure 7). The Q-Q plot is one of the most used plots to analyze the values against a normal distribution and to gain insights such as the skewness (right or left) and the tail (light or heavy) of the observed values. Also, as we are comparing multiple Q-Q plots side by side, we can analyze if the values from the same group (for example, the N state group) of two distinct regions have similar distributions and common scale. The main takeaway from this plot is
that if the values form a straight line, it can be said that they follow a normal distribution.

Comparing Mean and Median
The second sub-menu of the Statistical Analysis menu is called Compare Mean and Median (Figure 8). This screen, similarly to the Normality Test, also has two containers, where the one on the left also includes a table and the one on the right includes several visualizations and other additional information, such as statistical tests’ results. The main goal of the data and visualizations displayed on this screen is to understand if the values among the four patients states are significantly different.

The table contains three columns: one for the region, one that says if the data is normally distributed (true or false) and a third one with the p-value of a statistical test - ANOVA test, if the data is normally distributed, and Kruskal-Wallis test if it is not. This is described on the top of the table to inform the user. The second column gets its value from the table of the previous screen, and the value is true if the data of every state (N, EMCI, LMCI, and AD) follow a normal distribution.

This table is also interactive and includes a search input to filter regions, but unlike the table described in section , it only allows to select one row. When the user clicks on a row, several components are drawn on the container on the right side, including a box plot and a density plot, drawn side-by-side, a summary of the data, and results from two statistical tests. The box plot is the same as the one described in the section , with the addition of the p-value of the corresponding statistical test (ANOVA or Kruskal-Wallis) on the upper part of the plot. The mean plot is very similar to the box plot, showing the mean values of each group and also the standard error bars. It can be concluded if there can be or not a significant difference in the values among several groups, by observing if the error bars overlap in a pair of groups. Only if there is no overlap we can conclude that the difference may be significant, but it must be confirmed with additional statistical tests. The mean plot also has statistical data on the top part of the plot - the p-values of the corresponding paired statistical test (Welch Two Sample t-test or two-sample Wilcoxon test). This is described in detail in section . In addition, below these two plots, there is a table with the summary of the analyzed data such as the number of observations, the mean and the standard deviation of each group. The results of the ANOVA or Kruskal-Wallis are below in text format for additional consultation, as well as the results of the Pairwise comparisons using Wilcoxon rank sum test or the Tukey multiple comparisons of means (Figure 13).

Correlation analysis
The third and last sub-menu of the Statistical Analysis section is the Compare correlation screen (Figure 9). This screen contains the exact same table as the previous sub-menu (section ), with the difference in the interaction feature - it is possible to select up to two rows. When one row is selected, a simple line plot is drawn, with the values on the x-axis and the states on the y-axis. When a second row is selected, a scatter plot is drawn. With the values of one of the selected regions on the x-axis and the values of the other regions on the y-axis. This plot contains all the four groups of patients represented by points. Also, a regression line is drawn, as well as the shadow of the confidence interval. By analyzing this plot, it is possible to conclude if the data of the two selected regions have positive (the values increase together), negative (the values decrease together) or no correlation (no observed pattern, straight regression line). In addition, it is possible to compare multiple groups at the same time to conclude, for example, if the correlation in the N state is similar to the correlation in the AD state, and even in the states in between.
Another visualization of this section is the correlation matrix (Figure 10). It shows the correlation coefficient between sets of brain regions. As it is not possible to analyze more than one group at a time with this kind of visualization, there is a group of buttons to select which patient state the user wants to visualize (Figure 10 A). There is another group of buttons where it is possible to select the method to calculate the correlation coefficients - Pearson or Spearman (Figure 10 B). The matrix can be quite big when showing all the possible regions so we added a filtering option, a slider input where it is possible to select the range of values of the correlation coefficient the user wants to see in the matrix (Figure 10 C). As the range is reduced, only the regions with the corresponding coefficients are shown on the matrix (Figure 11).

