House affordability in Medellin, Colombia

Ricardo Arias (ID 30550971)

Lab #1 (Tam Vo)

June 7, 2021

Table of Contents

1. Introduction

Medellin is the second-largest city in Colombia, with 2.6 million inhabitants. The city is divided into 22 different areas called communes where people live. In Medellín, Colombia the segregation by the income level of the people throughout the territory is clear, unlike Melbourne, there you can easily identify which is the area inhabited by people with higher income and which by people with less, the objective of this project is to discover if this prior thought is true or if there is another unseen pattern.

To reach the goal mentioned before, we are going to use three different datasets, obtained from three different places. The first one is an open dataset provided by Properati (the Latin American real estate search site) that provides detailed information of properties around Latin America, like price, operation type, property type, and some other relevant attributes of a property. The second one is an open dataset of the local government of Medellin, Colombia that provides the shapes of the communes of the city and will help us to add one more unseen attribute of the last dataset, the commune where is located each property, and the third one is an open data set of the government of Colombia that provides the Multidimensional Poverty Index of each block of the city, that could also let us add an attribute of poverty to the Properati Dataset.

With this new dataset created, we would like to build a dashboard that could be helpful for different people to identify what is the relationship between poverty (measured by the Multidimensional Poverty Index), communes, house pricing.

The audience for this project is broad. One of the users of the dashboard could be the local government of Medellin, Colombia. They could use this information displayed in the dashboard to understand better the distribution of poverty around the city, also could be used to identify the real estate market in the city so they could be aware of new regulatory changes or tax payments, or some other things related to this market.

Other people could be interested in this dashboard and are people involved in the real estate market, like someone that wants to sell their home, or want to buy a new one, also real estate agents or an investor. All these people could use the dashboard to find some relevant information about the market. Like the approximate price of a property based on some information, the offer or availability of properties in the city.

The dashboard could be helpful to improve the knowledge of the people around a market that is not so open in my home city and answer some questions that could help everybody to understand better some unseen realities of Medellin, Colombia.

Questions

1. How is the availability of different type of properties in different suburbs of Medellin, Colombia?
2. How is the pricing of different type of properties in different suburbs of Medellin, Colombia?
3. What is the distribution of poverty in different suburbs of Medellin, Colombia?
4. How is the relationship between the price of the properties in Medellin, Colombia, and the poverty?
5. What is the proportion of properties located in poor and rich zones of Medellin, Colombia?

2. Design

The process to create the final dashboard, was the five-sheet design methodology, where we create five sheets with five parts each one to get the best design of the dashboard. As follows we are going to tell how this process to us was

Brainstorm

In this step we first identified all the possible attributes we wanted to show to the user from the data frame, then we found different possible ways to plot this information and just draw them as a brainstorm. The attributes that we wanted to plot are as follows

Features	Possible Plots
Latitude	Map
Longitude	Map
Price	Slider Filter Boxplot Density Plot Attribute of a dot in a Map Card of Average Price Histogram
Property Type	Pie Chart of properties by Type Bar chart of properties by Type Attribute of a dot in a Map Selection filter
Commune	Shape Map Selection filter Pie Chart of properties by Commune Bar chart of properties by Commune
Multidimensional Poverty Index and MPI Class	Slider Filter Boxplot Density Plot Attribute of a dot in a Map (Size or colour) Card of Average MPI Histogram
Date	Slider filter Selection Filter Pie Chart of properties by Month Bar chart of properties by Month Attribute of a dot in a Map

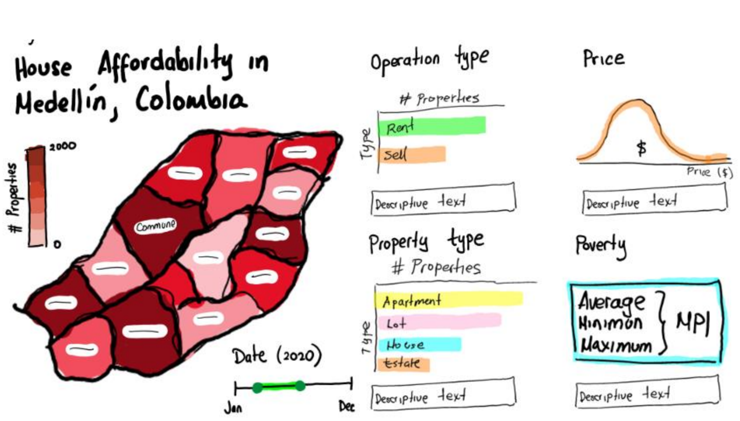
After having all the possible ways to plot the same attribute of the data frame, we must identify the best way to do it, that is why we identified the easiest and most effective ways to plot these attributes. That is why we selected 5 different plots that will be interactive with the filters too.

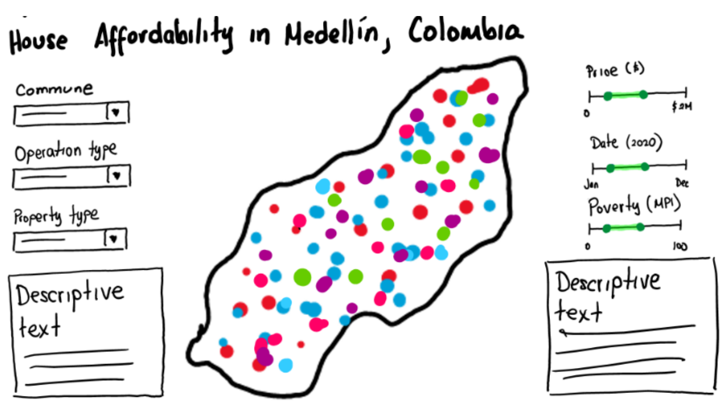
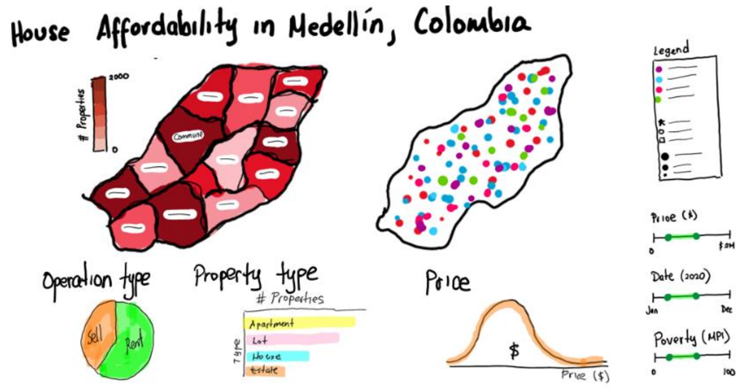
In each of these plots we select the attributes that could be plotted this way, and the idea in the sheet 2, 3 and 4 is to pick some of these options to display de information in the final dashboard. The ideas already filtered and refined are in the following table.

