Assignment 2: Creating Interactive Visualizations

188.308 Information visualization

Artjola Ganellari 12046001

Noah Watzal 01633746

Introduction

In this assignment, we will implement three interactive visualizations based on the previous assignment's designs and concepts on drawing relevant and important relationships of variables in the House Sales for Seattle and King County dataset. Following the feedback from the first assignment, we have decided on executing some improvements focusing on specific target users, setting distinctive goals, and using only a subset of the data that may be of higher interest to target users.
After conducting the qualitative analysis of this assignment, we had to make the decision on which interactive visualization tools to use to better explain our goals. Based on our past experience, we decided on using Python, Bokeh, and Folium libraries. Bokeh is a python library for creating interactive visualizations for modern web browsers, with which you can create JavaScript-powered visualizations without writing any JavaScript yourself. Likewise, Folium is again a python library which makes it easy to visualize data that's been manipulated in Python on an interactive leaflet map. Utilizing these two libraries, we have created three different interactive visualizations that will come in handy to a single target user, real estate agents.

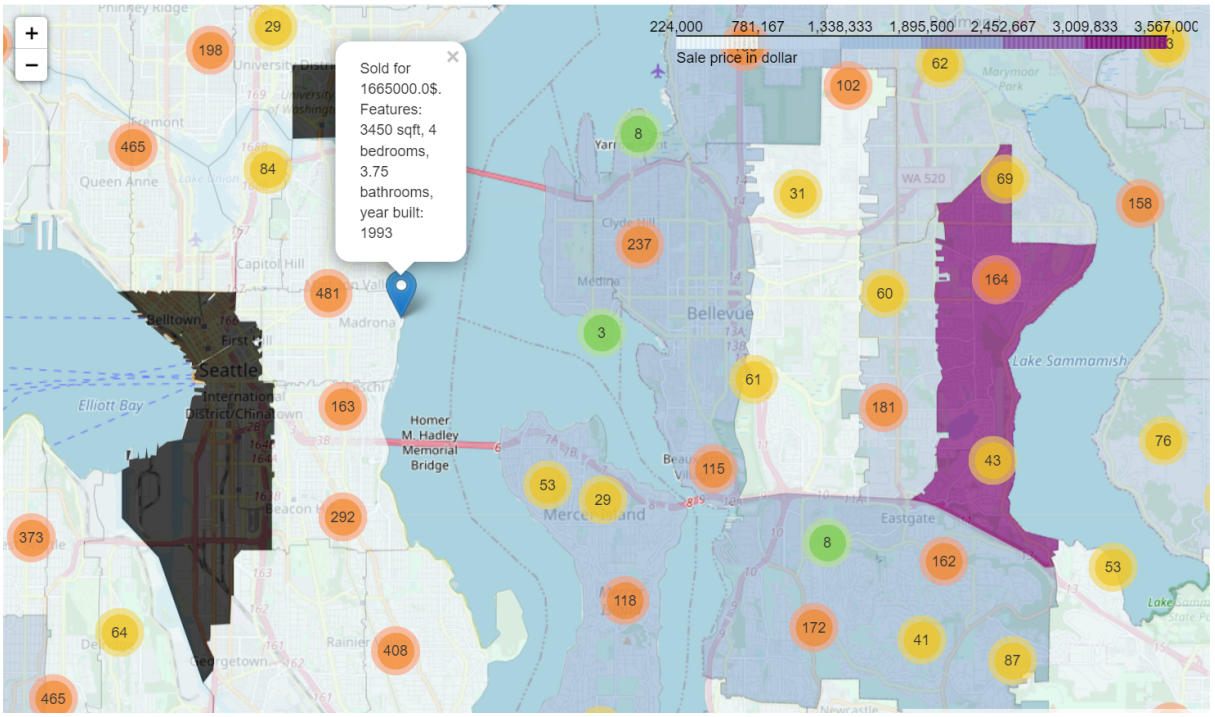
Users and Goals

In this assignment, we have decided on focusing only on a distinguished group of interest which are real estate agents. This target group may be highly interested in the visualization analysis of house sale prices and other characteristics as they can help them get an insight into the customers' behavior or maybe even competitors' strategies. Additionally, they can understand parts of the market activity that may profit the company.
In order to come in assistance of such a target group, three different goals have been assigned:

1- Gathering useful information on the distribution of the prices of the houses in different locations of the King County
2- Draw inferences on the periods when the highest number of transactions (total number of houses sold in a specific month) occur and their relationship with the monthly mean house prices.
3- Making remarks on the effect that the grade of a house has on its price during different periods.
In order to accomplish these goals, we will be using only a subset of the whole dataset, including variables such as price, yr_built, zip code, date, grade, longitude, and latitude.