![Figure 10. Correlation screen and its components (part 2 of 2).](image)

![Figure 11. Correlation matrix with filtering options and tooltip.](image)

Also, to navigate the matrix more easily, several features of the modebar can be used, such as zoom in and out, panning and box selecting. The colors of the matrix correspond to the correlation coefficient and range from blue (-1) to red (1). When hovering one tile of the matrix, the information of the two corresponding regions and their correlation coefficient is displayed (Figure 11).

**Statistical analysis**

The statistical analysis is used to aid and complete the visualizations we created. We decided what statistical tests to use by following a well-defined process, which can be seen in Figure 12. First, the Shapiro-Wilks test is used to find out if a set of values follows a normal distribution. This is necessary to know what other statistical tests can and can not be used on that data because many of them require the data to follow a normal distribution). Those tests are the parametric tests. If the data doesn’t follow a normal distribution, those tests can not be used so we use their alternatives, the non-parametric tests.

![Figure 12. Statistical tests used.](image)

We show the results, specifically the p-value, from the Shapiro-Wilks in a table. We can assume that if the p-value is greater than a certain alpha value, the distribution of the data is not significantly different from a normal distribution. In other words, we can assume the normality. The default for the alpha value is 0.05, but it can be changed by using the slider input in the sidebar. This input represents the confidence interval and the default is 95% (corresponding to the 0.05 alpha value), so for example, if the user changes it to 90%, the alpha value used to compare with the p-value will be 0.1. This input is available in all the screens of the Statistical Analysis (Figure 6 A).

Although the table with the p-values contains only tabular data, we added green coloring to the background to facilitate the reading and interpretation. The color intensity increases gradually the closer the p-value is to 1. If the p-value is less than the alpha value, then the background is left in white. Additionally, if all the states follow a normal distribution (by having all the p-values greater than alpha), we also paint the background of the region name in solid green, to highlight those regions.

Following this, the tables referenced in previous sections, the column "Is Normally Distributed" is true for those regions that have the data of all the states normally distributed. From here, we perform another statistical test, but the test we use depends on the value of that column. If the value is true (the data is normally distributed), then we can use the ANOVA test, if not, we use the Kruskal-Wallis tests, and the results of each, the p-value is shown in the third column of that table, and also in the box plot. The coloring of that table follows the same gradient scale as in the table referenced above - the greater the p-value, the greener the background of that cell. Here it is also possible to configure the confidence interval and consequently the alpha value. Regarding the mean plot referenced in section Comparing Mean and Median, it includes the p-values of another statistical test - if the data is normally distributed, the Welch Two Sample t-test is used, if not, we use the two-sample Wilcoxon test. The values are displayed in the upper part of the mean plot and there are always three p-values. The reason for this is that we need to select a reference group, to compare all the others groups to it. To accomplish this, we provide a group of buttons, where the user can select which reference group he wants to compare
So if the user chooses, for example, the group of the AD state, then the mean plot will include the three p-values corresponding to the comparison of the AD group to all the other three groups (N, EMCI, and LMIC). We can conclude that if the p-value is lower than the alpha value, then the data of one group is significantly different from the data of the other group.

Additionally, the complete results of the ANOVA test (degrees of freedom, Sums of Squares, Mean Squares, F-ratio, and p-value) or the Kruskal-Wallis test (chi-squared and degrees of freedom, and p-value) can be seen in text form (Figure 13). The results from the paired tests, Tukey multiple comparisons of means (parametric) and Pairwise comparisons using the Wilcoxon rank sum test (non-parametric) can also be analyzed in text form. They include the p-values for every group combination. The results of the Tukey test also include the difference in the observed means, the lower end point of the interval and the upper end point.

EVALUATION
The evaluation of our visualizations were conducted mainly with users without any connection to the field of medicine or neuroscience, and the goal was to test the usability of the application.

Initially, we made a brief presentation of the scope of our, the motivation, the solution, and how would the usability testing would be conducted. Apart from that, we explained the different names used in our application (like regions and metrics) are very technical and it is not important to know what they mean, because its irrelevant to the performance of the tasks. We also emphasized that it is the interface that is being tested, and not the user.