Possible Plots	Possible Plots
Map	Latitude – (Position) Longitude – (Position) Price – (Label or size) Property Type – (Label, shape, or colour) Operation Type – (Label, shape, or colour)
Density Plot	Price – (Divided in Rent and Sell) MPI
Box Plot	Price – (Divided in Rent and Sell) MPI
Pie Chart	Proportion of properties by: - Operation Type - Property Type - MPI Class - Communes
Bar Chart	Number of properties by: - Operation Type - Property Type - MPI Class - Communes
Filters	Date – (Date Range) MPI – (Slider Filter) Price – (Slider Filter) Operation Type – (Selection Filter) Property Type – (Selection Filter) MPI Class – (Selection Filter) Communes – (Selection Filter from the map)

Initial Designs

As we said before, having the final ideas for plots of each attribute, we build some possible dashboards that display all the information requires. Some of the ideas are shown as follows, the detail of each design is in the appendix.

Design 1	 House Affordability in Medellín, Colombia 2000 # Properties 0 Commune Date (2020) Jan Dec Operation type # Properties Type Rent Sell Descriptive text Price $ Price ($) Descriptive text Property type # Properties Type Apartment Lot House Estate Descriptive text Poverty Average Minimum Maximum MPI Descriptive text	In this design we selected the shape map with the information about each commune, two bar chart of operation and property type, one density plot of price and some statistical values of the MPI. From this dashboard we can observe all the global information about properties in Medellin, Colombia, but we do not have here detailed information about a specific property, as we did not plot each property. Even though is a good visualization it focuses in global information, and we miss too much small details that could be good to observe too, and also we only have one filter, so it is not too personalized.

Design 2	 House Affordability in Medellín, Colombia Commune Operation type Property type Descriptive text Price ($) 0 $2M Date (2020) Jan Dec Poverty (MPI) 0 100 Descriptive text	In this design we only chose to plot one big map with all the information embedded in each dot of the map. This map is the biggest and the most important plot of the dashboard. Now in this visualization we can observe all the details of each property, because each colour, shape, size and label can display all the information about one property. This visualization also has many filters, so the user can see the properties of their interest. This plot is the opposite of the design 1, here we can see every detail of each property, but we cannot see any global information, as the resume of number of properties or proportions of them. Therefore, we also miss relevant information of the real estate market in general
Design 3	 House Affordability in Medellín, Colombia 2000 # Properties 0 Commune Legend Price ($) 0 $2M Date (2020) Jan Dec Poverty (MPI) 0 100 Operation type Sell Rent Property type # Properties Type Apartment Lot House Estate Price $ Price ($)	Now that we build two different designs with different purposes, we wanted to build a dashboard that merges the purposes of both prior designs. So this one tries to show detailed information, as well as general information. That is why we included different charts of previous designs, so the user could observe anything that we wanted to see. Since detailed information about each property in the map until general information about the market, like price, property type, and operation type of properties in general. This dashboard also included, many different filters, that allow the user to personalize the dashboard as much as he wants, as these filters change the information across the dashboard.

In the appendix you can see more information about advantages disadvantages of each design, as well as the operations that can be done in each dashboard.

Final Design

The final selection is the design 3, as it was already said. It is the most complete design of all the three that we have, from here we can observe all the required information. The design implemented in R shiny may differ from this last design, because we found out some features of R shiny, that let us do some improvements to the final design. With Tab Panels we could plot more than one chart in the same position, so the user could observe more information. We also changed the position of the filters and added a few more, so the implementation of the design was a little bit better. But keeping the original idea. This implementation of the final design can be observed in the following section of the document.

3. Implementation

Using R as programming language, we used Shiny and some other libraries to build the final dashboard, the implementation process to reach the goal is described as follows:

The dashboard has two main parts, the ***front-end*** called ui and the ***back-end*** called server, in the front end the most important part is to design a dashboard that displays the information effectively but also in a pretty way. While in the server we store the data and create the plots that will be displayed in the dashboard (front-end).

Front-End (UI)

The first step in the creation of the front-end was to create the dashboard where the information was going to be displayed. To do that we used a library called ShinyDashboard, that helped us to divide the principal page in three different sections: Header, Side Bar and Body.

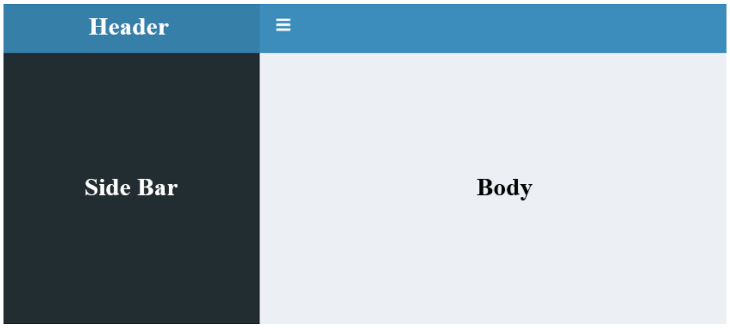


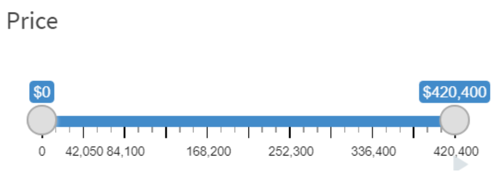
Figure 1. Dashboard distribution

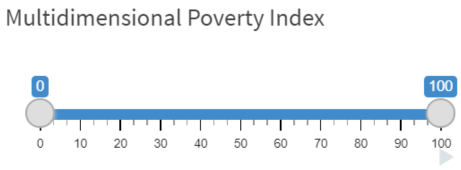
The header was used to show the title of the project, the side bar was used to display the filters, and the body to display the plots.

Filters

In this project we created three different types of filters

1. **Sliders:** This type of filters allowed us to filter information between two specific values, this type of filters was used for features like Price and Multidimensional Poverty Index as it can be seen as follows:



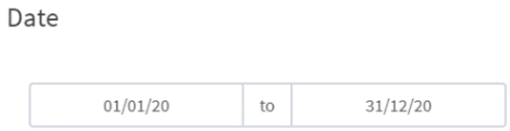


In both cases, the smallest value of the slider is the minimum value possible for each attribute, as well as the highest is the maximum value. This way we can display all the information stored when the filter is the whole possible domain of the slider filter.

The code used to plot them is:

```
sliderInput(name, label, min, max, value, step)
```

2. **Date:** All the properties have a published date, that is why we used this date range filter. To allow the user to display the information between some specific dates given by him. Again, the range includes the earliest and latest dates present in the data.



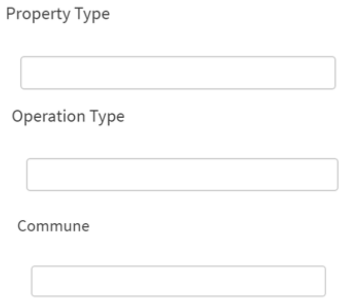
Date

01/01/20 to 31/12/20

The code used to plot them is:

```
dateRangeInput(name, label, start, end, min, max, format, separator)
```

3. **Select Input:** This filter allows the user to select an input from all the possible values of each feature, this filter was used was used for Property Type, Operation Type and Communes. Each filter has its own set of possible values that can be selected by the user to display in the plots.



The code used to plot them is:

```
selectInput(name, label, values, multiple)
```

As it was mentioned before, all these filters were displayed in the sidebar, each of them in a different box and in a different row so the design of the dashboard is better.

