# London Restaurant Scene project- MIBE

Edoardo Mustarelli

6/3/2021

```
# ipak function: install and load multiple R packages.
# check to see if packages are installed. Install them if they are not, then load them into the R sessi

ipak <- function(pkg){
new.pkg <- pkg[!(pkg %in% installed.packages()[, "Package"])]
if (length(new.pkg))
    install.packages(new.pkg, dependencies = TRUE)
sapply(pkg, require, character.only = TRUE)
}

# usage
packages <- c("dplyr", "tidyverse", "tidyr", "here", "magrittr", "purrr", "purrrlyr", "ggplot2", "forma
ipak(packages)
```

**Let's start with the analysis of the files. First I am going to load the libraries necessary, or that may be necessary, to move on with the analysis.**

```
##        dplyr    tidyverse        tidyr         here     magrittr        purrr
##         TRUE         TRUE         TRUE         TRUE         TRUE         TRUE
##     purrrlyr      ggplot2  formattable        rlist       gtools           tm
##         TRUE         TRUE         TRUE         TRUE         TRUE         TRUE
##    SnowballC    wordcloud RColorBrewer           sf         tmap    tmaptools
##         TRUE         TRUE         TRUE         TRUE         TRUE         TRUE
##        rgdal        rgeos        ggmap     tidytext       ggraph        readr
##         TRUE         TRUE         TRUE         TRUE         TRUE         TRUE
##    htmltools      webshot
##         TRUE         TRUE
```

```
restaurants_info <- readRDS(file="C:/Users/Edoardo/Desktop/LSU_project/resturants-mibe.rds")
info_delivery <- readRDS(file="C:/Users/Edoardo/Desktop/LSU_project/delivery-mibe.rds")
```

I now want to inspect the dataset regarding the restaurants' information to have a better understanding of what I'm working with.

```
glimpse(restaurants_info)
```

```
## Rows: 5,786
## Columns: 7
## $ restaurant_id        <dbl> 191295, 54515, 113653, 184167, 84922, 194571, 136~
## $ 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", "SW151JP~
## $ rest_neighborhood    <chr> "Hendon", "Fitzrovia", "Wembley", "Croydon", "Put~
## $ rest_rating          <dbl> NA, 4.7, NA, 3.8, 4.3, 4.4, 4.6, 4.2, 4.8, 4.6, 4~
## $ rest_menu_item_price <list> <2.80, 4.20, 5.60, 4.20, 5.60, 14.00, 16.80, 4.2~
```

```
head(restaurants_info)
```

```
## # 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 Wali~ <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 Dom~ <NA>       SW151JP       Putney                   4.3
## 6        194571 Kin + Deum <NA>       E146AB        Canary Wharf             4.4
## # ... with 1 more variable: rest_menu_item_price <list>
```

We can see that the restaurants_info file has 7 variables which are id, name, brand, postcode, neighborhood, rating, and menu item price. We can also see that we have the data for 5,786 restaurants in London. #### 1 Restaurant Information Analysis

1.1 top 10 neighborhoods by the number of restaurants

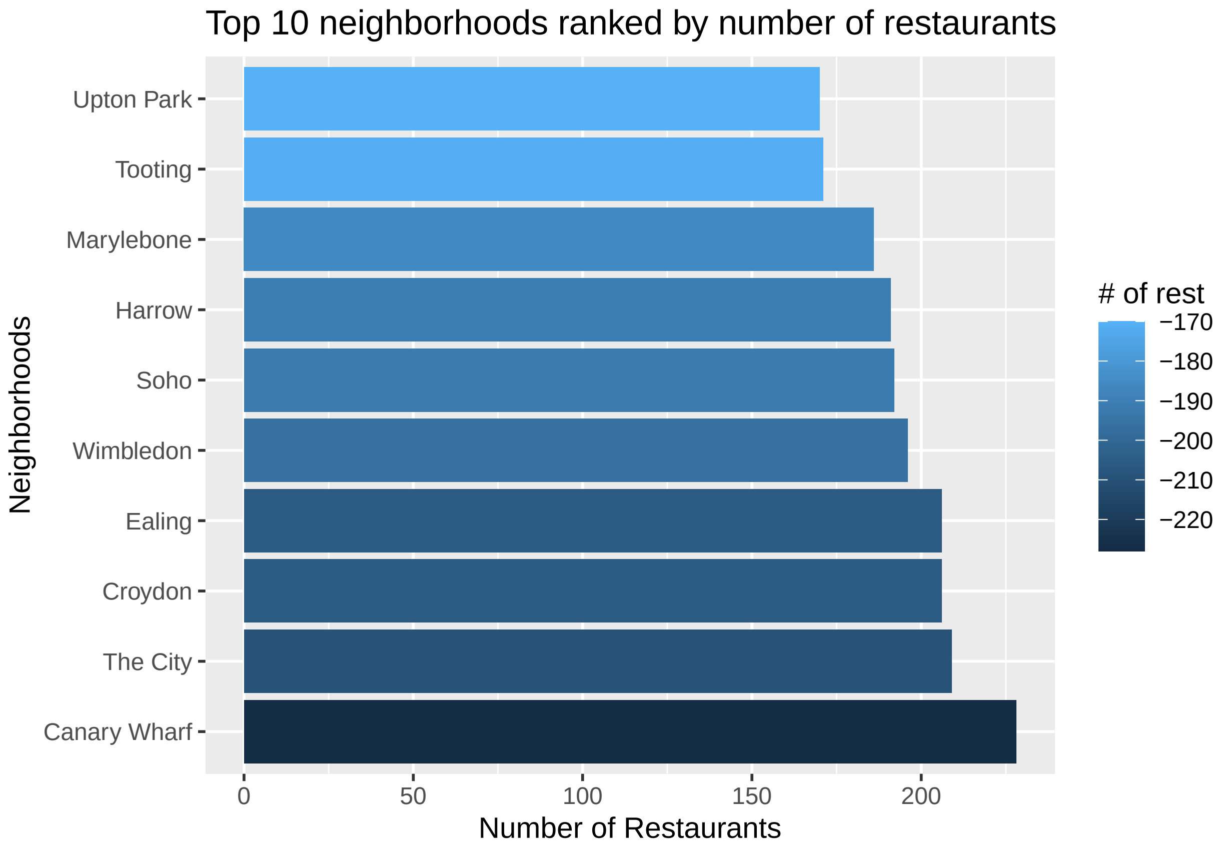
```
top10_neighborhoods <- restaurants_info %>%
  group_by(rest_neighborhood) %>%
  filter(!is.na(rest_neighborhood)) %>%
  summarise(number_of_rest=n()) %>%
  arrange(-number_of_rest) %>%
  slice(1:10)
top10_neighborhoods
```

```
## # A tibble: 10 x 2
##    rest_neighborhood number_of_rest
##    <chr>                      <int>
##  1 Canary Wharf                 228
##  2 The City                     209
##  3 Croydon                      206
##  4 Ealing                       206
##  5 Wimbledon                    196
##  6 Soho                         192
##  7 Harrow                       191
##  8 Marylebone                   186
##  9 Tooting                      171
## 10 Upton Park                   170
```

```
top10_neighborhoods$number_of_rest <- as.numeric(top10_neighborhoods$number_of_rest)

top10_neighborhoods %>%
ggplot(aes(reorder(rest_neighborhood, -number_of_rest), number_of_rest, fill=-number_of_rest)) +
  geom_bar(stat = "identity")+
  labs(x = "Neighborhoods", y = "Number of Restaurants", fill="# of rest")+
  ggtitle("Top 10 neighborhoods ranked by number of restaurants")+
  coord_flip()+
  ggsave("10neigh_by_#rest.png")
```

