Zomato Dataset Analysis

Vinesh Kumar Gande

2023-02-03

Dataset link: Zomato Dataset. (2022, December 23). Kaggle. https://www.kaggle.com/datasets/rajeshrampure/zomato-dataset
Description of Dataset: The Zomato dataset is imported from Kaggle, this has listing of Bengaluru City Restaurants that have Zomato accounts. We have more than 50000 rows and 17 columns in this dataset, making it pretty sizable.
For the analysis purpose, i have considered 8 columns, they are,
1.Restaurant Name: This contains the name of each restaurant
2.Accepting_Online_Order: This has "yes" or "no", stating the information whether the restaurant accepts online orders or not.
3.Facility_To_Book_Table: This has "yes" or "no", stating the information whether the restaurant accepts table booking or not.
4.Customer_Rating: This has the information about the rating given by the customers which is the average of the all the voter surveys.
5.Number_Of_Votes: The number of people voted for a given restaurant.
6.Restaurant_Location: The location at which the restaurant is located at.
7.Restaurant_Type: This lists the type of the restaurant like Casual Dining, Cafe, Quick Bites,Delivery.
8.Cost_For_Two_People: the average cost for two people that costs for a restaurant.

Preprocessing Steps:
1.Selected the features/variables required for the analysis and interpreting the business problem.
2.Renamed the column names.
3.Checked for the nulls in each column.
4.Observed few special characters in the Restaurant Name column and removed the special characters using iconv function.
5.Identified duplicate values in the columns, then, excluded from the dataframe.
6.Cleaned the Customer Rating column by removing the duplicates, nulls and the special characters.
7.Typecasted the datatype of the each column to their respective type.
8.For the remaining column, if there are duplicates and nulls are present, those were removed from the dataframe.

```
#Loading required libraries
library(readr)
library(dplyr)

data<-read_csv("D:/OneDrive - Northeastern University/NEU/IDMP/HW2/zomato.csv", col_names= FALSE,
               show_col_types=FALSE)
#Loading data from the local library
#problems(data)
```

```r
#Choosing the required columns needed for representing the business questions
#location and listed_in(city) columns has the same data, so selecting only
#location column
df_r<-select(data,Restaurant_Name=X3,Accepting_Online_Order=X4,
             Facility_To_Book_Table=X5,Customer_Rating=X6,
             Number_Of_Votes=X7,Restaurant_Location=X9,
             Restaurant_Type=X10,
             Cost_For_Two_People=X13)
#head(df_r,n=10)
df_r<-df_r[-1,]
# df_r<-select(data,Restaurant_Name=name,Accepting_Online_Order=online_order,
#              Facility_To_Book_Table=book_table,Customer_Rating=rate,
#              Number_Of_Votes=votes,Restaurant_Location=location,
#              Restaurant_Type=rest_type,
#              Cost_For_Two_People='approx_cost(for two people)')
lapply(df_r,function(x) { length(which(is.na(x)))})
```

```
## $Restaurant_Name
## [1] 14
##
## $Accepting_Online_Order
## [1] 16
##
## $Facility_To_Book_Table
## [1] 56
##
## $Customer_Rating
## [1] 7837
##
## $Number_Of_Votes
## [1] 76
##
## $Restaurant_Location
## [1] 122
##
## $Restaurant_Type
## [1] 336
##
## $Cost_For_Two_People
## [1] 505
```

```r
#checking nulls across the columns
df_c<-df_r #duplicating the dataframe
head(df_c,n=10)
```

```
## # A tibble: 10 x 8
##    Restaurant_Name       Accep~1 Facil~2 Custo~3 Numbe~4 Resta~5 Resta~6 Cost_~7
##    <chr>                 <chr>   <chr>   <chr>   <chr>   <chr>   <chr>   <chr>
##  1 "Jalsa"               Yes     "Yes"   "4.1/5" "775"   Banash~ Casual~ "800"
##  2 "Spice Elephant"      Yes     "No"    "4.1/5" "787"   Banash~ Casual~ "800"
##  3 "San Churro Cafe"     Yes     "No"    "3.8/5" "918"   Banash~ Cafe, ~ "800"
##  4 "Addhuri Udupi Bhoja~ No      "No"    "3.7/5" "88"    Banash~ Quick ~ "300"
##  5 "Grand Village"       No      "No"    "3.8/5" "166"   Basava~ Casual~ "600"
```

```
##  6 "Timepass Dinner"     Yes     "No"    "3.8/5" "286"   Basava~ Casual~ "600"
##  7 "Rosewood Internatio~ No      "No"    "3.6/5" "8"     Mysore~ Casual~ "800"
##  8 "Onesta"              Yes     "Yes"   "4.6/5" "2556"  Banash~ Casual~ "600"
##  9 "Penthouse Cafe"      Yes     "No"    "4.0/5" "324"   Banash~ Cafe    "700"
## 10 "service was sluggis~ ('Rate~ "'RATE~ "\\nTo~ "no ou~ a nice~ ('Rate~ "it tu~
## # ... with abbreviated variable names 1: Accepting_Online_Order,
## #   2: Facility_To_Book_Table, 3: Customer_Rating, 4: Number_Of_Votes,
## #   5: Restaurant_Location, 6: Restaurant_Type, 7: Cost_For_Two_People
```

