

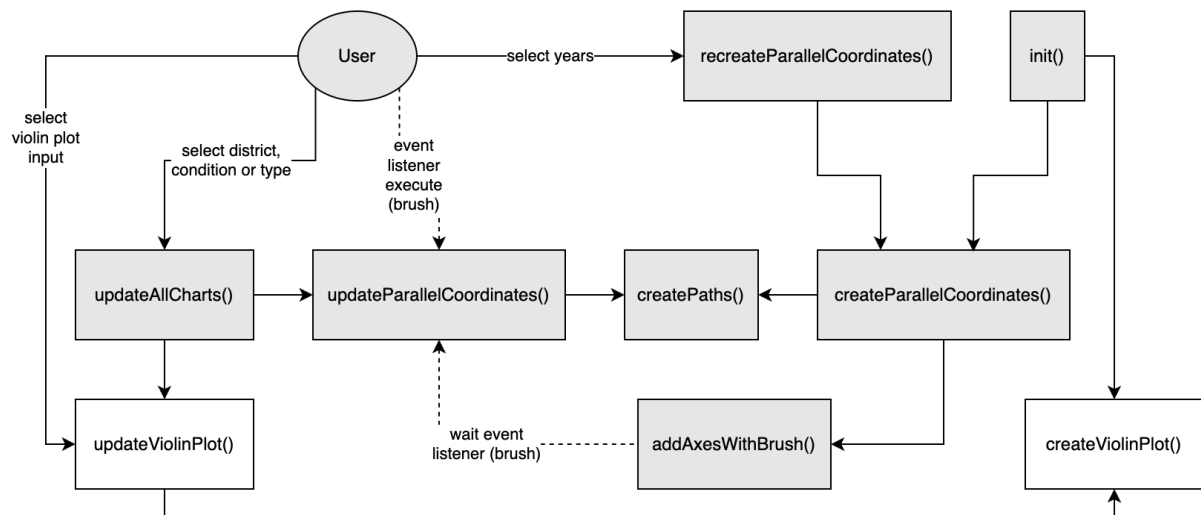


Checkpoint IV: Second Prototype

Group: G23

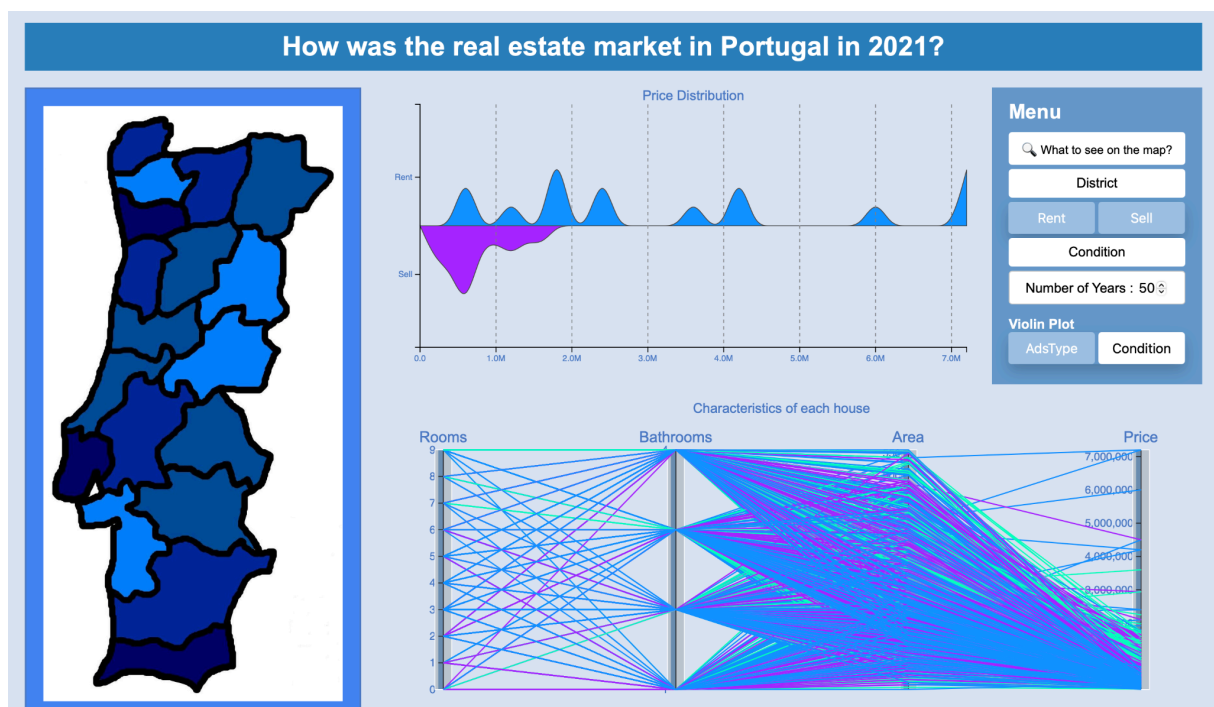
Date: 2024/10/18

Prototype Architecture



Regarding the last checkpoint (gray boxes), the methods **createViolinPlot** and **updateViolinPlot** have been added to the file **violin_plot.js**, since this file is responsible for implementing the functionalities of the Violin Plot.

Dashboard Layout



Regarding the dashboard layout presented in the previous checkpoint, the differences include the implementation of the Violin Plot (replacing its old placeholder) and the addition of buttons in the controllers menu to allow interaction with the Violin Plot.

Data Processing

For this checkpoint, we only made a small change to the data processing explained in the previous checkpoint: removing outliers from the prices. We noticed that our function to remove price outliers was not working correctly, as it was only removing outliers from house prices for sale and not from rental prices.

Chart Interaction

This chart interacts with two buttons on the controller, allowing to compare the **types of ads** (AdsType: “Rent” or “Sell”) and the **conditions of the properties** (Conditions: “New” or “Renovated”). When clicking on one of the buttons, the `selectViolinPlot(show)` function is triggered, where the `show` parameter indicates the chosen comparison.

This function calls the `updateViolinPlot(data, selector, show)` method, which recreates the violin plot according to the option selected. In this way, data analysis becomes dynamic, allowing the user to explore different categories simply and efficiently.



Chart Integration

We still maintain both the global data and filtered data in the `script.js` file. However, since the user might want to view only houses for sale in the Parallel Coordinates and Choropleth Map, after having explored the relationship between the prices of houses for rent and for sale in the Violin Plot, the Violin Plot data cannot be filtered by AdsType when this is the dimension being visualized. Similarly, it cannot be filtered by Condition if that is the selected visualization for the Violin Plot.

To handle this, we added a new variable (named `violin_data`) that filters the information based on these specific restrictions. This variable is always updated at the same time as the filtered data (whether through integration with the Controllers Menu or the Parallel Coordinates, as explained in the previous checkpoint report) and is used by the Violin Plot in the same way that the filtered data is used by the Parallel Coordinates.