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



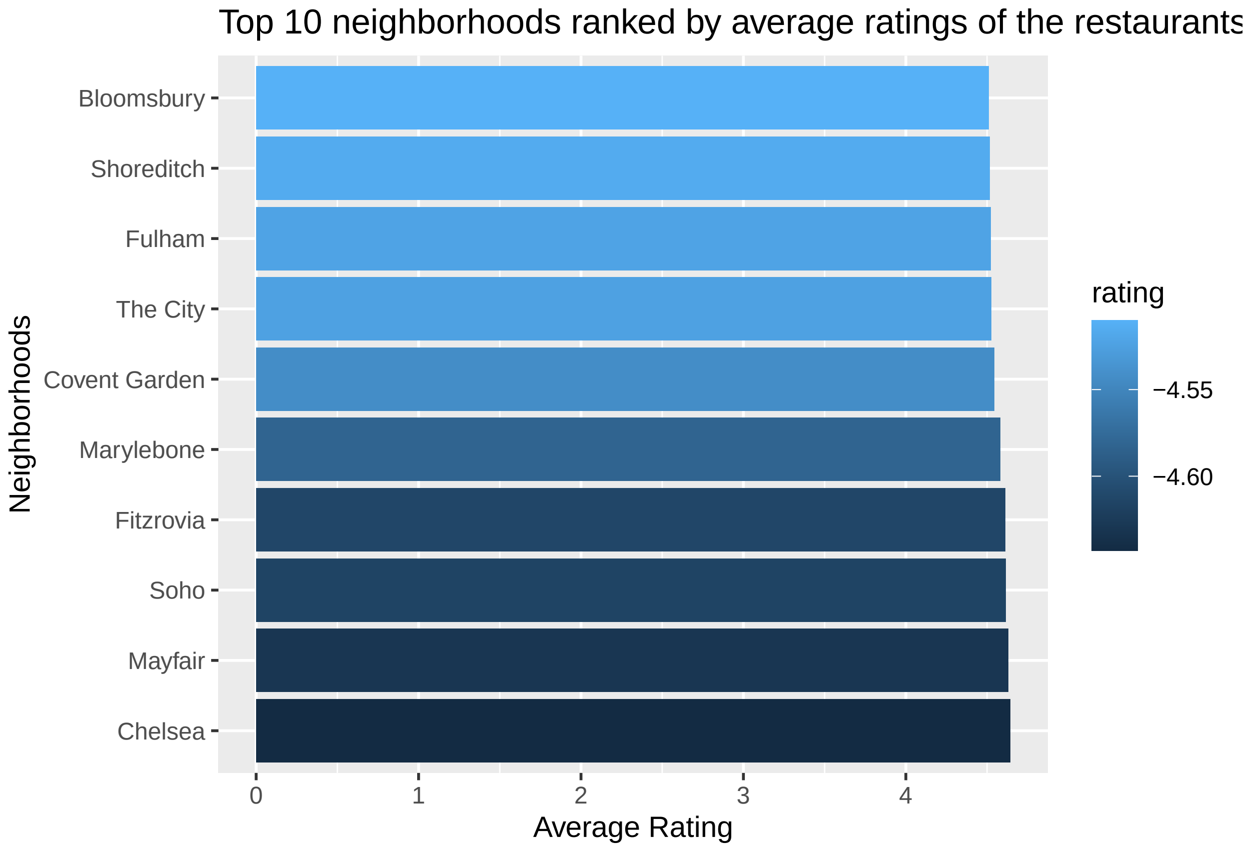
1.2 top 10 neighborhoods by the restaurant review score

```
top10_neig_ratings <- restaurants_info %>%
  group_by(rest_neighborhood) %>%
  summarise_at(vars(rest_rating),list(~mean(., na.rm = TRUE))) %>%
  arrange(-rest_rating) %>%
  slice(1:10)
top10_neig_ratings
```

```
## # A tibble: 10 x 2
##    rest_neighborhood rest_rating
##    <chr>                   <dbl>
##  1 Chelsea                  4.64
##  2 Mayfair                  4.63
##  3 Soho                     4.62
##  4 Fitzrovia                4.61
##   5 Marylebone                 4.58
##   6 Covent Garden              4.54
##   7 The City                   4.52
##   8 Fulham                     4.52
##   9 Shoreditch                 4.52
##  10 Bloomsbury                 4.51
```

```
top10_neig_ratings %>%
ggplot(aes(reorder(rest_neighborhood, -rest_rating), rest_rating, fill=-rest_rating)) +
  geom_bar(stat = "identity")+
  labs(x = "Neighborhoods", y = "Average Rating", fill="rating")+
  ggtitle("Top 10 neighborhoods ranked by average ratings of the restaurants")+
  coord_flip()+
  ggsave("10neigh_by_avg_rating.png")
```

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



```
top_neigh_overall <- inner_join(top10_neighborhoods, top10_neig_ratings)
```

```
## Joining, by = "rest_neighborhood"
```

```
top_neigh_overall
```

```
## # A tibble: 3 x 3
##   rest_neighborhood number_of_rest rest_rating
##   <chr>                      <dbl>       <dbl>
## 1 The City                     209        4.52
## 2 Soho                         192        4.62
## 3 Marylebone                   186        4.58
```

1.3 Top 10 biggest chains

```
top_chains <- restaurants_info %>%
  group_by(rest_brand) %>%
  filter(!is.na(rest_brand)) %>%
  summarise(number_of_spots = n()) %>%
  arrange(-number_of_spots) %>%
  slice(1:10)
top_chains
```

```
## # A tibble: 10 x 2
##    rest_brand           number_of_spots
##    <chr>                          <int>
##  1 Get drinks delivered              42
##  2 KFC                               42
##  3 PizzaExpress                      42
##  4 Pret A Manger                     33
##  5 Burger King                       22
##  6 itsu                              22
##  7 Pure                              21
##  8 Wasabi                            20
##  9 LEON                              19
## 10 Papa John's                       18
```

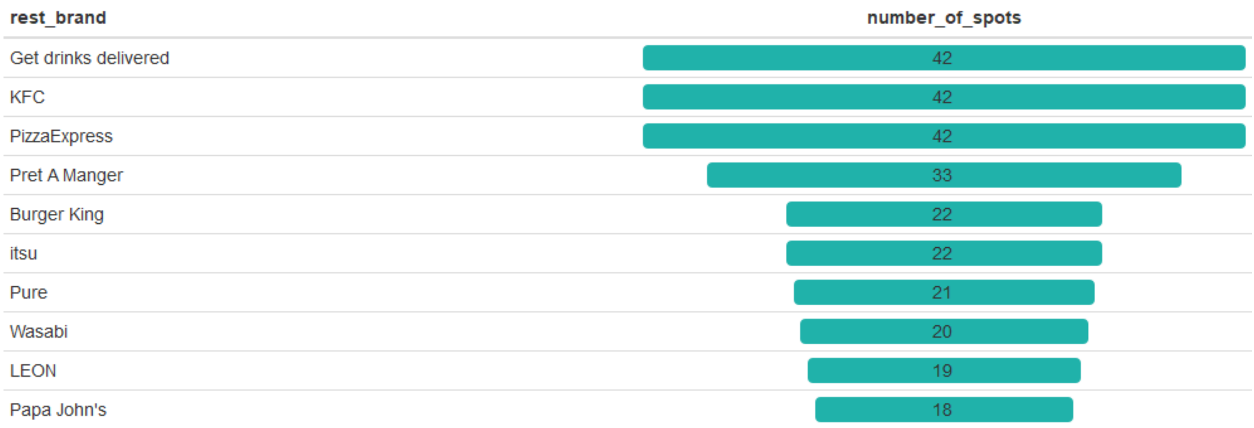
```
top_chains$number_of_spots <- as.numeric(top_chains$number_of_spots)
 webshot::install_phantomjs(force = TRUE)
```

```
## phantomjs has been installed to C:\Users\Edoardo\AppData\Roaming\PhantomJS
```

```
export_formattable <- function(f, file, width = "100%", height = NULL,
                               background = "white", delay = 0.2)
    {
      w <- as.htmlwidget(f, width = width, height = height)
      path <- html_print(w, background = background, viewer = NULL)
      url <- paste0("file:///", gsub("\\\\", "/", normalizePath(path)))
      webshot(url,
              file = file,
              selector = ".formattable_widget",
              delay = delay)
    }

FT <- top_chains %>%
  formattable(align = c("l","c"), list(number_of_spots = color_bar(("lightseagreen"))))

export_formattable(FT,"FT.png")
```



| rest_brand | number_of_spots |
|---|---|
| Get drinks delivered | 42 |
| KFC | 42 |
| PizzaExpress | 42 |
| Pret A Manger | 33 |
| Burger King | 22 |
| itsu | 22 |
| Pure | 21 |
| Wasabi | 20 |
| LEON | 19 |
| Papa John's | 18 |

```
FT
```