Interactive Visualizations

1. Visualization: Geographical Map



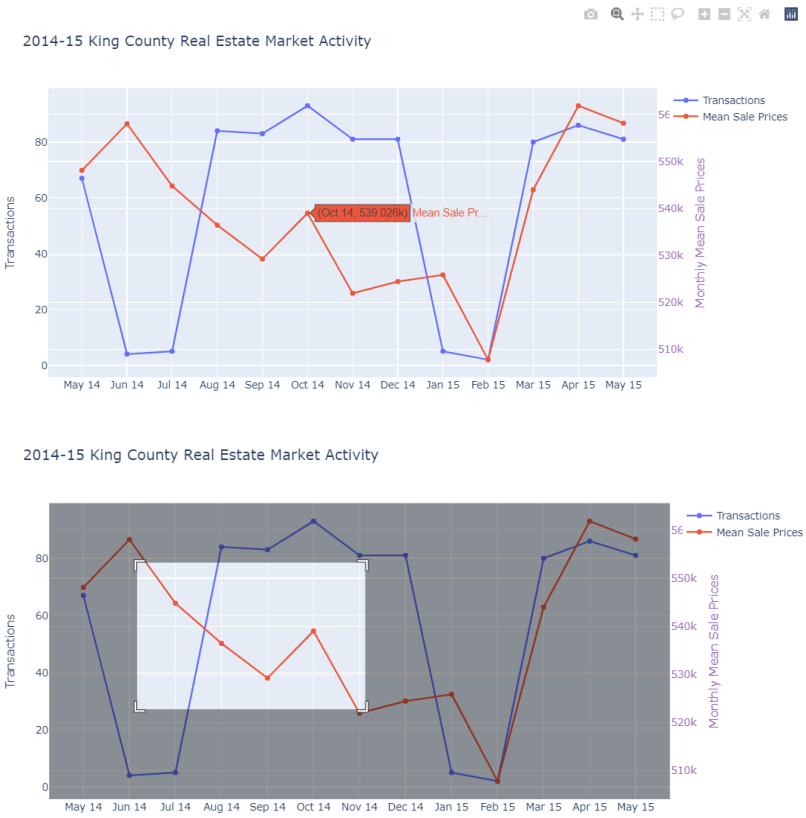
We decided to use a geographical map for the first visualization. So we created a map that displays Seattle and its different districts. Furthermore, the visualization includes markers for all the locations of sold houses. If you would put a marker for every single sold house in the overview of the map it would lead to confusion. That is the reason why we clustered markers that are near to each other or in the same district. You can see this in the numbers that pop up in the first impression of the map. When it comes to the interaction methods you can scroll in and see a specific district and the interaction here. Of course, you have the opportunity to scroll out. This function helps a user to see the whole map and all the interactions that happened over the years. If you click on a single marker you get information about the house that was sold.

The technique is excellent for the intentions of a real estate agent. You just take a short look at the visualization and you are informed about the prices in certain areas. Additionally, if you want to check certain streets you can scroll into this location and take a close look at houses nearby. The plot gives immediate information about the areas that are more expensive and where life is affordable for people with a normal standard of living. Real estate agents can also hover over the markers and browse through the properties to get an impression of the market.

If you compare this geographical map with the one of the first assignment you can clearly see that difference. We added a color scale and did not just use random colors for the areas. Furthermore, the user has now a legend where he can see the price range in which the different districts appear. The geodata that helped to implement the zip code boundaries is

an improvement too. The visualization had just circles to describe the areas. Now there are specific boundaries between them.
When it comes to the implementation the first step was to load all the libraries that were used in the Jupyter notebook. The next step was to load the dataset by using pandas. To get a quick overview of the data I used the command .head() and .shape. The first command displays the first five observations of the dataset and .shape shows you the number of observations and variables. The next line is a preprocessing step because it transforms the Zipcode to a string that it matches with the geodata of the zipcodes boundaries. The next step was to create the map by using the folium library. So we created the map and used the longitude and latitude of the City Seattle to focus the map on the right spot. Furthermore, we adjust the zoom that you get at first to get an overview of the whole city. Now we added the markers to the map. We used the loop command to add the markers of the houses one by one and added the information to all the real estate that was sold. The last step was to use the geodata of the city of Seattle to set and display the zip code boundaries. We set the zip code and price for the features that should be displayed. We added a legend of the color scale which gives information about the prices in certain districts.