Plots

All the plots were displayed in the body of the dashboard, some of them were plotted in a box, as we did with the filters, but some others were plotted with Tab Panels, the reason of why we used this is because we saved space in the dashboard, and we can plot more than one figure in the same space of the dashboard. This way we can still display all the information required but not exceeding the space and not altering the aesthetic of the dashboard.

The first step to create the body of the dashboard, was to divide it in two different rows one on top of the other one. In the first row we displayed the map of Medellin, Colombia with all the properties, and the second row was divided in three different columns. In the first column we used a Tab Panel to display the proportion of properties by Operation Type, and by MPI Class. Both plots are pie charts. In the second column we used a Tab Panel too to plot the number of properties by property type and by communes both are horizontal bar charts. Finally in the third column also with a Tab panel we plotted the density plot of the price of Rent and Sell.

The dashboard has 7 different plots, even though only 4 can be seen at the same time, because the other three are hidden in the Tab Panels waiting to be displayed by the user.

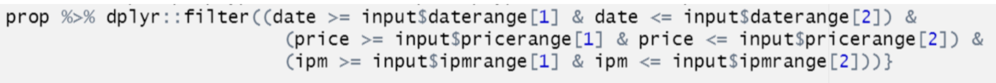
To display these plots in the body we used two different codes:

```
plotOutput(name of the plot in the server)
leafletPlot(name of the plot in the server)
```

It depends on the type of plot created in the server.

Back-End (Server)

In the back end we basically perform three different things, the first one is to import and store the data that will be displayed, then with this raw data we proceed to create new reactive data frames using the filters that are in the front end. As these filters can be change by the user in many ways, we need to get the input values in a reactive function and then filter the original data. To do that we used the function *reactive*. Inside this function we perform a regular dplyr filter but in this specific case the values that we are looking for are inputs of the different filters mentioned before. As follows we display an example:

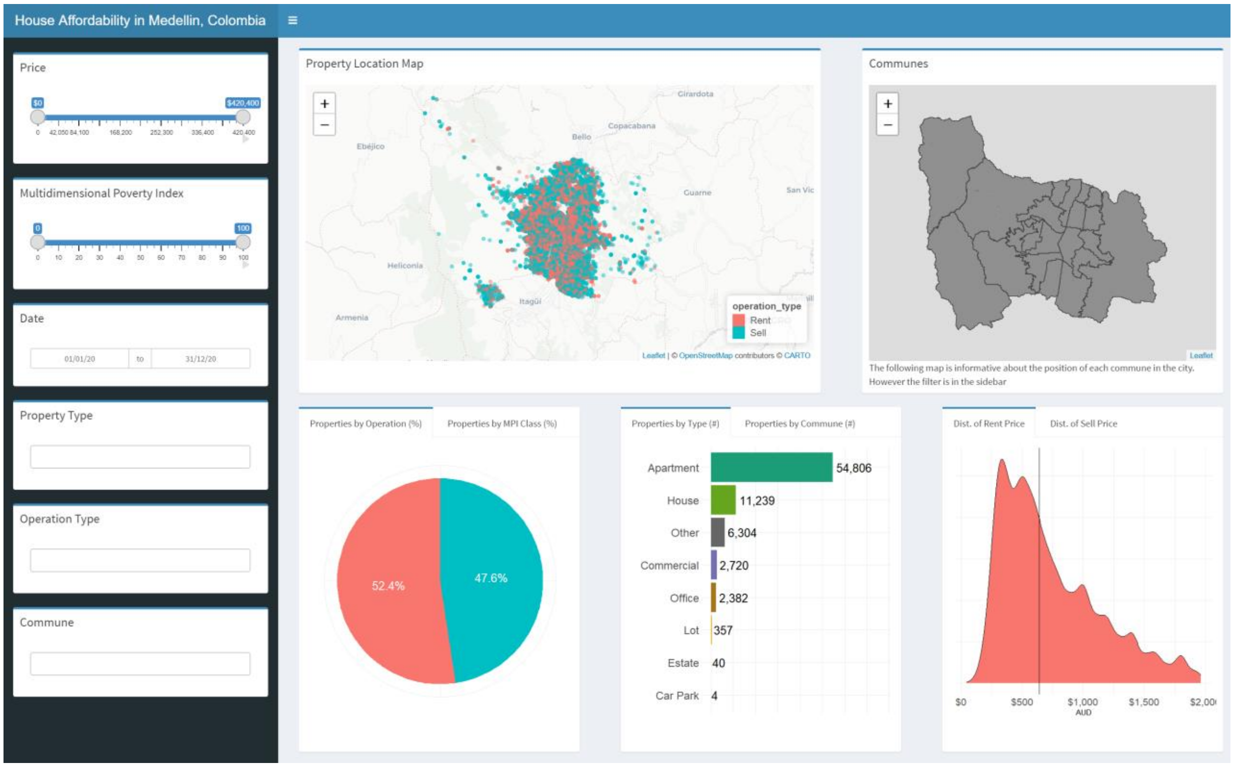


```
prop %>% dplyr::filter((date >= input$daterange[1] & date <= input$daterange[2]) &
                       (price >= input$pricerange[1] & price <= input$pricerange[2]) &
                       (ipm >= input$ipmrange[1] & ipm <= input$ipmrange[2]))}
```

Each **input$** statement gets the information of different filters, to then use this information to filter the original dataset and plot the requested information.

Finally, we proceed to plot the information as we usually do in R, but in this case, we are using the reactive data frame instead of the static one. This way the plot will change every time that each filter changes, making it an interactive dashboard.

Finally, implementing all the steps told before, this is the result:

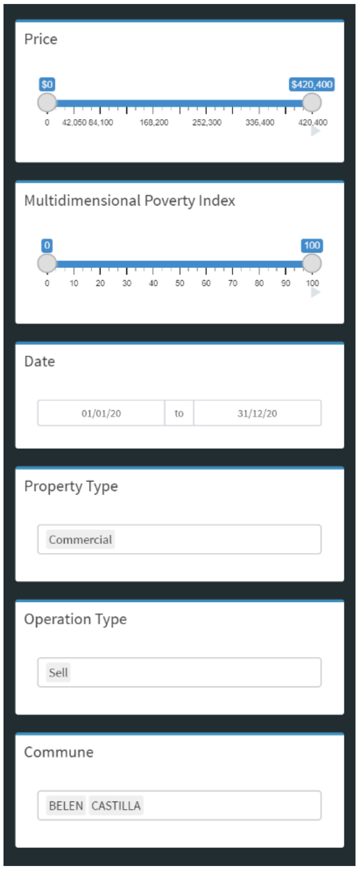


The following map is informative about the position of each commune in the city. However the filter is in the sidebar

4. User Guide

Controls

All the changes that a user wants to make to the dashboard, must be done from the Side Bar that is the place where all the filters are. From here the users can select the specific information they want as follows:



Price

$0 – $420,400

Multidimensional Poverty Index

0 – 100

Date

01/01/20 to 31/12/20

Property Type

Commercial

Operation Type

Sell

Commune

BELEN CASTILLA

Slider Inputs: From these two different filters (Price and MPI) we can set different ranges of information that we want to display in the dashboard. With the grey dots the user can change the minimum and the maximum value to be shown in the different plots. The left grey point sets the minimum value and the right one sets the maximum.