rest_brand

number_of_spots

Get drinks delivered

42

KFC

42

PizzaExpress

42

Pret A Manger

33

Burger King

22

itsu

22

Pure

21

Wasabi

20

LEON

19

Papa John’s

18

## 1.4 Average menu price and number of items for each restaurant

```r
number_of_items_gross <- restaurants_info %>%
  filter(restaurant_id %>%
           map_lgl(any)) %>%
  unnest(rest_menu_item_price)
number_of_items_gross
```

```
## # A tibble: 703,861 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        191295 Baba Wal~ <NA>       NW97DY        Hendon                    NA
##  3        191295 Baba Wal~ <NA>       NW97DY        Hendon                    NA
##  4        191295 Baba Wal~ <NA>       NW97DY        Hendon                    NA
##  5        191295 Baba Wal~ <NA>       NW97DY        Hendon                    NA
##  6        191295 Baba Wal~ <NA>       NW97DY        Hendon                    NA
##  7        191295 Baba Wal~ <NA>       NW97DY        Hendon                    NA
##  8        191295 Baba Wal~ <NA>       NW97DY        Hendon                    NA
##  9        191295 Baba Wal~ <NA>       NW97DY        Hendon                    NA
## 10        191295 Baba Wal~ <NA>       NW97DY        Hendon                    NA
## # ... with 703,851 more rows, and 1 more variable: rest_menu_item_price <dbl>
```

```r
number_of_items_net <- number_of_items_gross[apply(number_of_items_gross[c(7)],1, function(del) any(!del
number_of_items_net
```

```
## # A tibble: 581,092 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        191295 Baba Wal~ <NA>       NW97DY        Hendon                    NA
##  3        191295 Baba Wal~ <NA>       NW97DY        Hendon                    NA
##  4        191295 Baba Wal~ <NA>       NW97DY        Hendon                    NA
##  5        191295 Baba Wal~ <NA>       NW97DY        Hendon                    NA
##  6        191295 Baba Wal~ <NA>       NW97DY        Hendon                    NA
##  7        191295 Baba Wal~ <NA>       NW97DY        Hendon                    NA
##  8        191295 Baba Wal~ <NA>       NW97DY        Hendon                    NA
##  9        191295 Baba Wal~ <NA>       NW97DY        Hendon                    NA
## 10        191295 Baba Wal~ <NA>       NW97DY        Hendon                    NA
## # ... with 581,082 more rows, and 1 more variable: rest_menu_item_price <dbl>
```

```r
num_of_items <- number_of_items_net %>%
  group_by(rest_name) %>%
  summarise(num_items_menu = n()) %>%
  arrange(-num_items_menu)
num_of_items
```

```
## # A tibble: 4,522 x 2
##    rest_name             num_items_menu
##    <chr>                          <int>
##  1 PizzaExpress                    8802
##  2 KFC                             5492
##  3 Papa John’s                     4914
##  4 Pizza Hut Delivery              3903
##  5 Pret A Manger                   3405
##  6 BP                              3379
##  7 Tops Pizza                      2804
##  8 itsu                            2389
##  9 Wagamama                        2109
## 10 Pizza Hut Restaurants           2072
## # ... with 4,512 more rows
```

```
avg_price <- number_of_items_net %>%
  group_by(restaurant_id, rest_name) %>%
  filter(!is.na(restaurant_id)) %>%
  summarise_at(vars(rest_menu_item_price), list(~mean(.)), na.rm = TRUE)

colnames(avg_price)[3] <- "Avg_price"

avg_price$Avg_price <- sprintf(avg_price$Avg_price, fmt="%#.4f")

avg_price
```

```
## # A tibble: 5,772 x 3
## # Groups:   restaurant_id [5,772]
##    restaurant_id rest_name                                             Avg_price
##            <dbl> <chr>                                                 <chr>
##  1             3 "Busaba Chelsea"                                      9.8087
##  2             5 "Rossopomodoro"                                       11.2719
##  3             8 "New Culture Revolution"                              8.8381
##  4            10 "Mandaloun"                                           9.7169
##  5            15 "Busaba St Christopher's Place"                       9.6062
##  6            16 "Busaba Bloomsbury"                                   9.4978
##  7            18 "\U0001f1ef\U0001f1f5\U0001f1e7\U0001f1f7 YOOBI \U00~ 8.3522
##  8            19 "Noura"                                               58.8348
##  9            20 "Dozo Sushi"                                          10.9386
## 10            21 "Levant"                                              13.1274
## # ... with 5,762 more rows
```

1.5 how many items on the 5 most expensive restaurants’ menus?

```
#I want to create a joint dataset between the number of items per menu and the average price

how_many_for_how_much <- left_join(num_of_items, avg_price)
```

```
## Joining, by = "rest_name"
```

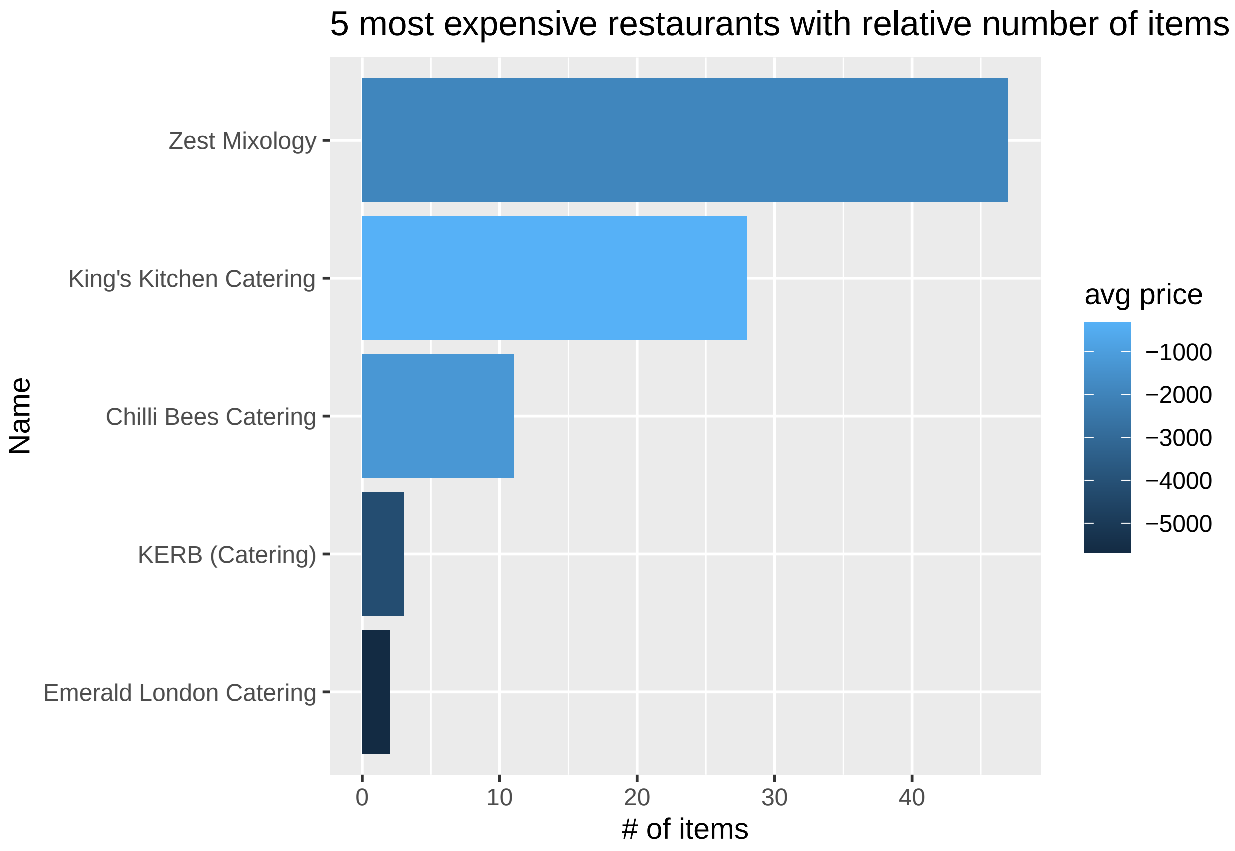
```
how_many_for_how_much$Avg_price <- as.numeric(how_many_for_how_much$Avg_price)

most_expensive <-  how_many_for_how_much %>%
  arrange(-Avg_price) %>%
  slice(1:5)
most_expensive
```

```
## # A tibble: 5 x 4
##   rest_name               num_items_menu restaurant_id Avg_price
##   <chr>                            <int>         <dbl>     <dbl>
## 1 Emerald London Catering              2        156503     5670
## 2 KERB (Catering)                      3        181459     4172.
## 3 Zest Mixology                       47        181439     1928.
## 4 Chilli Bees Catering                11        121538     1267.
## 5 King’s Kitchen Catering             28        177883      323.
```

```
most_expensive %>%
ggplot(aes(reorder(rest_name, num_items_menu), num_items_menu, fill=-Avg_price)) +
  geom_bar(stat = "identity")+
  labs(x = "Name", y = "# of items", fill="avg price")+
  ggtitle("5 most expensive restaurants with relative number of items")+
  coord_flip()+
  ggsave("who_howmuch_howmany.png")
```

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



From this last graph we can observe how there is almost an inverse relationship between the number of items listed on the menus and the average price. As a matter of fact, the two most expensive restaurants also happen to be the ones with less items on their menus. But we can also see that 4/5 of these ase catering services.

