# Report Data Visualization: Airbnb Discoverer

Discover data from Airbnb. Visually. Dynamically. Simple.

Jan Lennartz & Hornella Fakem Fosso

## Project Structure

### Algorithms

The project is structured into the following files:

- **ui.R**: User interface logic
- **server.R**: Server logic (calculations)
- **global.R**: Globally available variables
- **preprocessing.R**: Used for manual preprocessing of calendar data, only needed if new data is imported
- **read_data.R**: Methods to read the input files

### Data

All data is contained in the Data folder. It contains four subfolders:

- **Calendar**: Reservations for one year per listing for the given city
- **Listings**: The listings data for one city, e.g. price, host, location, etc.
- **Neighbourhoods**: Neighbourhoods data as geoJSON files. Boundaries of all neighbourhoods in the city
- **POIs**: Data for the points of interest

The Calendar folder and POIs folder each contain a subfolder **raw**. Here are the raw calendar data sets (unpacked) and raw POI data sets, respectively. As it takes a while to process them, each of them has to be preprocessed in order to avoid long waiting times. This can be done within the **preprocessing.R** file with the methods *preprocess_calendar* and *preprocess_pois*.

If you want to add another city you can simply download the files: **calendar.csv.gz**, **listings.csv**, **neighbourhoods.geojson** for a city from the source below and rename them to the cityname. Then you copy them to the appropriate folders and run the preprocessing.

To add the files for the POI to this app, the appropriate file has to be downloaded (based on the country or region). The dowloaded file needs to be extracted and the corresponding csv file (in a subfolder of the extracted folder) needs to be compressed into a .gz format. With the naming convention: nameofthecity-pois.osm.csv.gz. Note: During the preprocessing the input data gets reduced to only the points present in a circle around the city center. This cuts the data size down by a lot depending on how big the initial region of the file was. This file needs to go into the folder Data/POIs/raw and then the preprocessing for this city needs to be done. Analogue to the preprocessing of the calendar data.

The last step is to add the city's name to the *supportedCities* list in **global.R**.

# App Structure

## Data

To allow for this analysis we made use of two data sources, each delivering multiple input files. In the following the reader gets and overview of these data sources.

### Airbnb

This app allows to interactively discover data from Airbnb. The available data is from the year 2019/2020. The data source is: *http://insideairbnb.com/get-the-data.html*. There were three files used for each city: listings.csv, calendar.csv.gz and neighbourhoods.geojson. The corresponding calendar data is available for one year in advance (i.e. 2021).

### POIs

Additionally, there is a second source of data for the points of interests. It is scraped from Open Street Maps (*https://www.openstreetmap.org*) and converted to csv files that are available here: *http://download.slipo.eu/results/osm-to-csv/europe/*. The description of the data can be found here: *http://www.slipo.eu/?p=1551*.

### Limitations

It can be easily extended to more cites. The only limitation is your RAM. The only version of this app (*https://madnex.shinyapps.io/Airbnb/*) is limited to 1GB RAM. This is why some big cities would by themselves demand most of the RAM here and only some "minor" cities are shown.

## Pages

On top of each page there is a header which allows to switch dynamically between the available cities. Every calculation on every page is therefore easily applied to each city and it is possible to compare the available cities with each other.

### Info Page

An overview over the app and where to find what.

### Data Page

Here can be found descriptive statistics visually prepared for the reader's eyes. It is subdivided into six tabs:

- Summary: A simple summary output to get an overview of each variable in the listings data.
- Details: Depending on the selected data set (Calendar, Listings or POIs) this data set is ready to be explored in a dynamic data table.
- Descriptive Stats: Three plots can be found on this tab. The first plot is a horizontal bar plot of the variables "Room Type" or Neighbourhood. The second plot is a histogram of the variable "Price", "Minimum Nights", "Number of Listings" or "Number of Reviews". Additionally, one can adjust the number of bins and a cutoff point. The cutoff allows to exclude outliers and look only at the data points below the q-th quantile. The last plot is a scatter plot which allows for the same selection of

variables as the histogram and as well offers a cutoff point. This time the cutoff can be separately adjusted for the x variable and y variable.
- Calendar: The calendar allows to select a host, which is pre-filtered to have between 3-10 listings, so that the plot will be visually appropriate. For this host then, the availability of all his listings will be shown in a gantt diagram.
- Neighbourhoods: On this tab two plots can be found. The first is a pi chart which shows the proportion of number of listings by neighourhood for the top 15 neighbourhoods (based on the number of listings). The second plot is a bar plot of the average price per neighbourhood. Both plots use the same color scheme.
- Hosts: Here, a bar plot of the number of listings per host can be seen. Additionally, a boxplot is shown where the host can be selected and it will show the distribution of the prices of the listings owned by this host.

**Explore Page**

On this page it is possible to explore the data visually on a map and filter it by some extend. The map shows all filtered listings and the neighbourhoods. The listings are colored according to their price and the neighbourhoods are colored by either the number of listings per neighbourhood, the average price of the listings per neighbourhood or the number of reviews of all listings in a neighbourhood. This is selectable by the user. These statistics of the neighbourhoods are not influenced by the filtering on the listings, meaning that the neighbourhood colors are always calculated based on all listings.

It is also possible to select a host and have a look at all the listings on the map that belong to this host. The filtering allows to filter by price and/or by the minimum nights. The price filter removes all listings that have a higher price than the selected value. The nights filter removes all listings that have less minimum nights than the selected value. Therefore, one can choose the nights one wants to spend at the very minimum and check which listings offer stays of that length or more. For both inputs a custom slider is used. It has a range of the corresponding values from the minimal value to the 90%-quantile of the values. This is because otherwise outliers would drastically skew this slider and make it inpractical to select the appropriate filter. However, to ensure that the user can select all listings, the extreme values are added to the slider as well but without a selection range towards them, i.e. from the value 100 to 1000 it might be just one step. Additionally, one can choose which types of listings should be shown: entire homes/apartements, hotel rooms, private rooms and/or shared rooms.

As an extension it is possible to show points of interest ($POI$) on the map. If selected the POI's are shown in a clustered way since there are too many of them to display all at once. In fact they are so many, that only a category of them is shown at a time. This can be chosen from a radio button list. However, if the user selects a subcategory, the POI's are displayed directly without any clustering, as now the number of them is expected to be moderate. It is possible to select multiple subcategories at once. The selection and filtering input elements for the category and subcategory, respectively are hidden until the user selects to show the POIs to avoid confusion and allow for a clean interface.

Clicking on a listing reveals a pop-up which has information about its name, number of reviews, price, minimum nights, room type, host and a link to the listing on the Airbnb website. Furthermore, if a listing is selected on the map, a gantt chart is shown which displays the listing's availability for the next 12 months. However, if the listing is not available at all, the user gets to see a message explaining this to him. Clicking on a POI shows a pop-up with information about its name, category, subcategory and a link to Open Street Maps of the node. Whenever a listing or a POI is selected on the map there is a message shown under the map revealing the id of the selected listing or POI. When the user clicks on the area of a neighbourhood the name and the number of listings in this neighbourhood are displayed in a pop-up.

**Linear Model Page**

This page allows interactively conduct a linear model for the target variable price. The variables can be selected dynamically out of the variables: neighbourhood, latitude, longitude, room type, number of reviews

and availability. The output of a summary call of the corresponding linear model is then shown and the user can check how well the model is fitted.