Beatrice Tomasello
03.13.2020

# LONDON FOOD SCENE


MIBE
Master in International Business & Entrepreneurship

LSU

# London Restaurant Scene Report

Beatrice Tomasello

12/3/2020

#1. Introduction

As part of the Mibe x LSU challenge, this analysis has been carried out using R and the datasets provided. The analysis subject is the London restaurant scene, defined by variables such as rating, location, prices of the items in the menu and delivery times. As asked in the challenge, the analysis goes on around some business related questions provided in the challenge, plus a section of open analysis that I've chosen to conduct around the restaurant names, the Londoners food preferences and the location of the restaurants.

#2.Analysis

Let's start from loading the libraries we need for our project...

```
library(here)
library(magrittr)
library(tidyverse)
library(purrr)
library(dplyr)
library(purrrlyr)
library(ggplot2)
library(tidyr)
library(formattable)
library(rlist)
library(gtools)
library(tm)
library(SnowballC)
library(wordcloud)
library(RColorBrewer)
library(sf)
library(tmap)
library(tmaptools)
library(rgdal)
library(rgeos)
library(ggmap)
```

...and the datasets we are going to work with.

```
info_rest <- read_rds(here('data', 'resturants-mibe.rds'))
delivery <- read_rds(here('data', 'delivery-mibe.rds'))
```

Let's have a look at how the restaurants dataset is structured:

```
glimpse(info_rest)
```

```
## Observations: 5,786
## Variables: 7
## $ restaurant_id        <dbl> 191295, 54515, 113653, 184167, 84922, 194571, ...
```

```
## $ rest_name            <chr> "Baba Wali Hendon Broadway", "Burger & Lobster...
## $ rest_brand           <chr> NA, "Burger & Lobster", NA, "Europa 2 Go Pizza...
## $ rest_postcode        <chr> "NW97DY", "W1W7JE", "HA90TG", "SE255QF", "SW15...
## $ rest_neighborhood    <chr> "Hendon", "Fitzrovia", "Wembley", "Croydon", "...
## $ rest_rating          <dbl> NA, 4.7, NA, 3.8, 4.3, 4.4, 4.6, 4.2, 4.8, 4.6...
## $ rest_menu_item_price <list> [<2.80, 4.20, 5.60, 4.20, 5.60, 14.00, 16.80,...
```

```
head(info_rest)
```

```
## # A tibble: 6 x 7
##   restaurant_id rest_name rest_brand rest_postcode rest_neighborho~ rest_rating
##           <dbl> <chr>     <chr>      <chr>         <chr>                 <dbl>
## 1        191295 Baba Wal~ <NA>       NW97DY        Hendon                 NA
## 2         54515 Burger &~ Burger & ~ W1W7JE        Fitzrovia               4.7
## 3        113653 Afta Eats <NA>       HA90TG        Wembley                NA
## 4        184167 Europa 2~ Europa 2 ~ SE255QF       Croydon                 3.8
## 5         84922 Julia Do~ <NA>       SW151JP       Putney                  4.3
## 6        194571 Kin + De~ <NA>       E146AB        Canary Wharf            4.4
## # ... with 1 more variable: rest_menu_item_price <list>
```

#2.1 Restaurant Information Analysis

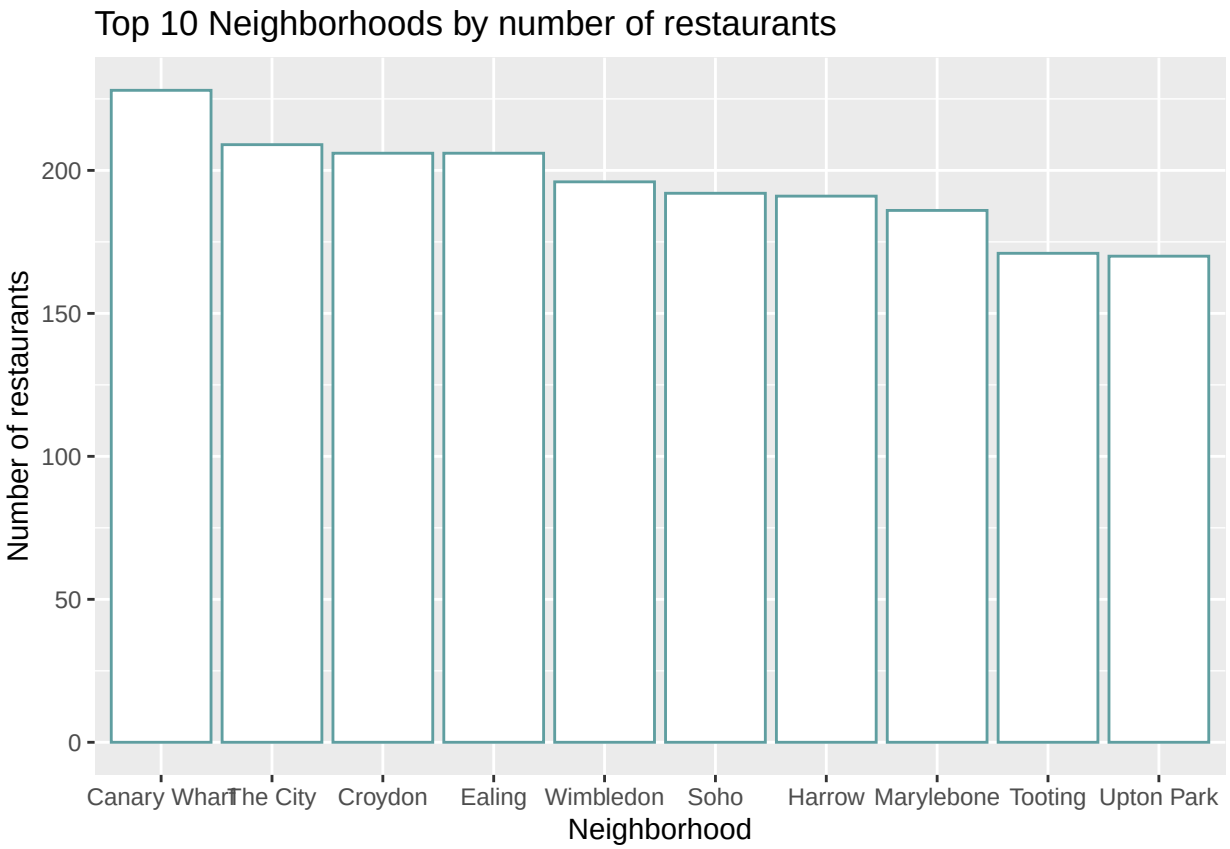
In this section we are going to work with the most relevant features in our first dataset to gain a better understanding of the London food scene for what concerns quality, location, presence of chains, and menu features of its restaurants.

