1 Google web scraper

While doing research to find datasets with regards to food inspections we decided to gather more information from webpages. The reviews from customers, written on google places, could give us information about how individual persons have experienced their visit to the specific food shop. In order to gather this information from the internet we set up a web scraper. We used the package *RSelenium* to execute an automated google search for all the food shops. This included a lot of trial and error due to unforeseen changes of the xPaths and other errors. A very fast but costly alternative would have been to use the Google Places API which provides access to all information stored by Google Places. Although it takes much more time we decided to go ahead with scraping the data ourselves. In terms of time a lot of patience was needed. Most of the errors only showed up in the middle of the scraping process, which had a duration of about 24 hours for all the food shops. Unfortunately most of the errors occurred towards the end of the scraping. Therefore, it was only possible to adjust the function after the script run during the night. This resulted in a lot of waiting and adjusting but finally the data was scraped successfully. Nonetheless the scraped data was not in a useable format yet. After tidying and joining the data to the original dataset it was ready to be used.

A brief analysis of the dataset revealed the incompleteness of the ratings. Approximately 40% of the shops don't have an entry in google places and thus neither rating. After a lot of discussion on how to handle the problem we agreed on using the number of reviews as a parameter of the internet popularity.

2 Eliminating duplicates and selecting shop chains

The data published by the state of New York includes not only a list of all the food ratings of different shops but also part of the history of the inspection grade development. This means the same shops could appear several times with different issues. To avoid the problem of having the identical shop more than one time we ordered the dataset ascending to the inspection date. Considering only the newest entry of every shop would have led to a bias regarding the hygiene grade because the bad graded shops improved themselves. Thus we decided to only take in account the oldest entry of every shop. This resulted in a data frame of around 7500 unique shops in the city of New York.
Additionally, the data set reveals not only the shops trade name but also its owner. This enabled us to identify if the shop is part of a chain. We added this parameter beside to the number of shops which belong to the specific chain. It seems possible to have an impact on the hygiene grade if the shop is part of a bigger chain.

3 Airbnb data

In order to acquire additional data and parameters on the location of the shops, we integrated data from Airbnb (average price and number of rooms). To join the original and the airbnb data frames a matching key is needed. Regarding the dataset only longitude and latitude were available. Therefore we used the above mentioned spatial data to assign the respective parameter to the ZIP codes in New York. Thereafter we grouped the data by ZIP code, calculated the respective means and counts and added it to the new tibble.

4 Random Forest Analysis

Due to the above mentioned imbalances in the dataset we used over- and under sampling to have evenly distributed data to train the random forest model.