Date Range: From this filter we can select a window of time to be displayed. Selecting two dates (from and to).

Select Input: If the user clicks each rectangle of Property Type, Operation Type or Commune, a set of valid values will be displayed, from these lists the use**r can select the values that he wants to display in the plots. The user can also select** more than one value from each filter, as well as delete a selection.

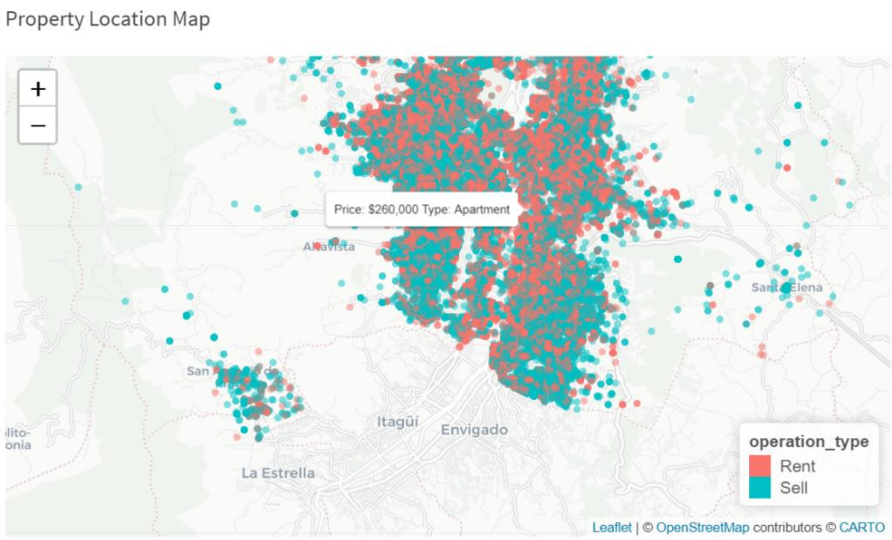
The possible selections for each filter are:

- **Property Type:**
  - "Apartment"
  - "Commercial"
  - "Car Park"
  - "Estate"
  - "House"
  - "Lot"
  - "Office"
  - "Other"

- **Operation Type:** "Rent" and "Sell"
- **Commune:**

5. "ARANJUEZ"
6. "BELEN"
7. "BUENOS AIRES"
8. "CASTILLA"
9. "DOCE DE OCTUBRE"
10. "EL POBLADO"
11. "GUAYABAL"
12. "LA AMERICA"
13. "LA CANDELARIA"
14. "LAURELES"
15. "MANRIQUE"
16. "PALMITAS"
17. "POPULAR"
18. "ALTAVISTA"
19. "ROBLEDO"
20. "SAN ANTONIO DE PRADO"
21. "SAN CRISTOBAL"
22. "SANTA ELENA"
23. "SAN JAVIER"
24. "VILLA HERMOSA"

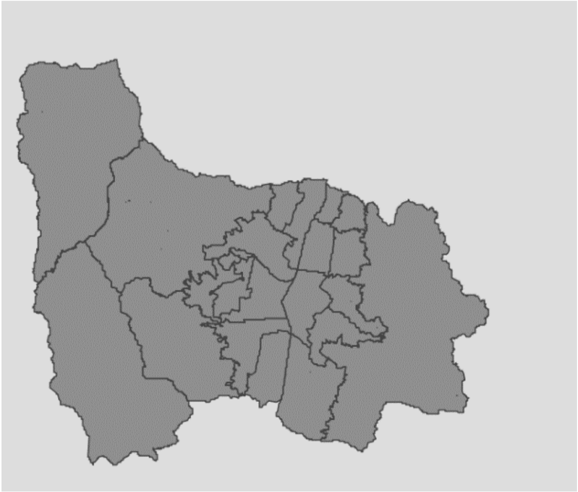
Plots



The map is the most important and the biggest plot of the dashboard, in this plot we can see a lot of information: the location of each property, the price, the type and the operation.

The location is given by the position of the dot in the map, the price and the type of the property can be seen by putting the mouse over the property and the operation type can be seen with the colour.

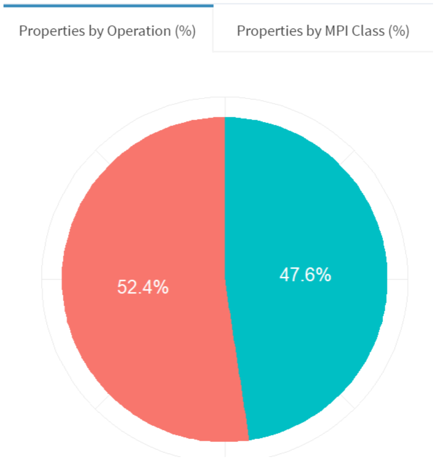
The user can zoom in or out the map by using the scroll of the mouse or the + and – present in the map

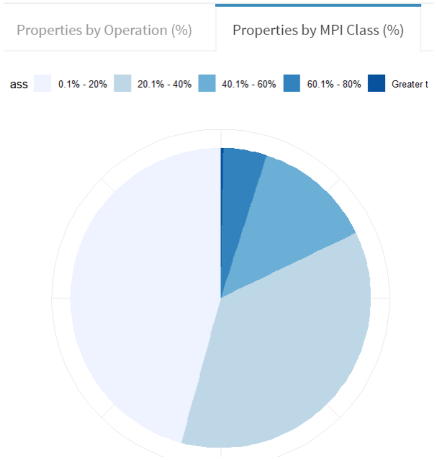


\

This plot gives the idea of the position of each commune, in the city. By putting the mouse over each shape, the map displays the name of the commune, so you can search for it in the filter that is the left sidebar. It could be interesting to use the map as a filter and shows interactive information with the change of the filters. Even though R Shiny, does not recognize the sf maps as data frames, so we could not perform reactive merges as we did with the other tables.

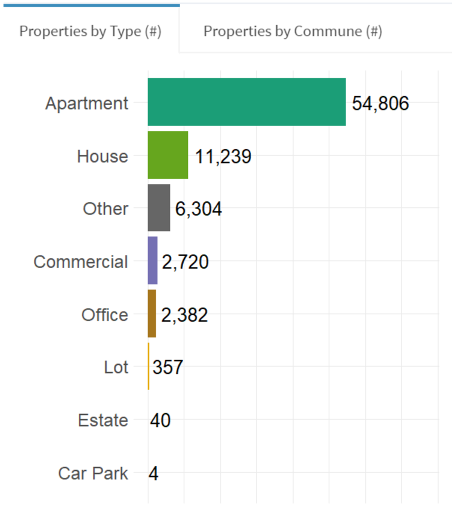
Even though the information displayed is not the expected, this map is helpful for the user, about the position of each suburb. That is why we keep it in the final design of the dashboard.