*2.1.1 Top 10 neighborhoods by number of restaurants*

Using the grouping function we are able to lead our analysis neighborhood-wise and count the number of restaurants in each one, finding our top 10.

```
top10_neighb <- info_rest%>%
  group_by(rest_neighborhood)%>%
  filter(!is.na(rest_neighborhood))%>%
  summarise(rest_number = n())%>%
  arrange(-rest_number)%>%
  slice(1:10)

top10_neighb$rest_number <- as.numeric(top10_neighb$rest_number)

top10_neighb%>%ggplot(aes(reorder(rest_neighborhood, -rest_number), rest_number)) +
  geom_bar(col = 'cadetblue', fill='white', stat='identity') +
  labs(x= 'Neighborhood', y = 'Number of restaurants') +
  ggtitle('Top 10 Neighborhoods by number of restaurants') +
  ggsave('top10number.png')
```

```
## Saving 6.5 x 4.5 in image
```




Canary Wharf and The City, the two CBDs (central business districts) of London, retain the first two places of the chart. These areas have a high density of offices and this is a feature that tipically influences the number of restaurants thanks to the high affluency during daytime, especially at lunch. Croydon, Ealing and Wimbledon are instead main residential areas.

*2.1.2 Top 10 neighborhoods by restaurant review score*

This time grouping by neighborhood I find specific information on the rating based on the location, then calculating the average review I get an idea of the areas with the highest density of restaurants that satisfy the customers.

```
top10_rating <- info_rest%>%
  group_by(rest_neighborhood)%>%
  summarise_at(vars(rest_rating), funs(mean(., na.rm=TRUE)))%>%
  arrange(-rest_rating)%>%
  slice(1:10)
```

```
## Warning: funs() is soft deprecated as of dplyr 0.8.0
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
```

```
## This warning is displayed once per session.
```

```
top10_rating%>%ggplot(aes(reorder(rest_neighborhood, - rest_rating), rest_rating )) +
  geom_bar(col = 'cadetblue', fill='blue', stat='identity') +
  labs(x= 'Neighborhood', y= 'Average rating') +
  ggtitle('10 Neighborhoods with the best restaurant rating') +
  ggsave('top10rating.png')
```

```
## Saving 6.5 x 4.5 in image
```


We can easily observe that we have just three common names between the top 10 neighborhoods for number of restaurants and the top 10 for rating: just Soho, Marylebone and The City. It looks like quantity doesn't mean quality!

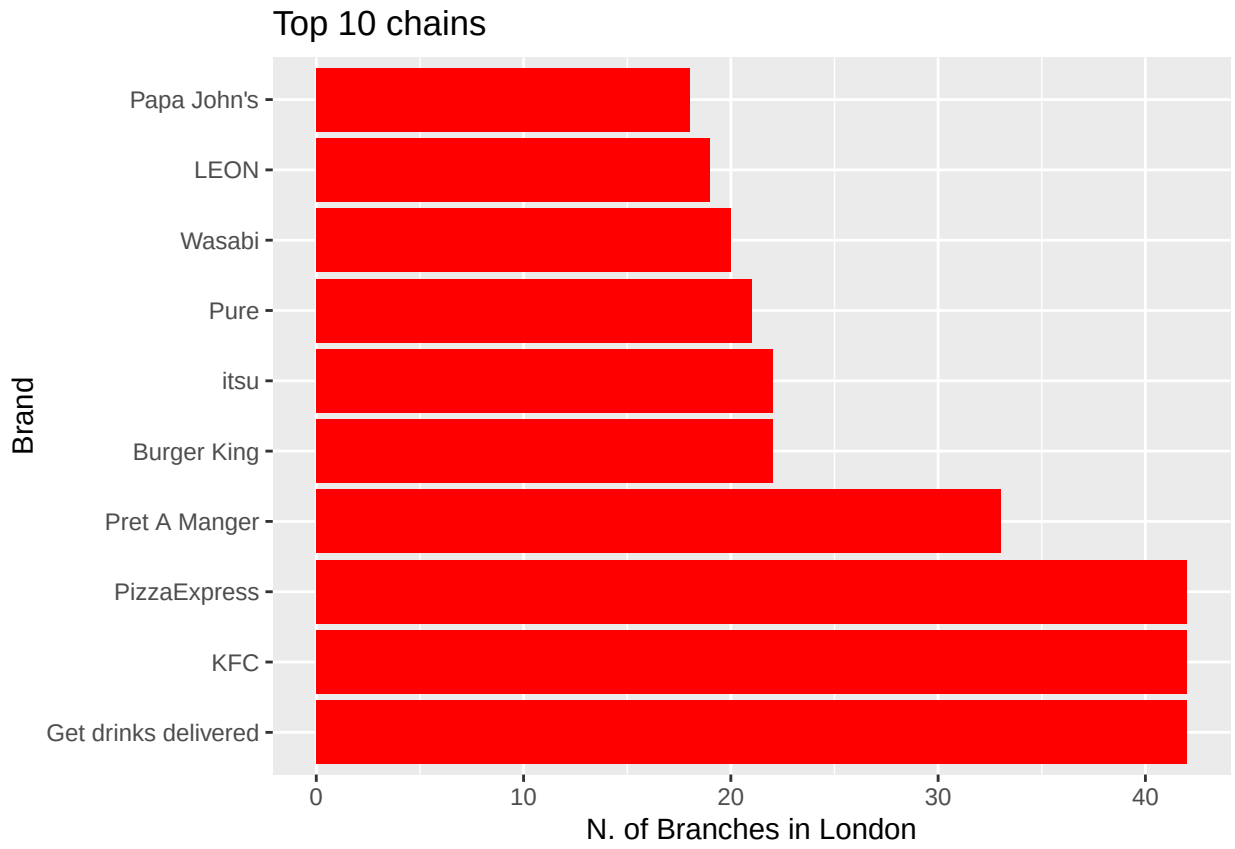
*2.1.3 Top 10 chains*

Let's now define which are the restaurant chains that have branches in London, and the ten chains that have the highest number of branches around the city.

```
chains <- info_rest%>%
  group_by(rest_brand)%>%
  filter(!is.na(rest_brand))%>%
  summarise(pos = n())%>%
  arrange(-pos)%>%
  slice(1:10)

chains$pos <- as.numeric(chains$pos)

chains%>%
```

```
  ggplot(aes(reorder(rest_brand, -pos), pos)) +
  geom_col(fill='red') +
  labs(x='Brand', y='N. of Branches in London') +
  ggtitle('Top 10 chains') +
  coord_flip() +
  ggsave('top10chains.png')
```

```
## Saving 6.5 x 4.5 in image
```




