Restaurants Delivery Times

Om Ajay Karthik

3/10/2021

Restaurants Delivery Times

In this section, you will analyze the restaurant's delivery times. To perform the following analysis, you will need to join the Restaurants information and Restaurants delivery time datasets using their restaurant_id and rest_key primary keys. Answer the following questions:

Count the number of neighborhoods where each restaurant delivers. (Each restaurant can deliver to multiple neighborhoods, independently from here the restaurants is physically located) Present in a bar chart the top 15 neighborhoods by the number of restaurants where restaurants make deliveries. (You will need to count the number of restaurants that deliver to each neighborhood) Compute the average delivery time for each restaurant. (Compute the average of all neighborhoods the restaurant delivers) Present in tabular format the top 20 restaurants by fasted average delivery time. In the same table, present the rating score, and postcode.

Installing necessary packages

```
#install.packages('ggplot2')
#install.packages('dplyr')
#install.packages("tidyverse")

#install.packages("xlsx")
#install.packages('expss')
#install.packages('gridExtra')
#install.packages('purrrlyr')
#install.packages('ggpubr')
#install.packages('data.table')
```

Importing necessary libraries

```
library(ggpubr)
```

```
## Loading required package: ggplot2
```

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.0.6     v purrr   0.3.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x gridExtra::combine() masks dplyr::combine()
## x dplyr::filter()      masks stats::filter()
## x dplyr::lag()         masks stats::lag()
```

```
library(purrrlyr)
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

Imporing datasets

```
resturants_mibe <- readRDS("C:/Users/Om Ajay Karthik/Desktop/LSU/resturants-mibe.rds")
delivery_mibe <- readRDS("C:/Users/Om Ajay Karthik/Desktop/LSU/delivery-mibe.rds")
```

Right outer join table and create a duplicate dataframe

```
resturant_delivery = merge(x = resturants_mibe, y = delivery_mibe, by.x='restaurant_id', by.y='r
est_key', all.y = TRUE)

rest_delivery_merged_subset = resturant_delivery %>% select(restaurant_id,rest_name,rest_neighbo
rhood,rest_postcode,neighborhood_name,rest_rating,rest_delivery_time_min)
```

1. Count the number of neighborhoods where each restaurant delivers. (Each restaurant can deliver to multiple neighborhoods, independently from here the restaurants is physically located)

```
neighbourhoods_delivery_count = rest_delivery_merged_subset %>%
  select(rest_name,neighborhood_name)   %>%
  group_by(rest_name,neighborhood_name) %>%
  unique() %>%
  summarise(n = n()) %>% count()
```

```
## `summarise()` has grouped output by 'rest_name'. You can override using the `.groups` argumen
t.
```

```
names(neighbourhoods_delivery_count)[1] <- "Restaurant Name"
names(neighbourhoods_delivery_count)[2] <- "No of Neighbourhood Delivery by Resturant"
```

```
writeLines("td, th { padding : 6px } th { background-color : brown ; color : white; border : 1px
solid white; } td { color : brown ; border : 1px solid brown }", con = "mystyle.css")
dset1 <- head(neighbourhoods_delivery_count,n=10)
knitr::kable(dset1, format = "html")
```

Restaurant Name	No of Neighbourhood Delivery by Resturant
&Cake – Wembley	6
<U+0001F1EF><U+0001F1F5><U+0001F1E7><U+0001F1F7> YOOBI <U+0001F363>	16
<U+0001F336> BOMBAY BICYCLE CHEF <U+0001F336>	16
<U+0001F3C4> Honi Poke - Mayfair<U+0001F3C4>	31
<U+0001F3C4> Honi Poke - Soho<U+0001F3C4>	10
<U+0001F451> Benito’s Hat <U+0001F451>	18
<U+5927><U+534E><U+5C0F><U+5403> Da Hua Street Food	5
<U+5927><U+534E><U+5FEB><U+9910>(Dahua Fast Food)	5
<U+5A46><U+5A46> Popo Chinese, Oriental, Thai & Dim Sum Cuisine	5
<U+6771><U+5712><U+5C0F><U+9928> - East Garden	2