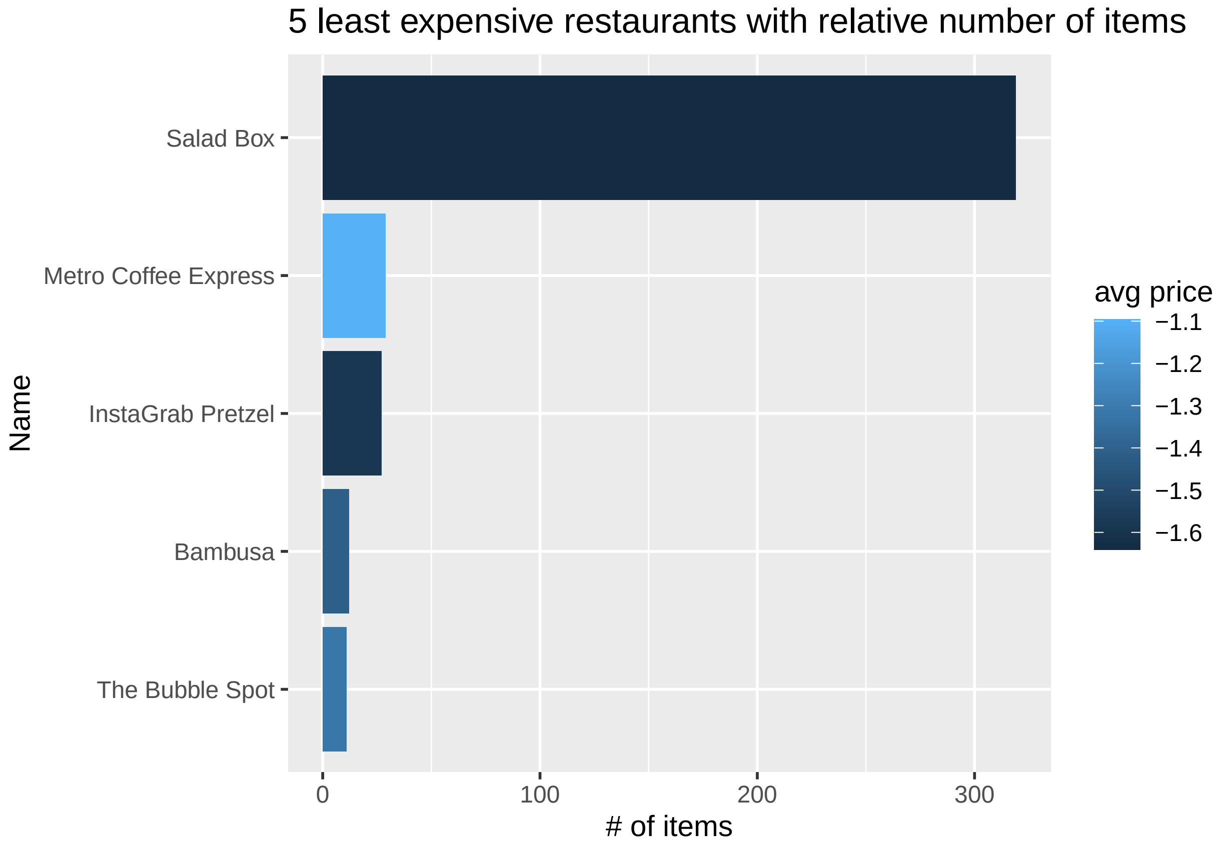
1.6 how many items on the 5 mleast expensive restaurants’ menus?

```r
least_expensive <-  how_many_for_how_much %>%
  arrange(Avg_price) %>%
  slice(1:5)
least_expensive
```

```
## # A tibble: 5 x 4
##   rest_name            num_items_menu restaurant_id Avg_price
##   <chr>                         <int>         <dbl>     <dbl>
## 1 Metro Coffee Express             29        161225      1.10
## 2 The Bubble Spot                  11        158430      1.32
## 3 Bambusa                          12        173211      1.41
## 4 InstaGrab Pretzel                27        113358      1.59
## 5 Salad Box                       319         61215      1.64
```

```r
least_expensive %>%
ggplot(aes(reorder(rest_name, num_items_menu), num_items_menu, fill=-Avg_price)) +
  geom_bar(stat = "identity")+
  labs(x = "Name", y = "# of items", fill="avg price")+
  ggtitle("5 least expensive restaurants with relative number of items")+
  coord_flip()+
  ggsave("who_howlittle_howmany.png")
```

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

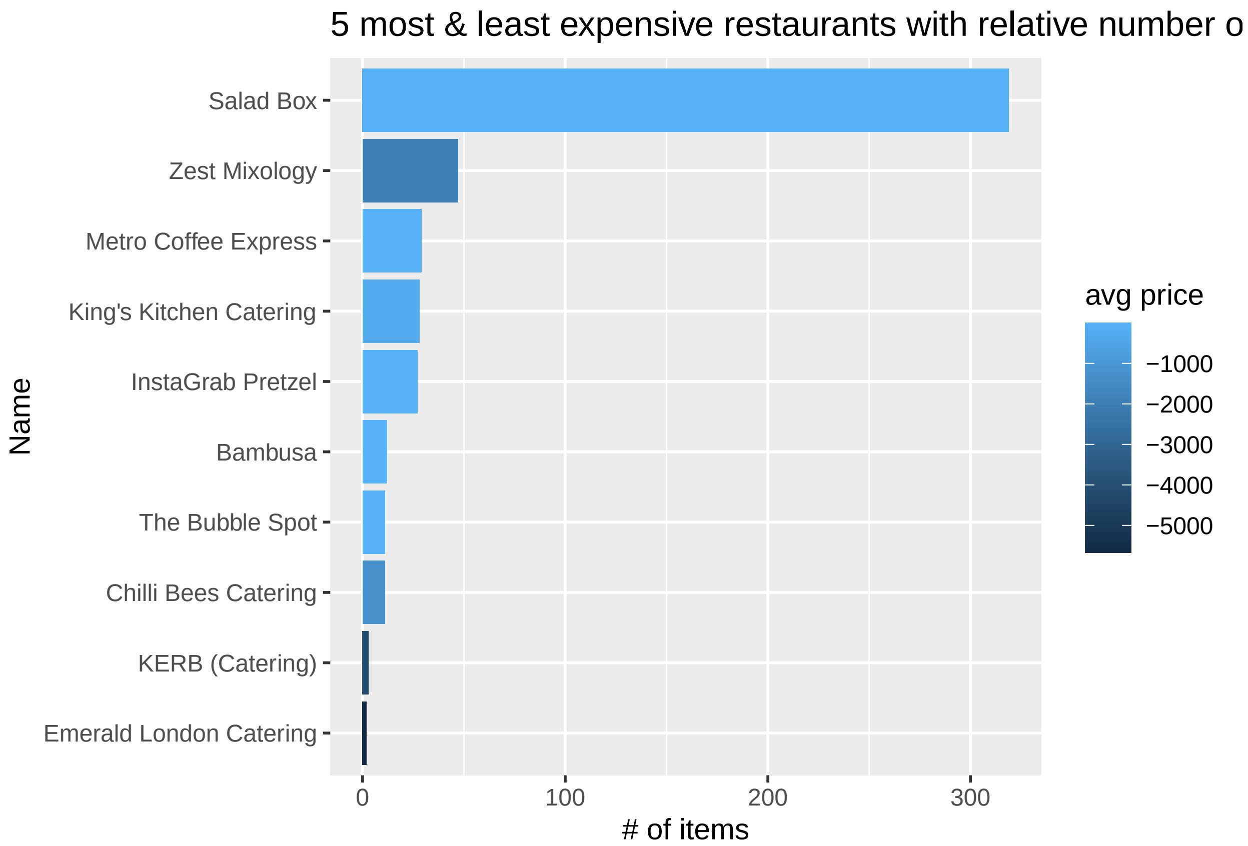


```
all_prices <- full_join(most_expensive, least_expensive)
```

```
## Joining, by = c("rest_name", "num_items_menu", "restaurant_id", "Avg_price")
```

```
all_prices %>%
ggplot(aes(reorder(rest_name, num_items_menu), num_items_menu, fill=-Avg_price)) +
  geom_bar(stat = "identity")+
  labs(x = "Name", y = "# of items", fill="avg price")+
  ggtitle("5 most & least expensive restaurants with relative number of items")+
  coord_flip()+
  ggsave("all_prices.png")
```

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



```
colnames(info_delivery)[1] <- "restaurant_id"

complete_rest_data <- left_join(info_delivery, restaurants_info, by = "restaurant_id")
glimpse(complete_rest_data)
```

## 2 Restaurants Delivery Times Analysis

```
## Rows: 152,217
## Columns: 9
## $ restaurant_id          <dbl> 98636, 167932, 902, 22555, 29850, 202819, 69871~
## $ neighborhood_name      <chr> "the-city", "the-city", "the-city", "the-city",~
## $ rest_delivery_time_min <dbl> 10, 10, 15, 15, 10, 25, 20, 10, 15, 15, 10, 15,~
## $ rest_name              <chr> "Farmer J - King William Street", "Acai Berry",~
## $ rest_brand             <chr> "Farmer J", "Acai Berry Boxpark", "La Cucina", ~
## $ rest_postcode          <chr> "EC4R9AJ", "E16GY", "E16RL", "EC2M4NQ", "E16SB"~
## $ rest_neighborhood      <chr> "The City", "Shoreditch", "Brick Lane", "The Ci~
## $ rest_rating            <dbl> 4.9, 4.8, 4.4, 4.6, 4.8, 4.4, 4.7, 4.6, 4.9, 4.~
## $ rest_menu_item_price   <list> <1.00, 9.60, 9.50, 2.00, 9.00, 10.25, 9.00, 8.~
```