The findings here are quite interesting: the first chain seems to be a mysterious Get Drinks Delivered. Diving deeper in our dataset we can analyze which kind of restaurants are part of this brand.

```
get_drinks_delivered <- info_rest%>%
  group_by(rest_brand)%>%
  filter(rest_brand == 'Get drinks delivered')%>%
  list()

get_drinks_delivered
```

```
## [[1]]
## # A tibble: 42 x 7
## # Groups:   rest_brand [1]
##    restaurant_id rest_name rest_brand rest_postcode rest_neighborho~ rest_rating
##            <dbl> <chr>     <chr>      <chr>         <chr>                  <dbl>
##  1         17365 Hop Burn~ Get drink~ SE229AX       Peckham                  4.9
##  2         61860 Wine Mart Get drink~ W127JA        Shepherd's Bush          4.8
##  3         46093 Majestic~ Get drink~ SW152RD       Putney                   4.9
```

```
##  4         46620 Off Lice~ Get drink~ N14AX         Dalston                  4.8
##  5         35467 Off Lice~ Get drink~ NW24QU        Willesden Green          4.7
##  6         55365 Winche    Get drink~ W138PH        Ealing                   4.6
##  7         53311 TST Wines Get drink~ W54QB         Ealing                   4.8
##  8         38371 Brew Hou~ Get drink~ CR43BF        Tooting                 NA
##  9         44640 Off Lice~ Get drink~ W120JH        Shepherd's Bush          4.3
## 10         38419 Loco Dri~ Get drink~ SW98RR        Brixton                  4.7
## # ... with 32 more rows, and 1 more variable: rest_menu_item_price <list>
```

To roam in an easier way in between these names it's useful to perform some text mining on the names of the restaurants part of Get drinks delivered.

```
get_drinks_vector <- unlist(get_drinks_delivered)
get_drinks_vector <- na.omit(get_drinks_vector)

text_drinks <- Corpus(VectorSource(get_drinks_vector))
text_drinks <- tm_map(text_drinks, removeNumbers)
text_drinks <- tm_map(text_drinks, removePunctuation)
text_drinks <- tm_map(text_drinks, stripWhitespace)
text_drinks <- tm_map(text_drinks, removeWords, stopwords('english'))

drinks <- TermDocumentMatrix(text_drinks)
matrix_drinks <- as.matrix(drinks)
matrix_sorted <- sort(rowSums(matrix_drinks),decreasing=TRUE)
result_drinks <- data.frame(word = names(matrix_sorted),freq=matrix_sorted)
head(result_drinks, 15)
```

```
##                word freq
## drinks       drinks   45
## delivered delivered   42
## get              get   42
## house          house    9
## wine            wine    8
## brew            brew    8
## ealing        ealing    7
## licence      licence    5
## off              off    5
## bush            bush    4
## shepherds shepherds    4
## stratford stratford    4
## tooting      tooting    4
## balham        balham    4
## wines          wines    3
```

Liquor stores in London are exactly as common as, for example, KFC or Pizza Express branches, and double they are almost twice the number of Papa John's restaurants.

*2.1.4 Average menu price and number of menu items*

First of all I have found the number of items on the menu of each restaurant unnesting the list rest_menu_item_price and turning the elements into logical ones to be able to calculate the mean.

```
items_raw <- info_rest%>%
  filter(restaurant_id %>%
           map_lgl(any))%>%
  unnest(rest_menu_item_price)
```

```
items_clean <- items_raw[apply(items_raw[c(7)],1,function(z) any(!z==0)),]

items_number <- items_clean%>%
  group_by(rest_name)%>%
  summarise(menu_items_number = n())

tail(items_number)
```

```
## # A tibble: 6 x 2
##   rest_name            menu_items_number
##   <chr>                            <int>
## 1 Zu's Eat                           120
## 2 Zubi Galley                         67
## 3 Zumbura                             52
## 4 ZuZu Grill                          51
## 5 ZZ Green- Shoreditch                38
## 6 <NA>                                14
```

We can now calculate the average price of the items on the menu of each restaurant, useful to get an idea of the range of customers and the quality of the food/service provided.

```
items_average_price <- items_clean%>%
  group_by(restaurant_id, rest_name)%>%
  filter(!is.na(restaurant_id))%>%
  summarise_at(vars(rest_menu_item_price), funs(mean (.)), na.rm=TRUE)

colnames(items_average_price)[3] <- 'average_price'
items_average_price$average_price <- sprintf(items_average_price$average_price, fmt = '%#.2f')

head(items_average_price)
```

```
## # A tibble: 6 x 3
## # Groups:   restaurant_id [6]
##   restaurant_id rest_name                    average_price
##           <dbl> <chr>                        <chr>
## 1             3 Busaba Chelsea               9.81
## 2             5 Rossopomodoro                11.27
## 3             8 New Culture Revolution       8.84
## 4            10 Mandaloun                    9.72
## 5            15 Busaba St Christopher's Place 9.61
## 6            16 Busaba Bloomsbury            9.50
```