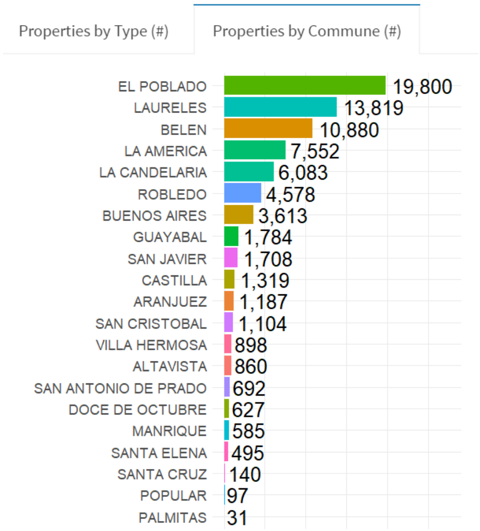




These two pie charts are in the left bottom corner of the dashboard, both are in the same Tab Panel and can be changed by clicking the Tab you are interested to see. The pie chart of Properties by Operation displays the proportion of properties that are available for Rent and Sell given the conditions selected in the filters. The colour of the plot follows the colours of the map shown above, being the red portion of the circle Rent and the blue portion of the circle Sell.

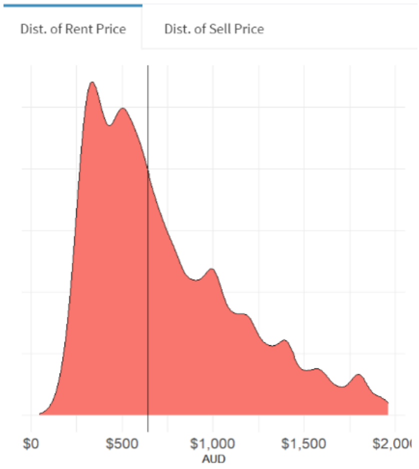
The pie chart of Properties by MPI Class displays the proportion of properties that are classified in different categories of Poverty following the MPI, the colour selected for this chart follows a scale of blues, being the lighter blue the richer zones and the dark blue the poorer zones. This chart also has the legend on top of the chart, for more information.

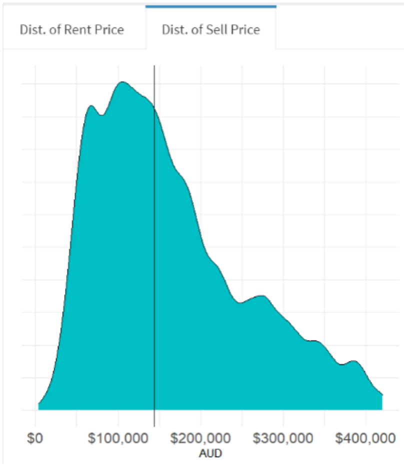




These two horizontal bar plots are in the centre bottom of the dashboard, both are in the same Tab Panel and can be changed by clicking the Tab you are interested to see. The bar chart of Properties by Type displays the number of properties by types of the property given the conditions selected in the filters.

The bar chart of Properties by Commune displays the same information displayed before, but in this case, we grouped the data in communes and not in type of the property. This way we could see the availability of the properties in two different ways.





These two density plots are in the right bottom corner of the dashboard, both are in the same Tab Panel and can be changed by clicking the Tab you are interested to see. The density plot Dist. Rent price shows the distribution of price of rent of the properties with the condition given.

The density plot Dist. Sell Price shows the same, but for the properties that are being sold, these two plots also follow the colours used for the other plots, red for Rent and blue for Sell. Also, these have the black line in the middle of the plot that shows the median price of the distribution. So, we can analyse the skewness of the density plot in each case.

The reason of why we created two different plots for each operation type is because the scale of price of sell and the scale of rent is very different as it can be seen in the plots above.

5. Conclusions

From the project in general and the visualization, we can answer the questions we asked at the beginning of the document, even though we could not answer one as well as was expected. From the visualization (Dashboard), we could identify the most relevant features of a property to determine its price. In this specific case, we explored the location of each property, the commune, the type of property, if it is rented or sold. But we found out in the exploration part of the project, that there was an attribute left that would have helped us to understand better the reality of the real estate market of Medellin, Colombia.

This attribute left was the area of the properties advertised. The reason why we need that is that we are required to standardize the price of the properties. After all, if not, we are comparing properties that are different between them. For example, we could be comparing one house of 1000 m2 in the poorest side of the city with a property of 80m2 in the richest part of the city, and we would think that just because the property is expensive is well located. But if we had the area of each property, we could compare the price of each property per square meter and that way we could demonstrate that maybe the price per square meter changes all our findings.

Another problem that we found in this project is that the data of poverty measured in the Multidimensional Poverty Index is unbalanced across the city. Almost 90% of the properties are in a zone of poverty between 0% and 20%, and just the other 10% are in the other 80% of the MPI. That is why we cannot get any conclusive statement regarding the poverty around the price of properties.

To improve this project, we would need another measurement of poverty, more accurate, and could show in a better way the reality of the city.

There were published approximately 78.000 properties in Medellin, Colombia last year, and almost 70% of those properties were apartments, also the 25% of the properties were published in one commune (El Poblado). With the dashboard we can get more insights like this using the filters so we can explore further the data following the requirements of each user. Each user can get its own conclusion based on the necessities.

6. References

DANE. (2021, 04 16). *Geoportal Dane*. Retrieved from https://geoportal.dane.gov.co/visipm/ : https://geoportal.dane.gov.co/visipm/

Medellin Goverment. (2021, 04 16). *GeoMedellin*. Retrieved from https://geomedellin-m-medellin.opendata.arcgis.com/: https://geomedellin-m-medellin.opendata.arcgis.com/datasets/l%C3%ADmite-catastral-de-comunas-y-corregimientos?page=2

Properati. (2021, 04 16). *Properati*. Retrieved from https://www.properati.com.ar/data/: https://www.properati.com.ar/data/

R Studio. (2021, 06 04). *R Shiny Dashboards*. Retrieved from https://rstudio.github.io: https://rstudio.github.io/shinydashboard/get_started.html

R Studio. (2021, 06 04). *R Studio*. Retrieved from https://shiny.rstudio.com: https://shiny.rstudio.com/gallery/