2.1 How many neighborhoods does each restaurant deliver to?

```
deliveries_where <- complete_rest_data %>%
  group_by(restaurant_id) %>%
  summarise(num_place_of_delivery = n())

glimpse(deliveries_where)
```

```
## Rows: 4,240
## Columns: 2
## $ restaurant_id         <dbl> 3, 5, 8, 10, 15, 16, 18, 19, 20, 21, 23, 24, 25,~
## $ num_place_of_delivery <int> 47, 52, 52, 29, 19, 65, 80, 128, 85, 80, 80, 42,~
```

2.2 Top 15 neighborhoods for number of restaurants served by.

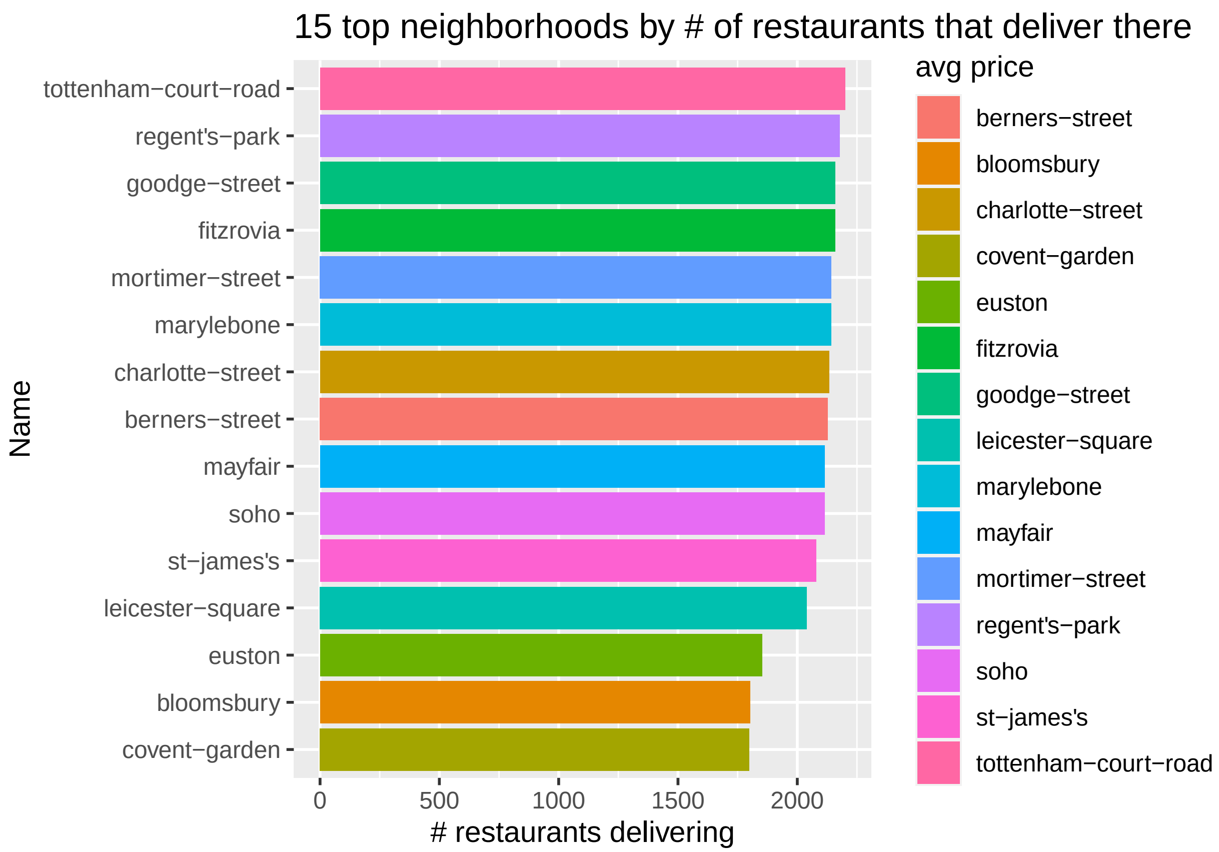
```
deliveries_by <- complete_rest_data %>%
  group_by(neighborhood_name) %>%
  summarise(how_many_deliver = n()) %>%
  arrange(-how_many_deliver) %>%
  slice(1:15)

glimpse(deliveries_by)
```

```
## Rows: 15
## Columns: 2
## $ neighborhood_name <chr> "tottenham-court-road", "regent's-park", "fitzrovia"~
## $ how_many_deliver  <int> 2201, 2177, 2158, 2158, 2143, 2142, 2133, 2128, 2115~
```

```
deliveries_by %>%
ggplot(aes(reorder(neighborhood_name, how_many_deliver), how_many_deliver, fill=neighborhood_name)) +
  geom_bar(stat = "identity")+
  labs(x = "Name", y = "# restaurants delivering", fill="avg price")+
  ggtitle("15 top neighborhoods by # of restaurants that deliver there")+
  coord_flip()+
  ggsave("most_served.png")
```

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



2.3 Average delivery time per restaurant

```
avg_del_time <- complete_rest_data %>%
  group_by(restaurant_id, rest_name, rest_postcode, rest_rating) %>%
  summarise(avg_time = mean(rest_delivery_time_min, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'restaurant_id', 'rest_name', 'rest_postcode'. You can override
```

```
avg_del_time
```

```
## # A tibble: 4,240 x 5
## # Groups:   restaurant_id, rest_name, rest_postcode [4,240]
##    restaurant_id rest_name                    rest_postcode rest_rating avg_time
##            <dbl> <chr>                        <chr>               <dbl>    <dbl>
##  1             3 "Busaba Chelsea"             SW35UZ                4.6     19.6
##  2             5 "Rossopomodoro"              SW109NB               4.5     17.1
##  3             8 "New Culture Revolution"     SW35EP                4.7     18.3
##  4            10 "Mandaloun"                  SW109TW               4.8     18.6
##  5            15 "Busaba St Christopher's Pl~ W1U1BU                4.7     21.8
##  6            16 "Busaba Bloomsbury"          WC1E7DF               4.7     22.4
##  7            18 "\U0001f1ef\U0001f1f5\U0001~ W1F0LL                4.8     20.6
##  8            19 "Noura"                      W1J5HP                4.7     23.7
##  9            20 "Dozo Sushi"                 W1D4TP                4.8     62.2
## 10            21 "Levant"                     W1U2SJ                4.8     21.8
## # ... with 4,230 more rows
```

2.4 Top 20 fastest restaurants.

```
avg_del_time$avg_time <- as.integer(avg_del_time$avg_time)

glimpse(avg_del_time)
```

```
## Rows: 4,240
## Columns: 5
## Groups: restaurant_id, rest_name, rest_postcode [4,240]
## $ restaurant_id <dbl> 3, 5, 8, 10, 15, 16, 18, 19, 20, 21, 23, 24, 25, 26, 34,~
## $ rest_name     <chr> "Busaba Chelsea", "Rossopomodoro", "New Culture Revoluti~
## $ rest_postcode <chr> "SW35UZ", "SW109NB", "SW35EP", "SW109TW", "W1U1BU", "WC1~
## $ rest_rating   <dbl> 4.6, 4.5, 4.7, 4.8, 4.7, 4.7, 4.8, 4.7, 4.8, 4.8, 4.5, 4~
## $ avg_time      <int> 19, 17, 18, 18, 21, 22, 20, 23, 62, 21, 21, 17, 13, 25, ~
```

```
fastest20 <- avg_del_time %>%
  arrange(-avg_time) %>%
  head(20)
fastest20
```