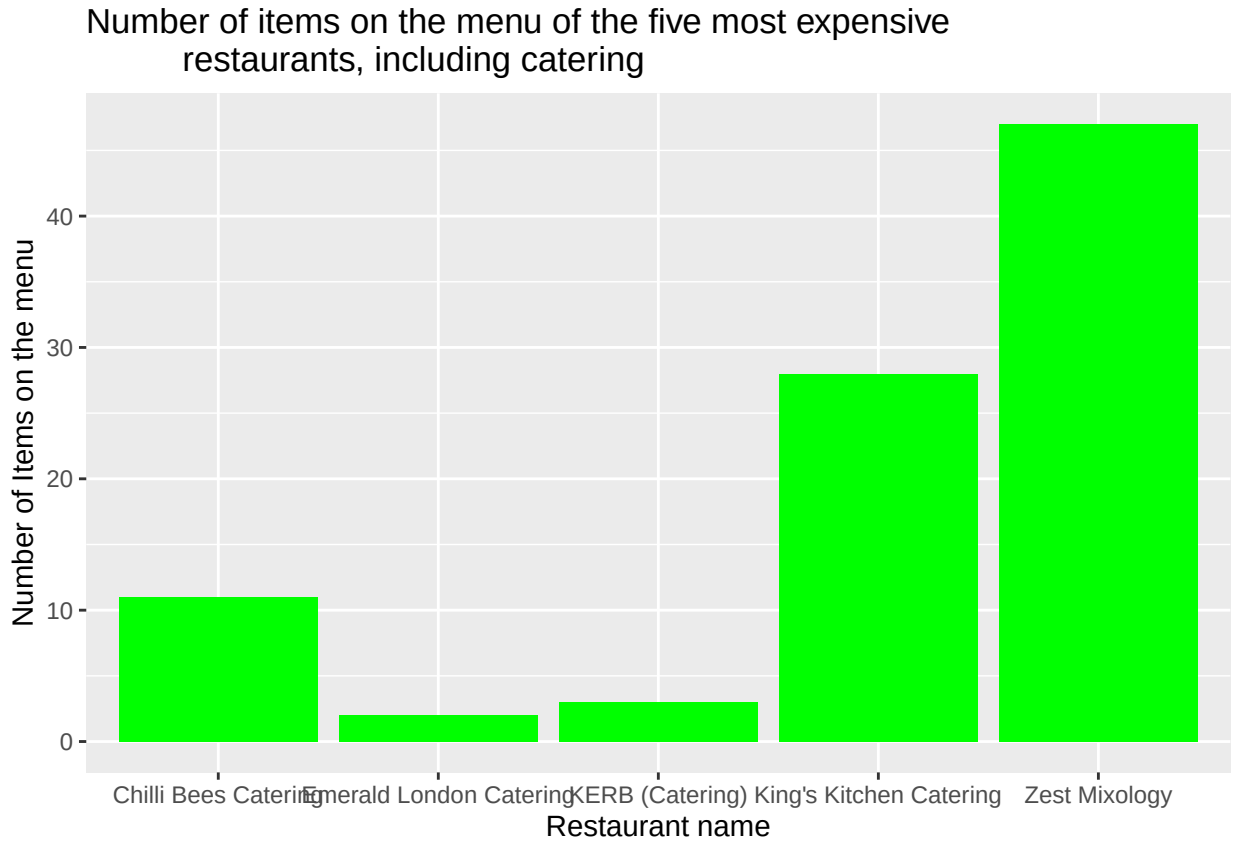
*2.1.5 Number of items on the menu of the five most expensive and cheapest restaurants*

In this section we cross the data that we have found answering the previous two questions.

```
pricey <- left_join(items_number, items_average_price, by = 'rest_name')

pricey$average_price <- as.numeric(pricey$average_price)

top5_with_catering <- pricey%>%
  arrange(-average_price) %>%
  head(5)

top5_with_catering%>%
  ggplot(aes(x=rest_name, y = menu_items_number),
```

```
       xlab = 'Restaurant', geom = 'bar',
       ylab = 'Number of items on menu') +
  geom_col(fill='green') +
  labs(x='Restaurant name', y='Number of Items on the menu') +
  ggtitle('Number of items on the menu of the five most expensive
          restaurants, including catering') +
  ggsave('5mostexpwcatering.png')
```

```
## Saving 6.5 x 4.5 in image
```

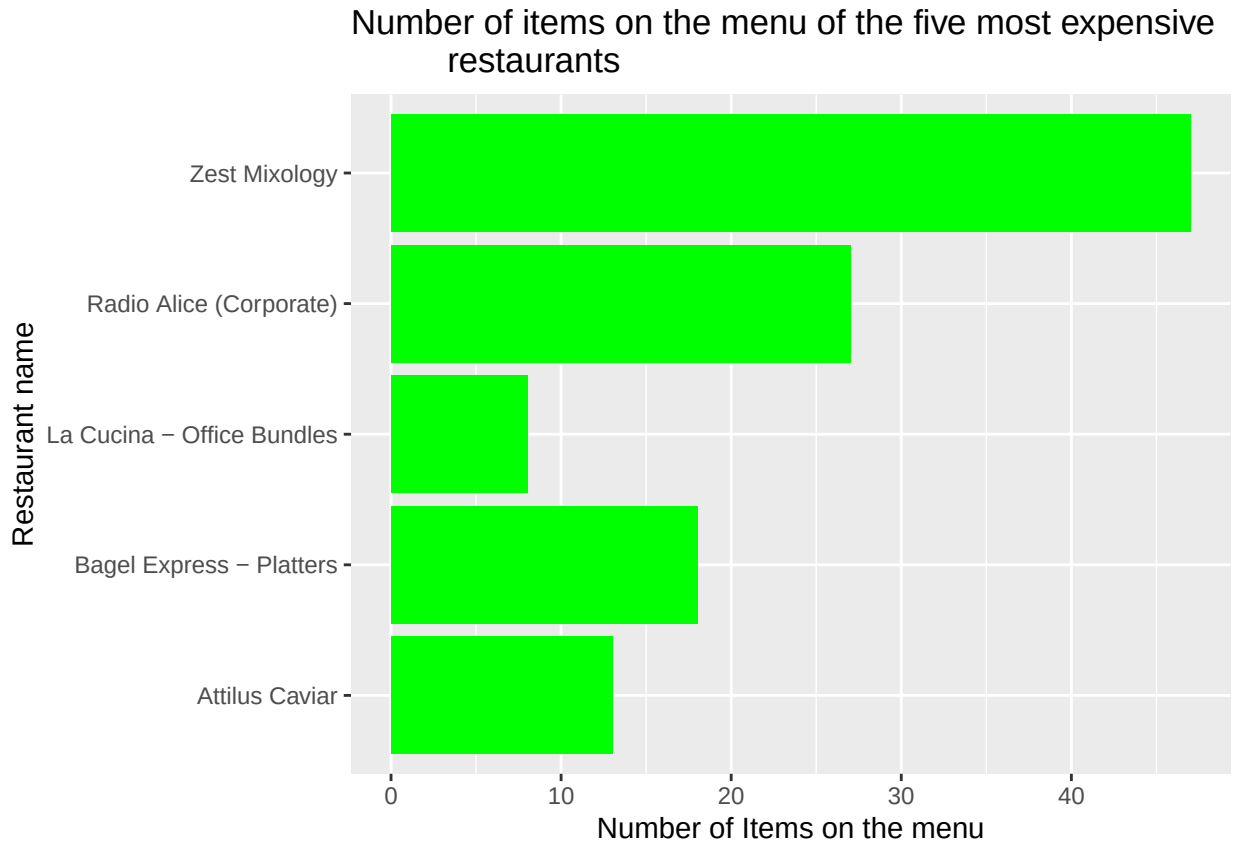



It's easy to notice from the chart that the first four names shown have a really high average price. This is explained by the fact that the names shown are not actual restaurants but catering companies that probably list the price of their services. These prices have been misinterpreted as menu items prices while collecting the data, so it makes sense to calculate this ranking again excluding the observations that have the word "catering" in the name.

```
top5_wo_catering <- pricey%>%
  filter(!grepl('Catering', rest_name))%>%
  arrange(-average_price) %>%
  head(5)

top5_wo_catering%>%
  ggplot(aes(x=rest_name, y = menu_items_number),
         xlab = 'Restaurant', geom = 'bar',
         ylab = 'Number of items on menu') +
  geom_col(fill='green') +
  labs(x='Restaurant name', y='Number of Items on the menu') +
```

```
  coord_flip() +
  ggtitle('Number of items on the menu of the five most expensive
          restaurants') +
  ggsave('5mostexpwocatering.png')
```