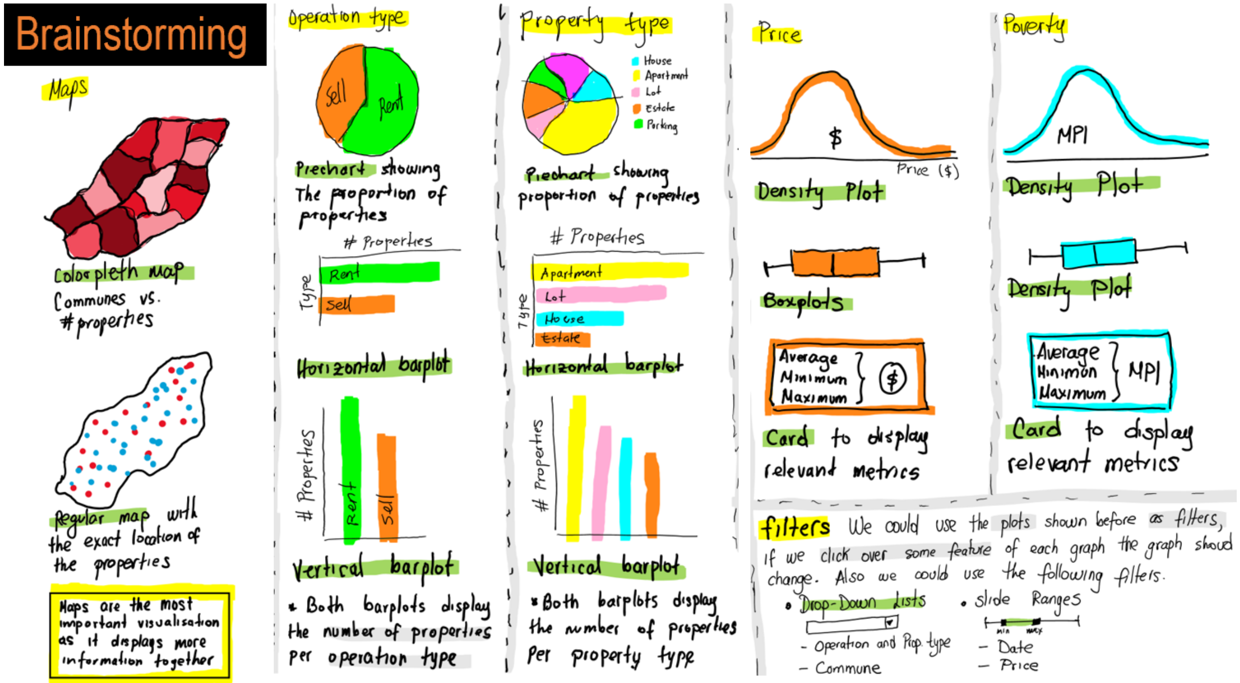
R Studio. (2021, 06 04). *R Studio*. Retrieved from https://rstudio.github.io: https://rstudio.github.io/leaflet/shapes.html

Santos, M. E., & Alkire, S. (2011). Training Material for Producing National Human Development Reports. *Oxford Poverty & Human Development Initiative*, 35.

Wikipedia. (2021, 04 06). *www.wikipedia.com*. Retrieved from Multidimensional Poverty Index: https://en.wikipedia.org/wiki/Multidimensional_Poverty_Index

Appendix: Five sheets design

Brainstorming


Maps
Coloropleth map
Communes vs. # properties
Regular map with the exact location of the properties
Maps are the most important visualisation as it displays more information together
Operation type
Sell
Rent
Piechart showing The proportion of properties
# Properties
Type
Rent
Sell
Horizontal barplot
# Properties
Rent
Sell
Vertical barplot
* Both barplots display the number of properties per operation type
Property type
House
Apartment
Lot
Estate
Parking
Piechart showing proportion of properties
# Properties
Type
Apartment
Lot
House
Estate
Horizontal barplot
# Properties
Vertical barplot
*Both barplots display the number of properties per property type
Price
$
Price ($)
Density Plot
Boxplots
Average
Minimum
Maximum
$
Card to display relevant metrics
Poverty
MPI
Density Plot
Density Plot
Average
Minimum
Maximum
MPI
Card to display relevant metrics
filters We could use the plots shown before as filters, if we click over some feature of each graph the graph should change. Also we could use the following filters.
• Drop-Down Lists
- Operation and Prop type
- Commune
• Slide Ranges
min
max
- Date
- Price

Combine/Refine

- There are some plots that show data to broad like choropleth map, pie charts, and barplots. From here we can get the resume data for property type, operation type and communes. But we would be ignoring the particular data of each property.
- Plots like the regular, can show the user detailed data for each property. We could use a mix of these plots show to build the best dashboard possible

Title:	House Affordability in Medellin, Colombia
Author:	Ricardo Arias Salazar (*ID: 30550971*)
Date:	May 16, 2021
Sheet:	1 of 3

Filter

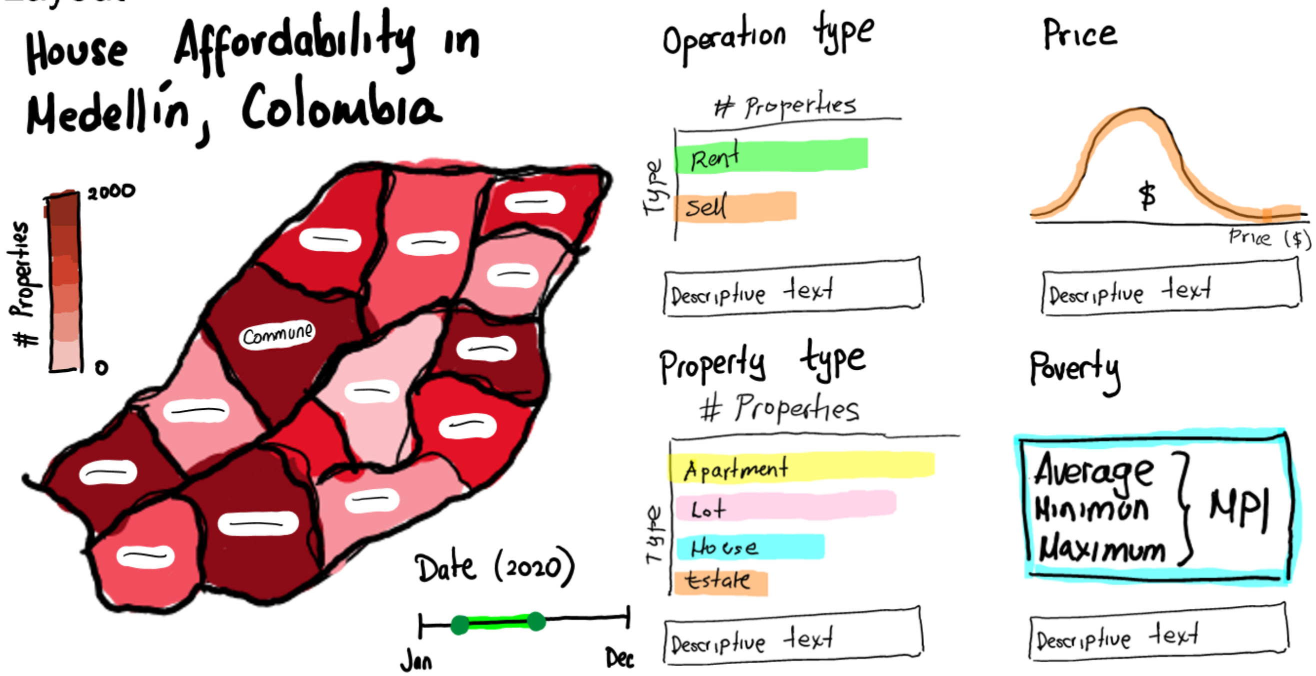
- Due the unbalance data of poverty we can not use boxplots or density plots because we would not show anything
- We could use filter for all the features, therefore we can personalise the visualization as much as we want.
-