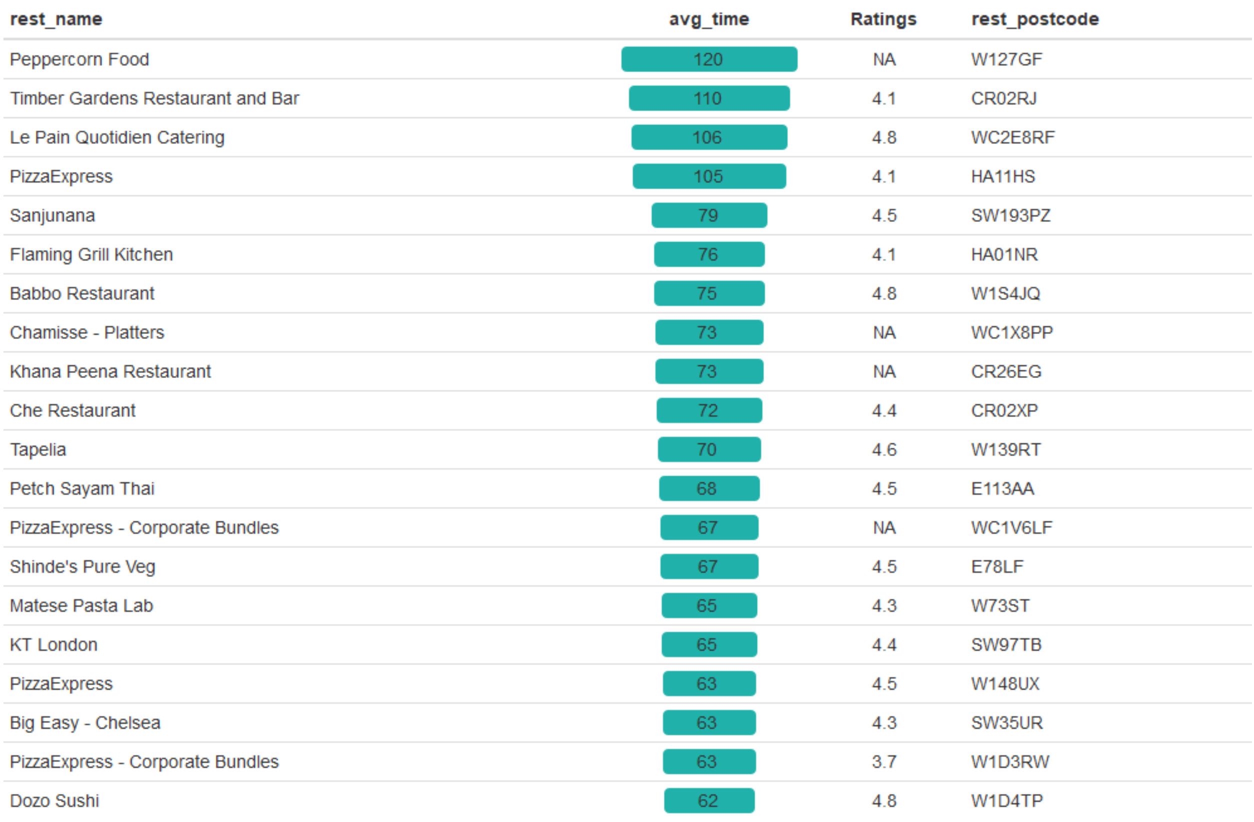
```
## # A tibble: 20 x 5
## # Groups:   restaurant_id, rest_name, rest_postcode [20]
##    restaurant_id rest_name                    rest_postcode rest_rating avg_time
##            <dbl> <chr>                        <chr>               <dbl>    <int>
##  1        186103 Peppercorn Food              W127GF               NA        120
##  2        134907 Timber Gardens Restaurant a~ CR02RJ                4.1      110
##  3        202181 Le Pain Quotidien Catering   WC2E8RF               4.8      106
##  4         27252 PizzaExpress                 HA11HS                4.1      105
##  5        107127 Sanjunana                    SW193PZ               4.5       79
##  6        113498 Flaming Grill Kitchen        HA01NR                4.1       76
##  7         22907 Babbo Restaurant             W1S4JQ                4.8       75
##  8         42727 Chamisse - Platters          WC1X8PP              NA         73
##  9        119799 Khana Peena Restaurant       CR26EG               NA         73
## 10        113471 Che Restaurant               CR02XP                4.4       72
## 11         64754 Tapelia                      W139RT                4.6       70
## 12         96972 Petch Sayam Thai             E113AA                4.5       68
## 13         68900 PizzaExpress - Corporate Bu~ WC1V6LF              NA         67
## 14         70371 Shinde’s Pure Veg            E78LF                 4.5       67
## 15        107158 Matese Pasta Lab             W73ST                 4.3       65
## 16        157210 KT London                    SW97TB                4.4       65
## 17         25142 PizzaExpress                 W148UX                4.5       63
## 18         30483 Big Easy - Chelsea           SW35UR                4.3       63
## 19         68371 PizzaExpress - Corporate Bu~ W1D3RW                3.7       63
## 20            20 Dozo Sushi                   W1D4TP                4.8       62
```

```
export_formattable <- function(f, file, width = "100%", height = NULL,
                               background = "white", delay = 0.2)
    {
      w <- as.htmlwidget(f, width = width, height = height)
      path <- html_print(w, background = background, viewer = NULL)
      url <- paste0("file:///", gsub("\\\\", "/", normalizePath(path)))
      webshot(url,
              file = file,
              selector = ".formattable_widget",
              delay = delay)
    }

f20 <- fastest20 %>%
  subset(select = -restaurant_id) %>%
  relocate(avg_time, .before = rest_postcode) %>%
  relocate(Ratings = rest_rating, .before = rest_postcode)

FT2 <- f20 %>%
  formattable(align = c("l", "c", "c", "l"), list(avg_time = color_bar(("lightseagreen"))))

export_formattable(FT2,"FT2.png")
```



| rest_name | avg_time | Ratings | rest_postcode |
|---|---|---|---|
| Peppercorn Food | 120 | NA | W127GF |
| Timber Gardens Restaurant and Bar | 110 | 4.1 | CR02RJ |
| Le Pain Quotidien Catering | 106 | 4.8 | WC2E8RF |
| PizzaExpress | 105 | 4.1 | HA11HS |
| Sanjunana | 79 | 4.5 | SW193PZ |
| Flaming Grill Kitchen | 76 | 4.1 | HA01NR |
| Babbo Restaurant | 75 | 4.8 | W1S4JQ |
| Chamisse - Platters | 73 | NA | WC1X8PP |
| Khana Peena Restaurant | 73 | NA | CR26EG |
| Che Restaurant | 72 | 4.4 | CR02XP |
| Tapelia | 70 | 4.6 | W139RT |
| Petch Sayam Thai | 68 | 4.5 | E113AA |
| PizzaExpress - Corporate Bundles | 67 | NA | WC1V6LF |
| Shinde's Pure Veg | 67 | 4.5 | E78LF |
| Matese Pasta Lab | 65 | 4.3 | W73ST |
| KT London | 65 | 4.4 | SW97TB |
| PizzaExpress | 63 | 4.5 | W148UX |
| Big Easy - Chelsea | 63 | 4.3 | SW35UR |
| PizzaExpress - Corporate Bundles | 63 | 3.7 | W1D3RW |
| Dozo Sushi | 62 | 4.8 | W1D4TP |

```
FT2
```

rest_name

avg_time

Ratings

rest_postcode

Peppercorn Food

120

NA

W127GF

Timber Gardens Restaurant and Bar

110

4.1

CR02RJ

Le Pain Quotidien Catering

106

4.8

WC2E8RF

PizzaExpress

105

4.1

HA11HS

Sanjunana

79

4.5

SW193PZ

Flaming Grill Kitchen

76

4.1

HA01NR

Babbo Restaurant

75

4.8

W1S4JQ

Chamisse - Platters

73

NA

WC1X8PP

Khana Peena Restaurant

73

NA

CR26EG

Che Restaurant

72

4.4

CR02XP

Tapelia

70

4.6

W139RT

Petch Sayam Thai

68

4.5

E113AA

PizzaExpress - Corporate Bundles

67

NA

WC1V6LF

Shinde’s Pure Veg

67

4.5

E78LF

Matese Pasta Lab

65

4.3

W73ST

KT London

65

4.4

SW97TB

PizzaExpress

63

4.5

W148UX

Big Easy - Chelsea

63

4.3

SW35UR

PizzaExpress - Corporate Bundles

63

3.7

W1D3RW

Dozo Sushi

62

4.8

W1D4TP

**3 Open Analysis** By looking at this data one question comes to my mind almost immediately. Are restaurants located in those neighborhoods that are served by the highest number of delivery places?