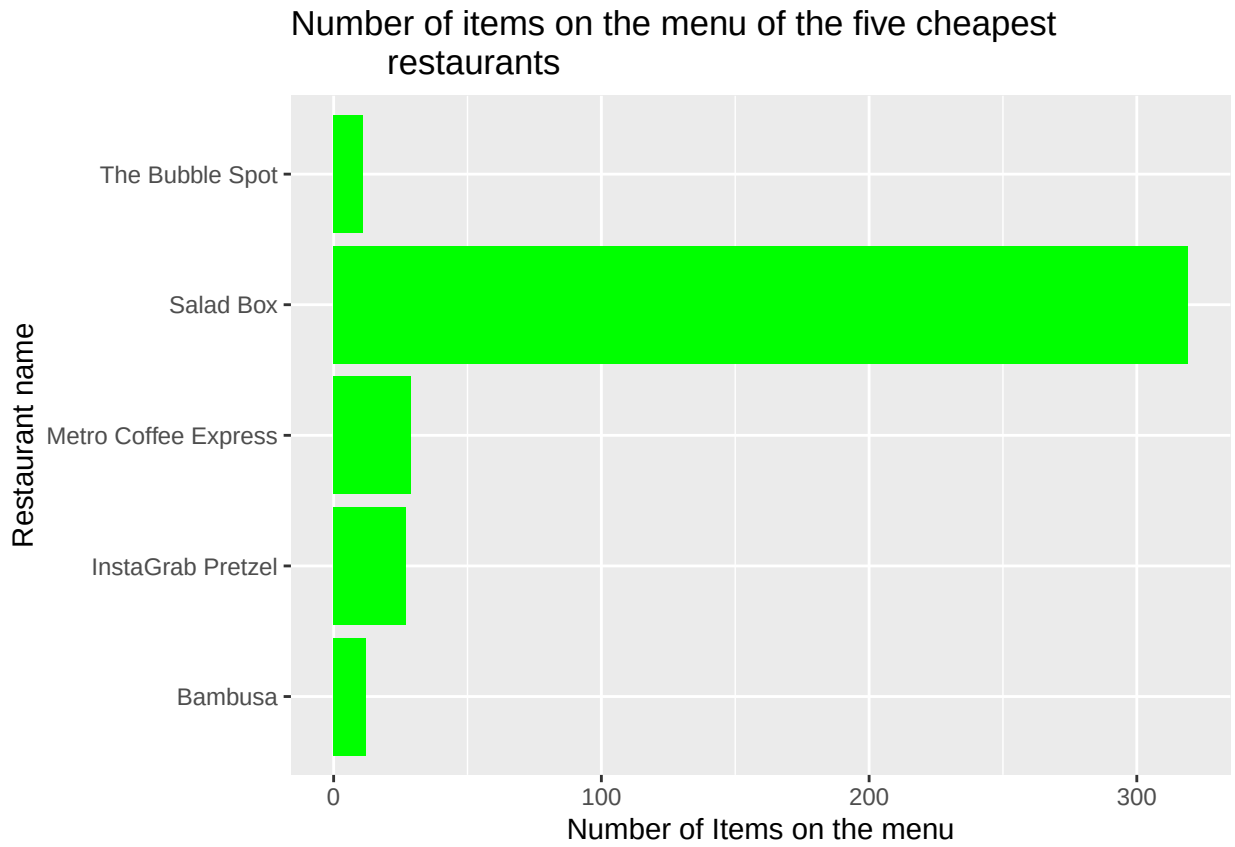
```
## Saving 6.5 x 4.5 in image
```




Let's do the same analysis for what concerns the 5 cheapest restaurants.

```
bottom5 <- pricey%>%
  arrange(average_price)%>%
  head(5)

bottom5%>%
  ggplot(aes(x=rest_name, y = menu_items_number),
         xlab = 'Restaurant', geom = 'bar',
         ylab = 'Number of items on menu') +
  geom_col(fill='green') +
  labs(x='Restaurant name', y='Number of Items on the menu') +
  coord_flip() +
  ggtitle('Number of items on the menu of the five cheapest
          restaurants') +
  ggsave('5cheapest.png')
```

```
## Saving 6.5 x 4.5 in image
```




In between the cheapest restaurants Salad Box is the one with the highest amount of items on the menu.

#2.2 Restaurant Delivery Times Analysis

Before starting the second part of the analysis we need to join the two datasets.

```
colnames(delivery)[1] <- 'restaurant_id'

restaurants <- info_rest%>%
  select(-rest_neighborhood)

london <- left_join(delivery, restaurants, by = 'restaurant_id')
head(london)
```

```
## # A tibble: 6 x 8
##   restaurant_id neighborhood_na~ rest_delivery_t~ rest_name rest_brand
##           <dbl> <chr>                      <dbl> <chr>     <chr>
## 1         98636 the-city                      10 Farmer J~ Farmer J
## 2        167932 the-city                      10 Acai Ber~ Acai Berr~
## 3           902 the-city                      15 La Cucina La Cucina
## 4         22555 the-city                      15 Polo      Polo
## 5         29850 the-city                      10 Crosstow~ Crosstown~
## 6        202819 the-city                      25 EggsQuis~ <NA>
## # ... with 3 more variables: rest_postcode <chr>, rest_rating <dbl>,
## #   rest_menu_item_price <list>
```

*2.2.1 Number of neighborhoods where each restaurant delivers*

Using the function group_by I have found the number of neighborhoods that each restaurant delivers to.

```
deliver_to <- london%>%
  group_by(restaurant_id)%>%
  summarise(neighb_served=n())
head(deliver_to)
```

```
## # A tibble: 6 x 2
##   restaurant_id neighb_served
##           <dbl>         <int>
## 1             3            47
## 2             5            52
## 3             8            52
## 4            10            29
## 5            15            19
## 6            16            65
```