Next, we explained what kind of data we were using in our application and how it was obtained, and then showed to the user some of the functionalities and all the screens, to facilitate the initial adaptation of the user to the system. Then, we let them use and investigate the application freely for around 5 minutes.

Following this, we asked the user to perform 12 tasks, in form of question, making the user discover where to go and what to observe in order to get the answers. While we were observing the user, we were timing each tasks and registering the errors the user made. At the end, we asked the user to fill a small System Usability Survey (SUS), and briefly discuss the experience with the user.

Results
There were several tasks that the users executed much faster which were very simple tasks and require little interaction, and there are most complex time which took more time.

We also observed some variation in the total time of the tasks. The reason for existing tasks in the middle with very low times is that those tasks were executed used the screens with a lot of statistical analysis. As most of our users didn’t have the knowledge to analyze that information, we asked some simpler questions which the users answered analyzing merely the visualizations, or simply finding a certain region in the table.

Most tasks do not have any errors and the maximum number of error is 2 and it happened only in one of the tasks.

In Figure 14 we can see the variation of the time in the tasks.

Discussion
The results of the usability tests are satisfactory. There was not a lot of errors in the tasks the user performed. The score System Usability Survey is 77.22 which is above average (68), and the feedback obtained from the users was good. The users thought the system was very intuitive and very few thought there was any inconsistency in the system (observed in question 6).

During the user testing, we saw the need in explaining several statistics concepts to the users. Some basic knowledge of statistics is necessary for the understanding of the application and several users didn’t have these notions.

CONCLUSIONS AND FUTURE WORK
Data is each day a big part of our lives and is up to us the use we give to it. Alzheimers is greatly affecting the elderly nowadays and it is a great threat for the future populations. Its
causes are unknown as well as its cure, but the visualization of the brain can contribute hugely to those discoveries. There are multiple methods of neuroimaging available and being used and the main challenge is to present the acquired data in an intuitive way and to choose what is relevant to visualize and analyze.

In this project we present a visualization application consists of multiple visualization techniques which, with the help of statistical analysis, will help researchers and clinicians to visualize brain structure and connectivity intuitively and in a flexible manner, to allow them to explore the data and seek for what is relevant.

Initially, we created simple idioms to understand the dimensions of the data, as it had multiple variables, representing the many regions in the brain. Then, following an iterative and incremental approach, we started to compose more useful visualizations.

Our interface consists of multiple components, such as the distribution analysis which allow analyzing the distribution of the data among several patient’s groups, using visualizations like histograms, density plot and box plot.

Apart from this, our solution has a big statistical analysis component, which is used together with the visualizations to gather better insights.

With our solution it is possible to compare two distinct metrics and conclude if their values are different, across four patients states(N, EMCI, LMCI, AD). It is possible to compare how the values of one metric vary in the same patient group and compare it with other groups. With this, we can see, for example, if the values increase or decrease gradually from the N state to the AD state. With the results from the statistical tests, we can compare several groups and conclude if their values are significantly different.

We provide several filtering and configuration settings to give the user some flexibility over the visualizations and also over the statistical tests that are used.

In addition, we provide multiple interaction features, such as tooltip on hover, zooming, selecting and deselecting groups, and also we allow to interaction in the data tables which results in the generation of visualizations.

Nevertheless, there are many improvements and extension that can be made to our solution. One of the main requisites is to allow the upload of new data to the application. Also, filtering the updating data, to discard some patients, or using the patient’s age and gender on the filter is also an improvement that could be made. It would also be very useful to upload the data of a single patient and then, the application, by comparing the new data with the existing data and other statistical values could estimate which is the state of the patient.

We observed that some metrics do not have data for certain region. The region dropdown could be improved to remove that region, if it happens.

Also, the correlation matrix could be improved. Due to a large number of regions, it can be hard to analyze. A search box could be added to search for a region name in the x ou y-axis.

With these improvements, these visualizations would bring even more value and more insights and therefore help to understand what are the most relevant metrics and regions of the AD that can indicate in which state category the patient is.

REFERENCES