I start by creating a new data frame that contains the data all the data from the restaurants and their deliveries, but only for the top 10 neighborhoods for number of restaurants.

```
complete_data_fortified <- inner_join(complete_rest_data, top10_neighborhoods, by = "rest_neighborhood"
complete_data_fortified
```

```
## # A tibble: 31,978 x 10
##    restaurant_id neighborhood_name rest_delivery_tim~ rest_name      rest_brand
##            <dbl> <chr>                          <dbl> <chr>          <chr>
##  1         98636 the-city                          10 Farmer J - Ki~ Farmer J
##  2         22555 the-city                          15 Polo           Polo
##  3        108441 the-city                          10 Subway         Subway
##  4        126532 the-city                          10 Doughnut Time  <NA>
##  5         40178 the-city                          10 Le Pain Quoti~ Le Pain Qu~
##  6         65537 the-city                          15 People’s Choi~ <NA>
##  7         34430 the-city                          10 Pure - Breakf~ Pure
##  8        205253 the-city                          15 Original Bage~ <NA>
##  9         69042 the-city                          15 Yummy Bagels   <NA>
## 10         78979 the-city                          10 Nisa           Nisa
## # ... with 31,968 more rows, and 5 more variables: rest_postcode <chr>,
## #   rest_neighborhood <chr>, rest_rating <dbl>, rest_menu_item_price <list>,
## #   number_of_rest <dbl>
```

I now want to calculate how many restaurants deliver to a neighborhood and where they are from.

```
new_ranking <- complete_data_fortified %>%
  group_by(neighborhood_name, rest_neighborhood) %>%
  summarise(how_many_to_and_from = n()) %>%
  arrange(-how_many_to_and_from) %>%
  slice(1:15)
```

```
## ‘summarise()‘ has grouped output by ’neighborhood_name’. You can override using the ‘.groups‘ argumen
```

```
new_ranking
```

```
## # A tibble: 310 x 3
## # Groups:   neighborhood_name [188]
##    neighborhood_name rest_neighborhood how_many_to_and_from
##    <chr>             <chr>                            <int>
##  1 acton             Ealing                             170
##  2 aldgate           The City                           186
##  3 aldgate           Canary Wharf                         7
##  4 anerley           Croydon                             10
##  5 anerley           Tooting                              2
##  6 angel             The City                             1
##  7 balham            Tooting                            103
##  8 balham            Wimbledon                            2
##  9 bank              The City                           201
## 10 bankside          The City                             1
## # ... with 300 more rows
```

I now want to join together the previous set with how many restaurants deliver to each neighborhood (just the 15 neighborhoods with the highest number of restaurants that deliver there)

```
who_delivers <- inner_join(new_ranking, deliveries_by)
```

```
## Joining, by = "neighborhood_name"
```

```
who_delivers
```

```
## # A tibble: 40 x 4
## # Groups:   neighborhood_name [15]
##    neighborhood_name rest_neighborhood how_many_to_and_from how_many_deliver
##    <chr>             <chr>                            <int>            <int>
##  1 berners-street    Soho                               544             2128
##  2 berners-street    Marylebone                         443             2128
##  3 berners-street    The City                             2             2128
##  4 bloomsbury        Soho                               506             1802
##  5 bloomsbury        Marylebone                         168             1802
##  6 bloomsbury        The City                            19             1802
##  7 charlotte-street  Soho                               551             2133
##  8 charlotte-street  Marylebone                         421             2133
##  9 charlotte-street  The City                             2             2133
## 10 covent-garden     Soho                               526             1799
## # ... with 30 more rows
```

At this point I am interested in knowing just the number of restaurants that deliver to each neighborhood and from which neighborhood they are.

```
from_where <- who_delivers %>%
  subset(select = c("neighborhood_name", "rest_neighborhood", "how_many_to_and_from"))
from_where
```

```
## # A tibble: 40 x 3
## # Groups:   neighborhood_name [15]
##    neighborhood_name rest_neighborhood how_many_to_and_from
##    <chr>             <chr>                            <int>
##  1 berners-street    Soho                               544
##  2 berners-street    Marylebone                         443
##  3 berners-street    The City                             2
##  4 bloomsbury        Soho                               506
##  5 bloomsbury        Marylebone                         168
##  6 bloomsbury        The City                            19
##  7 charlotte-street  Soho                               551
##  8 charlotte-street  Marylebone                         421
##  9 charlotte-street  The City                             2
## 10 covent-garden     Soho                               526
## # ... with 30 more rows
```

```
how_many <- who_delivers %>%
  subset(select = c("neighborhood_name", "rest_neighborhood", "how_many_deliver"))
how_many
```

```
## # A tibble: 40 x 3
## # Groups:   neighborhood_name [15]
##    neighborhood_name rest_neighborhood how_many_deliver
##    <chr>             <chr>                        <int>
##  1 berners-street    Soho                          2128
##  2 berners-street    Marylebone                    2128
##  3 berners-street    The City                      2128
##  4 bloomsbury        Soho                          1802
##  5 bloomsbury        Marylebone                    1802
##  6 bloomsbury        The City                      1802
##  7 charlotte-street  Soho                          2133
##  8 charlotte-street  Marylebone                    2133
##  9 charlotte-street  The City                      2133
## 10 covent-garden     Soho                          1799
## # ... with 30 more rows
```

```
library(reshape)
```

```
##
## Attaching package: ’reshape’
```

```
## The following objects are masked from ’package:tidyr’:
##
##     expand, smiths
```

```
## The following object is masked from ’package:dplyr’:
##
##     rename
```

```
from_where.m <- melt(from_where)
```

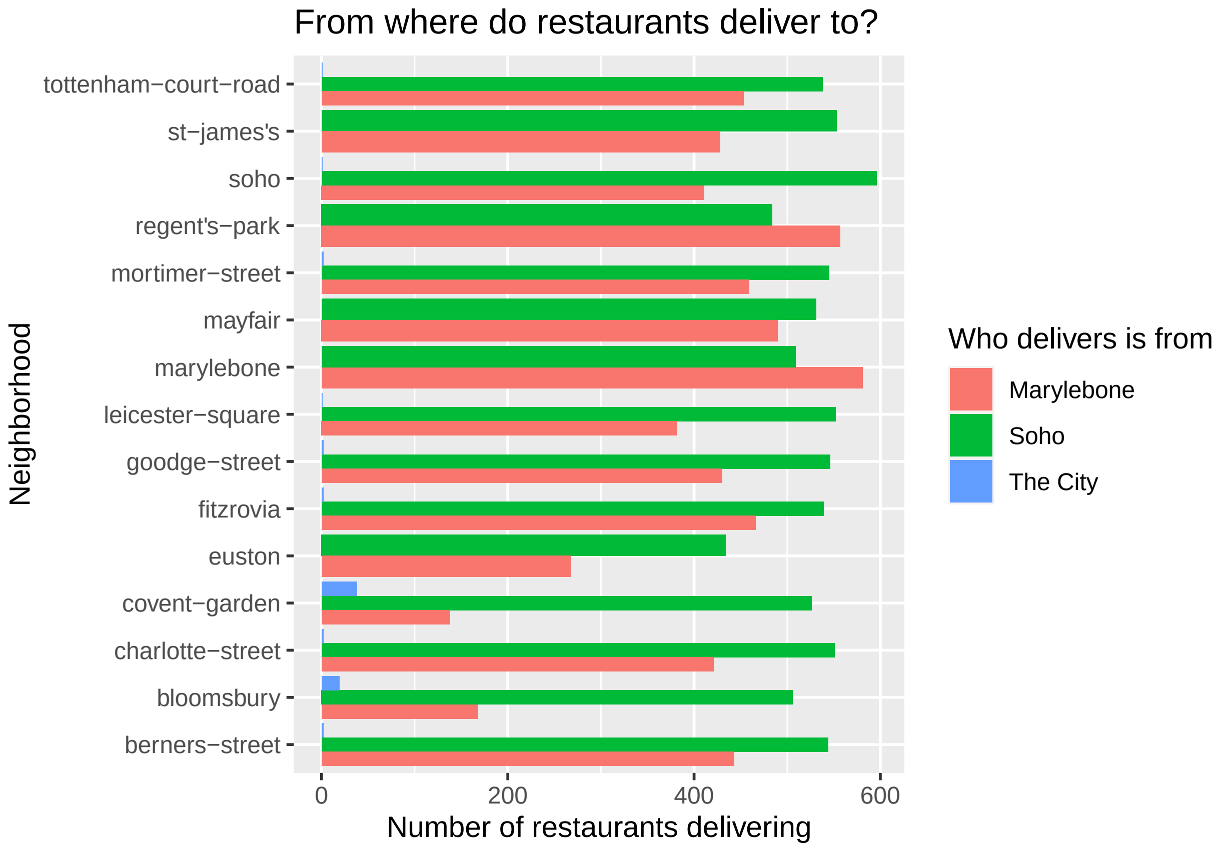
```
## Using neighborhood_name, rest_neighborhood as id variables
```

```
from_where.m
```

```
##       neighborhood_name rest_neighborhood             variable value
## 1        berners-street              Soho how_many_to_and_from   544
## 2        berners-street        Marylebone how_many_to_and_from   443
## 3        berners-street          The City how_many_to_and_from     2
## 4            bloomsbury              Soho how_many_to_and_from   506
## 5            bloomsbury        Marylebone how_many_to_and_from   168
## 6            bloomsbury          The City how_many_to_and_from    19
## 7      charlotte-street              Soho how_many_to_and_from   551
## 8      charlotte-street        Marylebone how_many_to_and_from   421
## 9      charlotte-street          The City how_many_to_and_from     2
## 10        covent-garden              Soho how_many_to_and_from   526
## 11        covent-garden        Marylebone how_many_to_and_from   138
## 12        covent-garden          The City how_many_to_and_from    38
## 13               euston              Soho how_many_to_and_from   434
## 14               euston        Marylebone how_many_to_and_from   268
## 15            fitzrovia              Soho how_many_to_and_from   539
## 16            fitzrovia        Marylebone how_many_to_and_from   466
## 17            fitzrovia          The City how_many_to_and_from     2
## 18        goodge-street              Soho how_many_to_and_from   546
## 19        goodge-street        Marylebone how_many_to_and_from   430
## 20        goodge-street          The City how_many_to_and_from     2
## 21     leicester-square              Soho how_many_to_and_from   552
## 22     leicester-square        Marylebone how_many_to_and_from   382
## 23     leicester-square          The City how_many_to_and_from     1
## 24           marylebone        Marylebone how_many_to_and_from   581
## 25           marylebone              Soho how_many_to_and_from   509
## 26              mayfair              Soho how_many_to_and_from   531
## 27              mayfair        Marylebone how_many_to_and_from   490
## 28      mortimer-street              Soho how_many_to_and_from   545
## 29      mortimer-street        Marylebone how_many_to_and_from   459
## 30      mortimer-street          The City how_many_to_and_from     2
## 31        regent’s-park        Marylebone how_many_to_and_from   557
## 32        regent’s-park              Soho how_many_to_and_from   484
## 33                 soho              Soho how_many_to_and_from   596
## 34                 soho        Marylebone how_many_to_and_from   411
## 35                 soho          The City how_many_to_and_from     1
## 36           st-james’s              Soho how_many_to_and_from   553
## 37           st-james’s        Marylebone how_many_to_and_from   428
## 38 tottenham-court-road              Soho how_many_to_and_from   538
## 39 tottenham-court-road        Marylebone how_many_to_and_from   453
## 40 tottenham-court-road          The City how_many_to_and_from     1
```

```
ggplot(from_where.m, aes(neighborhood_name, value, fill = rest_neighborhood)) +
  geom_bar(stat="identity", position = "dodge")+
  labs(y = "Number of restaurants delivering", x = "Neighborhood", fill = "Who delivers is from", title
  coord_flip()+
  ggsave("fromwhere.png")
```