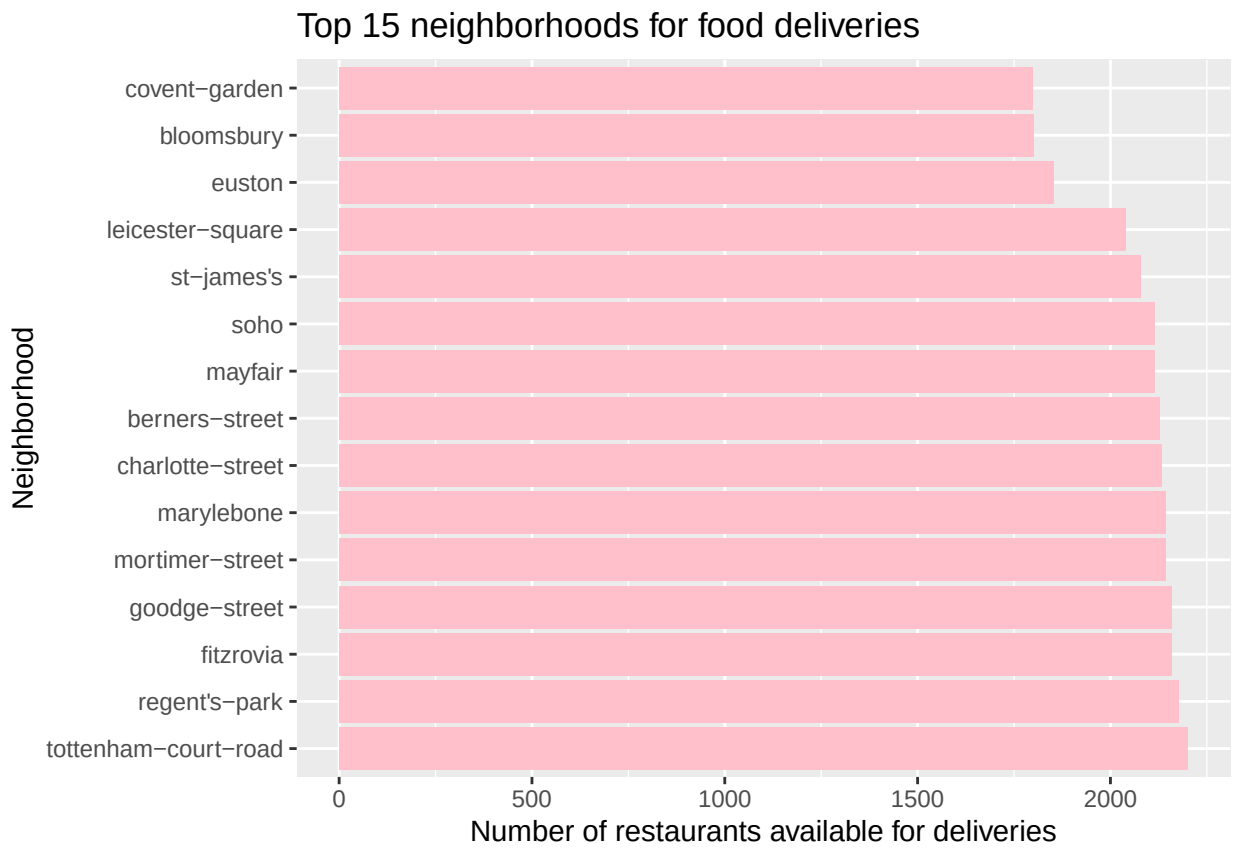
```
average_neighborhoods_served <- as.integer(mean(deliver_to$neighb_served))
cat('On average each restaurant in London that has the delivery option delivers to', average_neighborho
```

```
## On average each restaurant in London that has the delivery option delivers to 35 neighborhoods.
```

*2.2.2 Top 15 neighborhoods where restaurants make deliveries*

```
delivered <- london%>%
  group_by(neighborhood_name)%>%
  summarise(delivered_by_n = n())

most_delivered <- delivered%>%
  arrange(-delivered_by_n)%>%
  head(15)

most_delivered%>%
  ggplot(aes(reorder(neighborhood_name, -delivered_by_n), delivered_by_n)) +
  geom_col(fill='pink') +
  coord_flip() +
  labs(x='Neighborhood', y='Number of restaurants available for deliveries') +
  ggtitle('Top 15 neighborhoods for food deliveries') +
  ggsave('15topdeliveries.png')
```

```
## Saving 6.5 x 4.5 in image
```




More than 2000 of the restaurant included in the dataset deliver to Tottenham Court Road, Regent's Park and Fitzrovia.

*2.2.3 Average delivery time for each restaurant*

Using the functions group_by and mean I can calculate the average time of delivery for each restaurant.

```
average_times <- london%>%
  group_by(restaurant_id, rest_name, rest_postcode, rest_rating)%>%
  summarise(average_delivery_time = mean(rest_delivery_time_min, na.rm=TRUE))

average_times$average_delivery_time <- as.integer(average_times$average_delivery_time)

head(average_times)
```

```
## # A tibble: 6 x 5
## # Groups:   restaurant_id, rest_name, rest_postcode [6]
##   restaurant_id rest_name           rest_postcode rest_rating average_delivery_~
##           <dbl> <chr>               <chr>               <dbl>              <int>
## 1             3 Busaba Chelsea      SW35UZ                4.6                 19
## 2             5 Rossopomodoro       SW109NB               4.5                 17
## 3             8 New Culture Revolu~ SW35EP                4.7                 18
## 4            10 Mandaloun           SW109TW               4.8                 18
## 5            15 Busaba St Christop~ W1U1BU                4.7                 21
## 6            16 Busaba Bloomsbury   WC1E7DF               4.7                 22
```

*2.2.4 Top 20 restaurants by average delivery time*

Once calculated the average delivery time for each of the restaurants in the dataset it's easy to find the

fastest 20 restaurants.

```
fast_and_20 <- average_times%>%
  arrange(average_delivery_time)%>%
  head(20)

fast_and_20
```

```
## # A tibble: 20 x 5
## # Groups:   restaurant_id, rest_name, rest_postcode [20]
##    restaurant_id rest_name           rest_postcode rest_rating average_delivery~
##            <dbl> <chr>               <chr>               <dbl>             <int>
##  1         14632 Vagabond Wines - F~ SW61AX                4.8                 8
##  2         21767 OREE                SW109PZ               4.8                 8
##  3         24414 Le Pain Quotidien   SW35ED                4.8                 8
##  4         13095 Pure                W1F9SA                4.7                 9
##  5        178124 Wasabi              W1D7EH                4.6                 9
##  6        178142 Wasabi              W1T1BG                4.6                 9
##  7          1229 Mangio Pasta        EC4V5JT               4.5                10
##  8          1518 LEON                WC1V6AZ               4.5                10
##  9          3853 Amorino - Old Comp~ W1D6HF                4.8                10
## 10          4219 Brew Café           SW195DX               4.5                10
## 11         19501 Thai Tho            SW195DX               4.6                10
## 12         25160 PizzaExpress        W1D4DU                4.5                10
## 13         28260 Jeroboams Wine Mer~ SW31RB                4.9                10
## 14         31181 Le Pain Quotidien   SW109PJ               4.8                10
## 15         44709 Pret A Manger       SW197NS               4.6                10
## 16         54832 Pret A Manger       SW61BW                4.7                10
## 17         54846 Pret A Manger       W1H6HJ                4.8                10
## 18         55365 Winche              W138PH                4.6                10
## 19         55892 Maison St Cassien   SW195EE               4.6                10
## 20         60643 Friarwood Fine Win~ SW195BY               4.9                10
```