Categorize

- Categorical Values: Property type, operation type, communes.
  - For these features we can use bar plots or pie charts
- Continuous Values: Price, IPM, Latitude and Longitude
  - For these features we can use densitiplots, box plots, maps

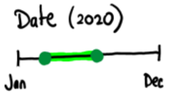
Title:	House Affordability in Medellin, Colombia
Author:	Ricardo Arias Salazar (*ID: 30550971*)
Date:	May 16, 2021
Sheet:	1 of 3

Layout

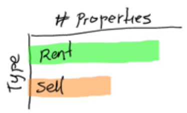


Focus/Zoom

Choropleth Map
displaying each shape filled in function of the quantity of properties per commune



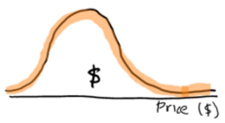
Range Slider Filter
With this filter we can select the months we are willing to display in all the graphs



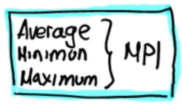
Horizontal Barplot
displaying the quantity of properties listed under different Operation Type

# Properties
Type
Apartment
Lot
House
Estate

Horizontal Barplot
displaying the quantity of properties listed under different Property Type



Density Plot
displaying the price of the properties listed



Card
displaying the most relevant metrics related to the poverty measured with MPI

Operations

- Sliding range of the date
- Filtering Operation Type by clicking bars
- Filtering Property Type by clicking bars
- Filtering Communes by map shapes

Advantages

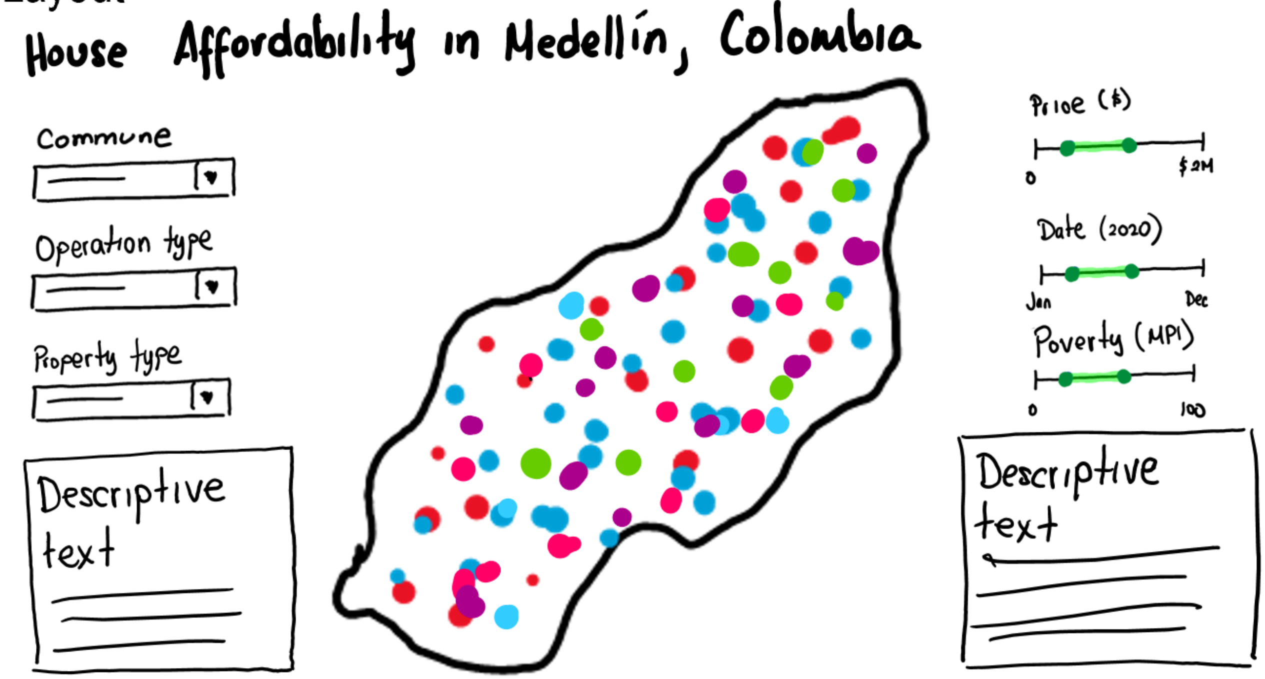
- Simplicity
- Interactive data

Disadvantages

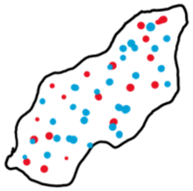
- Not displaying all the information given (Latitude and Longitude left)

Title:	House Affordability in Medellin, Colombia
Author:	Ricardo Arias Salazar (*ID: 30550971*)
Date:	May 16, 2021
Sheet:	2 of 3

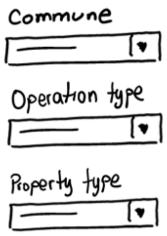
Layout



Focus/Zoom



Map
Displaying the following information:
- Position: Latitude and Longitude
- Shape: Operation Type
- Color: Property type
- Size: Price



Drop-down Filter
From this filter we can select one or more values of the filters shown

Price ($)
0
$2M
Date (2020)
Jan
Dec
Poverty (MPI)
0
100

Range Slide Filter
From this filter we can select the range of certain value that we want to display in the map

Operations

- Sliding range of the date, Price and Proverty (MPI)
- Drop-Down Filtering of Operation Type, Property Type and Commune

Advantages

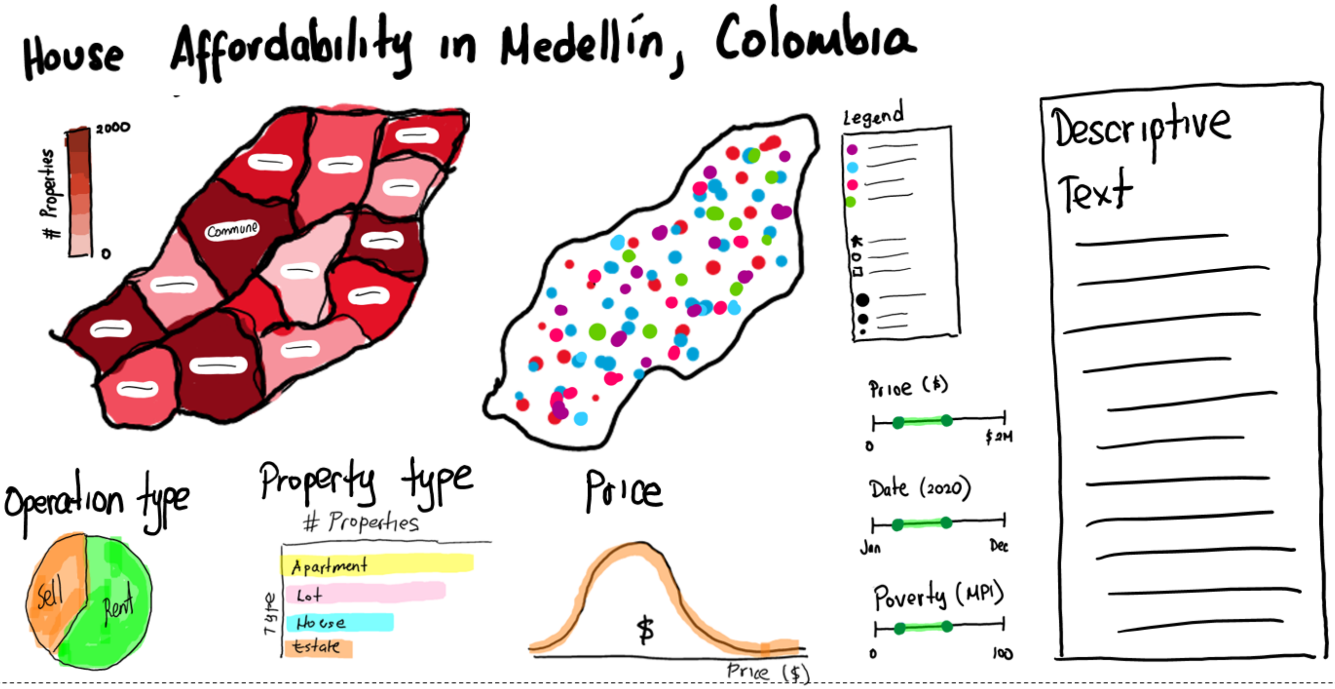
- Even simpler than sheet 1
- Customization of the visualization by using filters