2. Present in a bar chart the top 15 neighborhoods by the number of restaurants where restaurants make deliveries.(You will need to count the number of restaurants that deliver to each

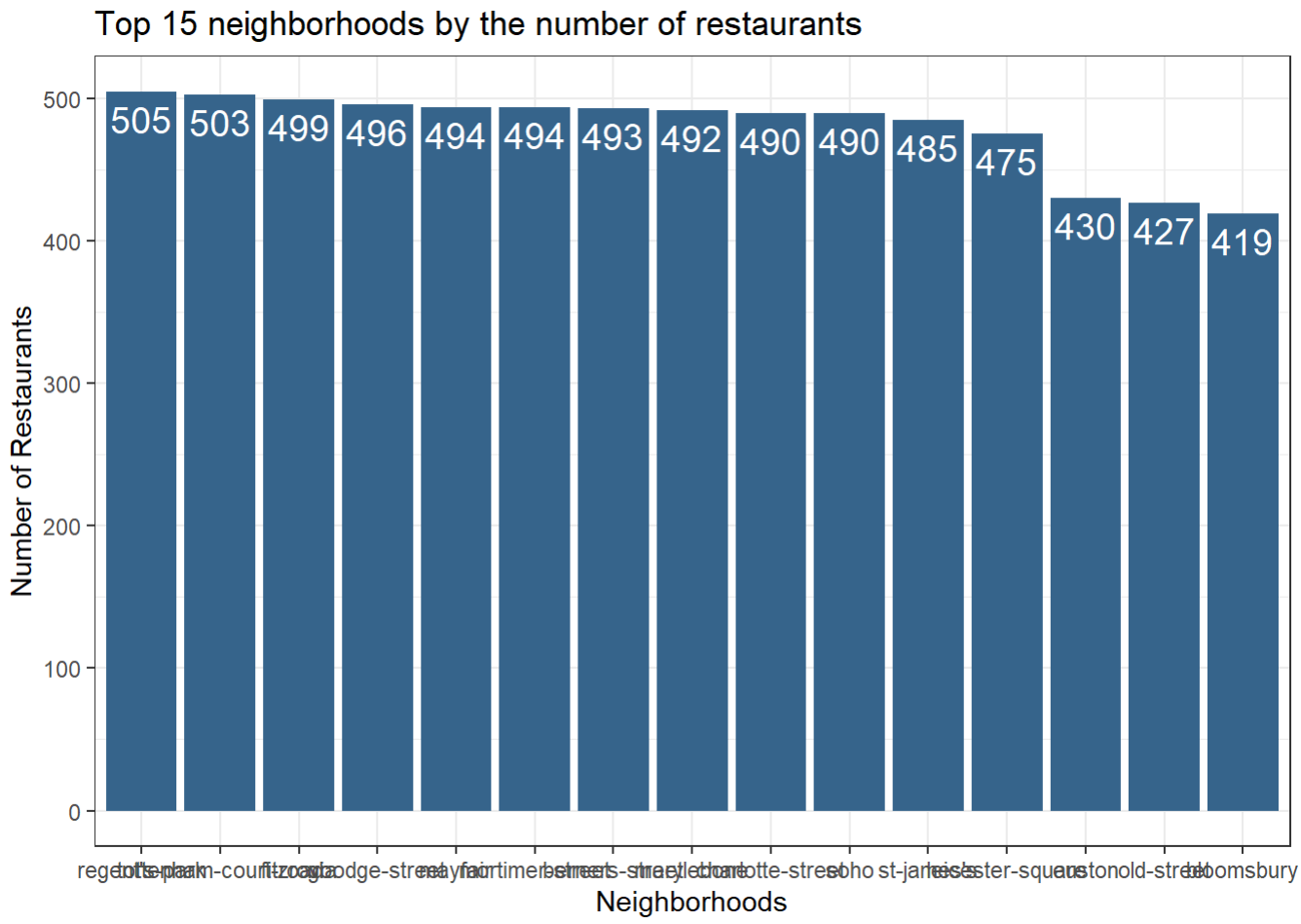
neighborhood)

```
restaurants_per_neighbourhood = rest_delivery_merged_subset %>%
  select(rest_name,neighborhood_name)   %>%
  group_by(neighborhood_name,rest_name) %>%
  unique() %>%
  summarise(n = n()) %>%
  count()%>%
  arrange(-n)
```

```
## `summarise()` has grouped output by 'neighborhood_name'. You can override using the `.groups` argument.
```

```
names(restaurants_per_neighbourhood)[2] <- "No_of_Restaurants"

top_n_restaurants_per_neighbourhood = head(restaurants_per_neighbourhood,n=15)
ggplot(top_n_restaurants_per_neighbourhood, aes(x = reorder(neighborhood_name, -No_of_Restaurant
s), y = No_of_Restaurants)) +
  geom_bar(stat="identity", fill="steelblue4")+
  geom_text(aes(label=No_of_Restaurants), vjust=1.5, color="white", size=5)+
  labs(title = "Top 15 neighborhoods by the number of restaurants") +
  xlab("Neighborhoods") +
  ylab("Number of Restaurants") +
  theme_bw()
```