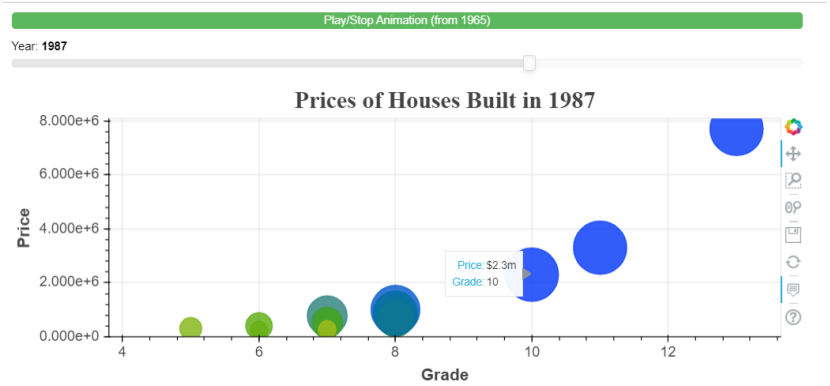
2. ***Visualization: Timeline of the number of transactions and mean sale prices over 2014-2015***



Since we are implementing a task that possesses time-series data such as the date of the sale's transaction and the mean sale price of a house's, one way of visualization that can facilitate the understanding of the occurrences to the interested target is to show specific information on a line plot. As this graph contains time series data in a 2 dimension space and uses lines as graphical elements, the most significant graphical property is orientation showing the fluctuations of the transactions through time.

In this graph, target users can initially view a full plot containing data for the whole year. By hovering over a certain point in the line plot, they can be able to see the exact date and month, as well as the total number of transactions on that date. Furthermore, users can select a specific region of the plot, and look closely to a more specific period (i.e a specific week) or a range of dates they can choose themselves.
This visual representation shows the number of transactions of the house sales during a period of one year (2014-2015). As it can be seen from the graph, the transactions are highly volatile and looking at the full plot, there can be concluded that during the winter season (December-January-February), the transactions incur a significant decline, indicating that in this period, people are not that interested to buy a new house. As the graph is magnified and a weekly period is taken into the analysis (i.e. 19-25 January 2015), there can be determined that during the weekends(24-25 January 2015) there is a considerably smaller number of transactions.
Considering the concept visualization that we did on the first assignment, and following its feedback, we made a meaningful improvement to this graph by dispersing the time frame where the inferences on the number of transactions on a period of one year, can be made more smoothly. Additionally, we added a double line graph, representing the monthly mean house sales prices, because we thought that it is crucial for a real estate agent to understand and confirm the relationship between the number of houses sold and their mean prices. Using this graph, real estate agents can infer that on times when there occur more transactions which indicate a greater demand for house sales, the prices undergo inflations.
The implementation of this visualization was done by using the python library, Plotly, a simple library to create interactive visualizations. For this double axis line plot, we have first initialized the variables that will be used the date of transaction, from which we count the number of transactions that occurred in that specific date by using group by function. After this, we created two traces indicating the two-dimensionality of the y-axis of this graph, and assign the x and y values for each of them. This was followed by creating the layout containing the aforementioned traces, specifying the x and y-axis titles, the graph title, as well as the legend.

3. *Visualization: Bubble plot of the relationship between grade of a house with price through years*



In this visualization, the categorical variable grade is the main variable of interest, which represents the quality of construction and design of the buildings starting from 1-13 from the lowest to the highest quality. For this graphical representation, the visualization technique used is a bubble plot. The graphical elements used in this case are points, and the graphical

properties that matter are color and size which represent the amount of sale price (yellow to purple and smaller to larger circle), as well as the position which shows the relationship between price and grade.

The visualization above can create an understanding of how the grade variable of the houses is related to price in different years that they have been built. It can be seen from the visual representation that as the years have passed especially from 1976-2015, the quality of the buildings has increased proportionally with the price. It can be concluded that the majority of the houses that have been built before 1976 have a considerably lower construction quality in comparison to the years after 1976.

Following the feedback of the first assignment, we have managed to create a better version of this graph, by firstly transitioning it from a scatter plot to a bubble plot containing more graphical elements, as well as adding a slider that manually changes the year that a house has been built and a button that creates an animation of the plot through years. The target can interact with this graph, by clicking on a specific circle to see the grade and price. Additionally, they can select a specific region of the plot, by using different selection forms to see more closely the relationship of the variables. By utilizing this graph, the real estate agents can infer a conclusion on how the construction quality of the houses has changed over the years, as well as draw inferences on the possible sale price a house can have considering its grade point.

It was somewhat challenging implementing this graph using Bokeh, a python library used for interactive visualizations with which you can create JavaScript-powered visualizations without writing any JavaScript. The first step to be implemented, after importing the necessary libraries, was creating variables necessary for the color and the size of the bubbles that would represent the price on a specific grade point as has been explained even in the above paragraph. Next, the variables that would be used in this graph have been grouped such as price, grade, and year built. The upcoming step is to create the bubble plot and assign the x and y variables, titles, labels as well as variables range. There is also added a hover tool which makes it possible to display information (price and grade) when hovering over a specific point in the plot. Next, a callback CustomJS function has been created in order to update the plot when changing the value of the year from the slider. Following this step, a slider has been initialized, specifying the maximum and minimum values the yr_built variable can take, the steps that it should undergo when sliding left or right, the initial value, the title, and the callback function that was created earlier. Lastly, another callback CustomJS function has been created, this time for the toggle button that creates an animated plot.