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



In this graph we can see how the top 15 neighborhoods for number of restaurants served by, are segmented between the restaurants located in 3 of the top 10 neighborhood for number of restaurants.

```
how_many.m <- melt(how_many)
```

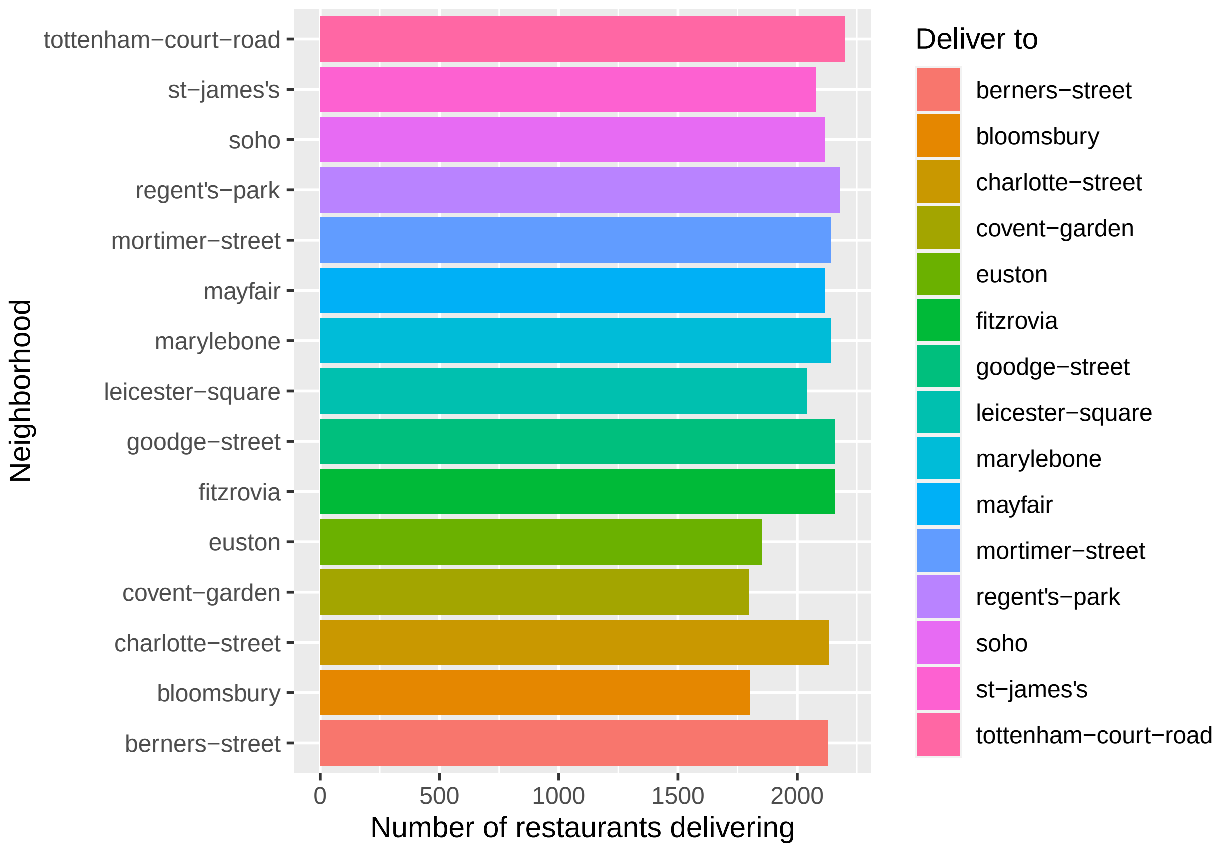
```
## Using neighborhood_name, rest_neighborhood as id variables
```

```
how_many.m
```

```
##       neighborhood_name rest_neighborhood         variable value
## 1        berners-street              Soho how_many_deliver  2128
## 2        berners-street        Marylebone how_many_deliver  2128
## 3        berners-street          The City how_many_deliver  2128
## 4            bloomsbury              Soho how_many_deliver  1802
## 5            bloomsbury        Marylebone how_many_deliver  1802
## 6            bloomsbury          The City how_many_deliver  1802
## 7      charlotte-street              Soho how_many_deliver  2133
## 8      charlotte-street        Marylebone how_many_deliver  2133
## 9      charlotte-street          The City how_many_deliver  2133
## 10        covent-garden              Soho how_many_deliver  1799
## 11        covent-garden        Marylebone how_many_deliver  1799
## 12        covent-garden          The City how_many_deliver  1799
## 13               euston              Soho how_many_deliver  1852
## 14               euston        Marylebone how_many_deliver  1852
## 15            fitzrovia              Soho how_many_deliver  2158
## 16            fitzrovia        Marylebone how_many_deliver  2158
## 17            fitzrovia          The City how_many_deliver  2158
## 18        goodge-street              Soho how_many_deliver  2158
## 19        goodge-street        Marylebone how_many_deliver  2158
## 20        goodge-street          The City how_many_deliver  2158
## 21     leicester-square              Soho how_many_deliver  2040
## 22     leicester-square        Marylebone how_many_deliver  2040
## 23     leicester-square          The City how_many_deliver  2040
## 24           marylebone        Marylebone how_many_deliver  2142
## 25           marylebone              Soho how_many_deliver  2142
## 26              mayfair              Soho how_many_deliver  2115
## 27              mayfair        Marylebone how_many_deliver  2115
## 28      mortimer-street              Soho how_many_deliver  2143
## 29      mortimer-street        Marylebone how_many_deliver  2143
## 30      mortimer-street          The City how_many_deliver  2143
## 31        regent’s-park        Marylebone how_many_deliver  2177
## 32        regent’s-park              Soho how_many_deliver  2177
## 33                 soho              Soho how_many_deliver  2114
## 34                 soho        Marylebone how_many_deliver  2114
## 35                 soho          The City how_many_deliver  2114
## 36           st-james’s              Soho how_many_deliver  2079
## 37           st-james’s        Marylebone how_many_deliver  2079
## 38 tottenham-court-road              Soho how_many_deliver  2201
## 39 tottenham-court-road        Marylebone how_many_deliver  2201
## 40 tottenham-court-road          The City how_many_deliver  2201
```

```
ggplot(how_many.m, aes(neighborhood_name, value, fill = neighborhood_name)) +
  geom_bar(stat="identity", position = "dodge")+
  labs(y = "Number of restaurants delivering", x = "Neighborhood", fill = "Deliver to")+
  coord_flip()+
  ggsave("howmany.png")
```

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



Based also on this last chart, I would then seggest restaurants that are not on the territory to maybe think of opening stores in the neighborhoods more served, as it may represent a way of, in the long run, reducing costs due to delivery and increasing their revenues as well.