3. Compute the average delivery time for each restaurant. (Compute the average of all neighborhoods the restaurant delivers)

```
resturant_avg_delivery_time = rest_delivery_merged_subset %>%
  select(rest_name,neighborhood_name,rest_delivery_time_min)   %>%
  filter(!is.na(rest_delivery_time_min)) %>%
  group_by(rest_name) %>%
  summarise(average_delivery_time = mean(rest_delivery_time_min))
```

```
writeLines("td, th { padding : 6px } th { background-color : brown ; color : white; border : 1px
solid white; } td { color : brown ; border : 1px solid brown }", con = "mystyle.css")
dset1 <- head(resturant_avg_delivery_time,n=50)
knitr::kable(dset1, format = "html")
```

rest_name	average_delivery_time
&Cake – Wembley	22.77778
<U+0001F1EF><U+0001F1F5><U+0001F1E7><U+0001F1F7> YOOBI <U+0001F363>	20.62500
<U+0001F336> BOMBAY BICYCLE CHEF <U+0001F336>	36.91176
<U+0001F3C4> Honi Poke - Mayfair<U+0001F3C4>	21.07143
<U+0001F3C4> Honi Poke - Soho<U+0001F3C4>	11.62500
<U+0001F451> Benito’s Hat <U+0001F451>	24.22535
<U+5927><U+534E><U+5C0F><U+5403> Da Hua Street Food	25.00000
<U+5927><U+534E><U+5FEB><U+9910>(Dahua Fast Food)	25.00000
<U+5A46><U+5A46> Popo Chinese, Oriental, Thai & Dim Sum Cuisine	22.66667
<U+6771><U+5712><U+5C0F><U+9928> - East Garden	21.66667
<U+7C73>Kome Japanese & Korean Sushi	18.51562
<U+98DF><U+5168><U+5C0F><U+53A8> FODAL Oriental Kitchen	17.50000
11 Coffee and Co	27.14286
14228 - Burger King	21.87500
2 Girls’ Cafe	21.58537
201 Asian Kitchen	27.50000
24/7 Supermarket	16.00000
36 Streets Food	27.36111
4021 Firezza	15.00000
491094Adam’s Cafe	28.23529
5 Star Pizza	23.25000
500 Degrees Pizzeria	17.33333
696 mangal	16.83333
805 Restaurant	30.00000
A’La Pizza	26.17647
A Burgers by Taster	23.43750
A K Chicken	21.11111
A&Y Wines Beers	13.98438
A.D Food & Wine	16.18750
A1 Kebabish	21.66667
Aachis Masala Chennai	25.00000
Abbotshill Wine Bar & Deli	16.66667