#2.3 Type of food analysis_

Within this open analysis I have decided to dive deeper in Londoners food preferences, performing some text mining on the names of the restaurants as a first step.

*2.3.1 Name your food*

Often the name of the restaurant is the main and first feature customers are attracted by. It is a main vehicle through which the owner can communicate to the public in general about which food is served there, the ethnicity, the heritage and sometimes also convey an idea about the prices of the menu. To gain a deeper understanding of the food scene in London I have performed some text mining on the restaurant names.

```
name_vector <- as.vector(info_rest$rest_name)
name_vector <- na.omit(name_vector)

text_tba <- Corpus(VectorSource(name_vector))
text_tba <- tm_map(text_tba, removeNumbers)
```

```
## Warning in tm_map.SimpleCorpus(text_tba, removeNumbers): transformation drops
## documents
```

```
text_tba <- tm_map(text_tba, removePunctuation)
```

```
## Warning in tm_map.SimpleCorpus(text_tba, removePunctuation): transformation
## drops documents
```

```
text_tba <- tm_map(text_tba, stripWhitespace)
```

```
## Warning in tm_map.SimpleCorpus(text_tba, stripWhitespace): transformation drops
## documents
```

```
text_tba <- tm_map(text_tba, removeWords, stopwords('english'))
```

```
## Warning in tm_map.SimpleCorpus(text_tba, removeWords, stopwords("english")):
## transformation drops documents
```

```
dtm <- TermDocumentMatrix(text_tba)
a <- as.matrix(dtm)
b <- sort(rowSums(a),decreasing=TRUE)
c <- data.frame(word = names(b),freq=b)
head(c, 10)
```

```
##                  word freq
## pizza           pizza  344
## the               the  293
## restaurant restaurant  241
## chicken       chicken  179
## kitchen       kitchen  175
## grill           grill  147
## cafe             cafe  139
## bar               bar  127
## burger         burger  119
## food             food  112
```

Looks like Londoners love pizza more than any other food! Visualizing the the results it's easier with a nice wordcloud.

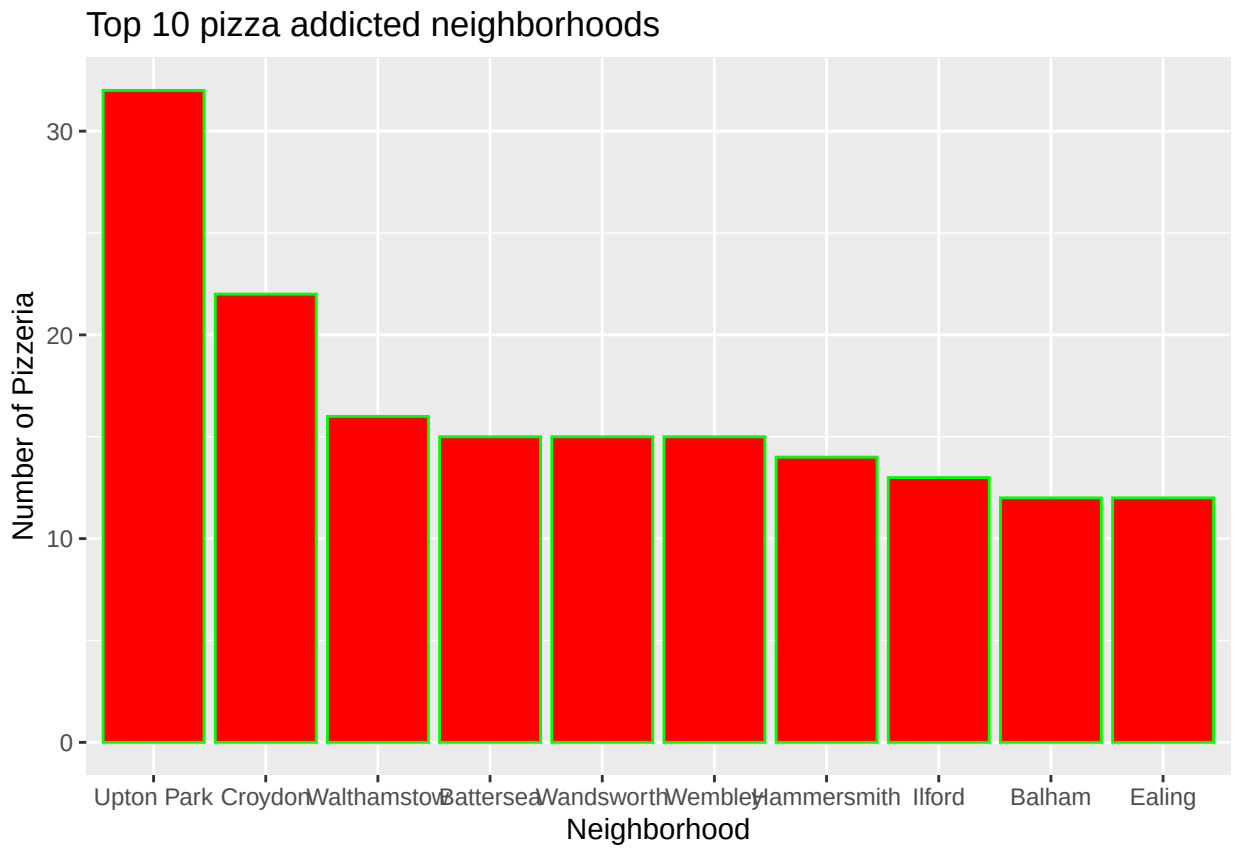
```
wordcloud (words = c$word, freq = c$freq, min.freq = 10,
          max.words=200, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))
```