Disadvantages

- It could be tedious to move filters each time to display the information we want.
- Lack of visualizations to understand better the insights

Title:	House Affordability in Medellin, Colombia
Author:	Ricardo Arias Salazar (*ID: 30550971*)
Date:	May 16, 2021
Sheet:	3 of 3

Layout



Focus/Zoom

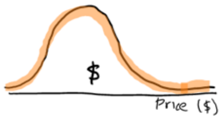
Choropleth Map
displaying each shape filled in function of the quantity of properties per commune

Map
- Position: Latitude & Longitude
- Shape: Operation Type
- Color: Property type
- Size: Price

Pie Chart
displaying the proportion of properties listed under different Operation Type

# Properties
Apartment
Lot
House
Estate
Type

Horizontal Barplot
displaying the quantity of properties listed under different Property Type



Density Plot
displaying the price of the properties listed

Price ($)
0
$2M
Date (2020)
Jan
Dec
Poverty (MPI)
0
100

Range Slider Filter
With this filter we can select the months, the price and the MPI that we want to display in all the graphs

Operations

- Sliding range of the Date and Poverty
- Filtering Operation Type by clicking Pie
- Filtering Property Type by clicking bars
- Filtering Communes by map shapes

Advantages

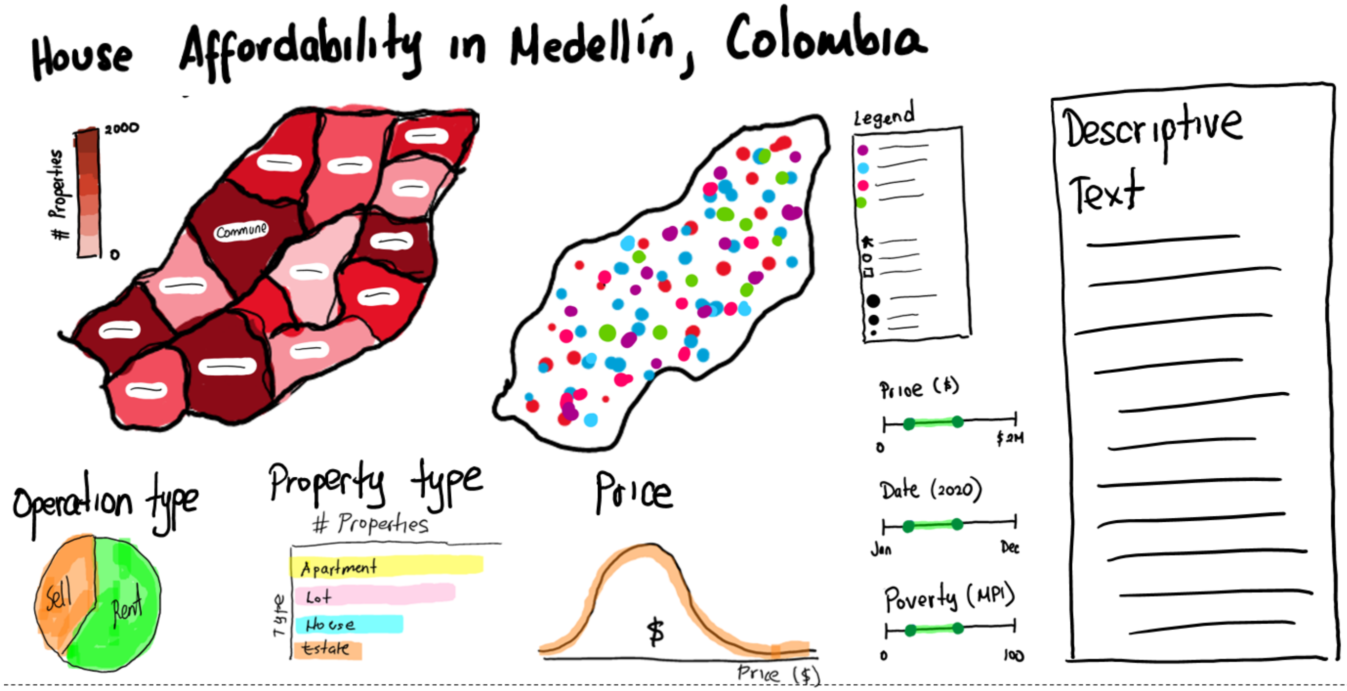
- Display all the information given
- Interactive data

Disadvantages

- Complexity

Title:	House Affordability in Medellin, Colombia
Author:	Ricardo Arias Salazar (*ID: 30550971*)
Date:	May 16, 2021
Sheet:	Final Design

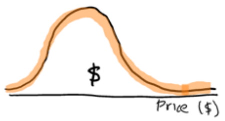
Layout



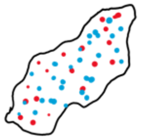
Focus/Zoom

Choropleth Map
displaying each shape filled in function of the quantity of properties per commune

Pie Chart
displaying the proportion of properties listed under different Operation Type



Density Plot
displaying the price of the properties listed



Map
- Position: Latitude & Longitude
- Shape: Operation Type
- Color: Property type
- Size: Price

# Properties
Type
Apartment
Lot
House
Estate

Horizontal Barplot
displaying the quantity of properties listed under different Property Type

Price ($)
0
$2M
Date (2020)
Jan
Dec
Poverty (MPI)
0
100

Range Slider Filter
With this filter we can select the months, the price and the MPI that we want to display in all the graphs

Operations

- Sliding range of the Date and Poverty
- Filtering Operation Type by clicking Pie
- Filtering Property Type by clicking bars
- Filtering Communes by map shapes

Details

To create this final design we will use R Shiny. Using R we can wrangle the data and using ggplot and leaflet we can plot it. Then using R shiny we can create interactive designs.

The time of development of the project is 2 weeks, using the following libraries of R Studio: dplyr, shiny, leaflet, sf, ggplot