rest_name	average_delivery_time
Abby’s Food and Wine	23.57143
Abshar	29.16667
Abu Zaad	25.00000
Abu Zaad - Ealing	20.00000
Acacus	39.00000
Acai Berry	12.30159
Açaí Sisters	31.56716
Adam’s Ethiopian Restaurant	51.94444
Adana	22.26190
Ading’s Kitchen	31.56250
AFC Fried Chicken & Pizza	31.30952
Afters	20.95238
Afters Original	21.66667
Aglio e Olio	31.25000
Agora Greek Bakery	16.62500
Aha Wok	23.12500
Ahaa Dosa	25.29412
Ahi Poké	12.68657

4. Present in tabular format the top 20 restaurants by fasted average delivery time.In the same table, present the rating score, and postcode.

```
resturant_avg_delivery_time_rating_postcode = rest_delivery_merged_subset %>%
select(rest_name,neighborhood_name,rest_rating,rest_delivery_time_min,rest_postcode)   %>%
group_by(rest_name) %>%
unique() %>%
summarise(rest_name,neighborhood_name,rest_rating,rest_delivery_time_min,rest_postcode,average_d
elivery_time = mean(rest_delivery_time_min))
```

```
## `summarise()` has grouped output by 'rest_name'. You can override using the `.groups` argumen
t.
```

```
resturant_avg_delivery_time_rating_postcode_desc <-resturant_avg_delivery_time_rating_postcode[o
rder(resturant_avg_delivery_time_rating_postcode$average_delivery_time),]
resturant_avg_delivery_time_rating_postcode_desc_top_n = resturant_avg_delivery_time_rating_post
code_desc %>% select(rest_name,rest_postcode,neighborhood_name,rest_rating,average_delivery_tim
e) %>% head(., n=20)
```

```
writeLines("td, th { padding : 6px } th { background-color : brown ; color : white; border : 1px
solid white; } td { color : brown ; border : 1px solid brown }", con = "mystyle.css")
dset1 <- head(resturant_avg_delivery_time_rating_postcode_desc_top_n,n=50)
knitr::kable(dset1, format = "html")
```

rest_name	rest_postcode	neighborhood_name	rest_rating	average_delivery_time
Vagabond Wines - Fulham	SW61AX	fulham	4.8	8.000000
Vagabond Wines - Fulham	SW61AX	parsons-green	4.8	8.000000

rest_name	rest_postcode	neighborhood_name	rest_rating	average_delivery_time
Vagabond Wines - Fulham	SW61AX	earl’s-court	4.8	8.000000
Vagabond Wines - Fulham	SW61AX	west-kensington	4.8	8.000000
Vagabond Wines - Fulham	SW61AX	south-kensington	4.8	8.000000
OREE	SW109PZ	knightsbridge	4.8	8.333333
OREE	SW109PZ	south-kensington	4.8	8.333333
OREE	SW109PZ	chelsea	4.8	8.333333
OREE	SW109PZ	parsons-green	4.8	8.333333
OREE	SW109PZ	belgravia	4.8	8.333333
OREE	SW109PZ	fulham	4.8	8.333333
Amorino - Old Compton St	W1D6HF	st-james’s	4.8	10.000000
Amorino - Old Compton St	W1D6HF	leicester-square	4.8	10.000000
Amorino - Old Compton St	W1D6HF	soho	4.8	10.000000
Amorino - Old Compton St	W1D6HF	westminster	4.8	10.000000
Amorino - Old Compton St	W1D6HF	mayfair	4.8	10.000000
Andys Quality Fish & Chips	W139RJ	ealing	4.6	10.000000
Bill’s Breakfast	W52XA	ealing	4.6	10.000000
Brew Café	SW195DX	wimbledon	4.5	10.000000
Chic-o-land Kebab	KT12AA	molesey	NA	10.000000