*2.3.2 Pizza addicted Neighborhoods*

We have found out that pizza stands out as the word that has the highest frequency within the list of London restaurants names. This means that pizza places are the most common restaurants in our analysis. But are there neighborhoods that love pizza more than others? Probably yes, so let's try to find out our top 10.

```
areas_names <- info_rest%>%
  select(2,4,5)

areas_names$rest_name <- tolower(areas_names$rest_name)

pizza_areas <- areas_names%>%
  group_by(rest_neighborhood, .drop = FALSE)%>%
  filter(grepl('pizza', rest_name))%>%
  summarise(pizzerias = n())

pizza_lovers_neighborhood <- pizza_areas%>%
  arrange(-pizzerias)%>%
  slice(1:10)

pizza_lovers_neighborhood%>%
  ggplot(aes(reorder (rest_neighborhood, -pizzerias), pizzerias)) +
  geom_col(fill= 'red', color = 'green') +
  labs(x='Neighborhood', y='Number of Pizzeria') +
  ggtitle('Top 10 pizza addicted neighborhoods') +
  ggsave('top10pizzaddicted.png')
```

```
## Saving 6.5 x 4.5 in image
```




Upton Park has the highest density of pizza places, followed by Croydon and Walthamstow.

*2.3.3 Restaurant Maps*

```
london_map <- here('data','londonshp.shp')%>%
  st_read()
```

```
## Reading layer `londonshp' from data source `C:\Users\Panda\Desktop\london_res\data\londonshp.shp' usi
## Simple feature collection with 181150 features and 13 fields
## geometry type:  POINT
## dimension:      XY
## bbox:           xmin: 503978 ymin: 156844 xmax: 561051 ymax: 200536
## epsg (SRID):    NA
## proj4string:    +proj=tmerc +lat_0=49 +lon_0=-2 +k=0.9996012717 +x_0=400000 +y_0=-100000 +datum=OSGB3
```

```
clean_london_map <- london_map%>%
  mutate(postcode = gsub(" ", "", postcode))

real_map <- ggplot() + geom_polygon(data = clean_london_map, aes(x = eastings, y = northings), colour =
```

We don't have information about longitude and latitude in our restaurant dataset, so to plot the points on our map we need to join the dataset of the map with northing and easting to the restaurant dataset, with an inner join to keep just the postcodes from the restaurant dataset.

```
colnames(areas_names)[2] <- 'postcode'

restaurants_to_plot <- inner_join(areas_names, clean_london_map, by = 'postcode')
```

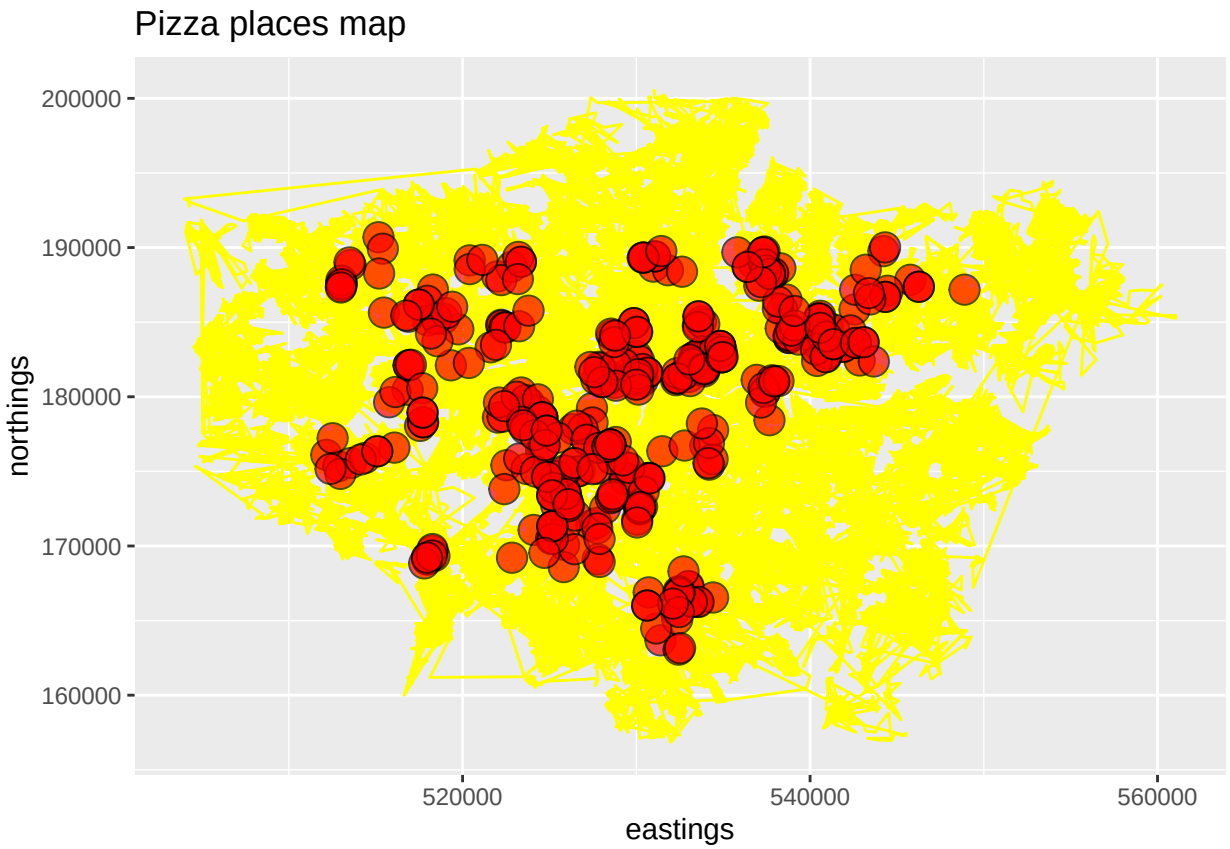
```
real_map +
  geom_point(data = restaurants_to_plot, aes(x=eastings, y=northings),  colour='Deep Pink' , fill='pink
  ggtitle('Restaurants map') +
  ggsave('mapall.png')
```

```
## Saving 6.5 x 4.5 in image
```


Then I have decided to create a map that only includes the restaurants that have pizza in the name.

```
pizzerias_to_plot <- restaurants_to_plot%>%
  filter(grepl('pizza', rest_name))

real_map +
  geom_point(data = pizzerias_to_plot, aes(x= eastings, y=northings),  fill='red',pch=21, size=5, alpha=
  ggtitle('Pizza places map') +
  ggsave('pizzamap.png')
```

```
## Saving 6.5 x 4.5 in image
```


Pizza places map
200000
190000
180000
170000
160000
northings
520000
540000
560000
eastings