#cleaning Restaurant_Name column

```
df_c$Restaurant_Name <- iconv(df_c$Restaurant_Name, "UTF-8", "ASCII", sub = "")

# Identify duplicate values based on two or more columns
duplicates <- duplicated(df_c[,c("Restaurant_Name")])

# Remove duplicates based on two or more columns
df_unique <- df_c[!duplicates, ]

# Drop the remaining row accordingly based on two or more columns
df_c <- df_unique[!duplicated(df_unique[,c("Restaurant_Name")]), ]
dim(df_c)
```

```
## [1] 11880     8
```

```
lapply(df_c,function(x) { length(which(is.na(x)))})
```

```
## $Restaurant_Name
## [1] 1
##
## $Accepting_Online_Order
## [1] 3
##
## $Facility_To_Book_Table
## [1] 10
##
## $Customer_Rating
## [1] 1880
##
## $Number_Of_Votes
## [1] 25
##
## $Restaurant_Location
## [1] 49
##
## $Restaurant_Type
## [1] 93
##
## $Cost_For_Two_People
## [1] 110
```

```
#checking nulls across the columns
```

#Cleaning Customer_Rating column as it has more nulls when compared to other columns

```
df_c$Customer_Rating <- sub("/5", "", df_c$Customer_Rating)
#replacing the /5 with empty string, observed the dataset through excel and
#decided to trim the rating from a/b to a
df_c <- df_c[!grepl("rated", df_c$Customer_Rating,ignore.case = TRUE), ]
#observed there are multiple unwanted rows which mixed up with invalid data that
#is reviews data is mixed, so cleaning the rows containing rated
df_c$Customer_Rating <- gsub("\\s", "", df_c$Customer_Rating)
#observed few column data being repeated due to white spaces in the string
#(like "1.8" and "1.8 " been identified as separate values), so removed white
#spaces
df_c<-transform(
  df_c,Customer_Rating= ifelse(nchar(Customer_Rating)>3 | is.na(Customer_Rating)
                               | Customer_Rating=="NEW"  |
                                 Customer_Rating=="etc"| Customer_Rating=="BTM"|
                                 Customer_Rating=="dal" |
                                 Customer_Rating=="-", -1, Customer_Rating))
#Removing the rows having the data other than rating
# df_c
df_c <- df_c[grepl("[^''!@#$%^&*()]+", df_c$Customer_Rating), ] #Removing the
#rows having special characters other than rating
df_c$Customer_Rating <- gsub("[\\[\\]]",-1, df_c$Customer_Rating)

#lapply(df_c,function(x) { length(which(is.na(x)))})

df_c$Customer_Rating<-as.numeric(df_c$Customer_Rating)
# str(df_c)
```

#Cleaning the Accepting_Online_Order column

```
library(stringr)

#removing rows which are not matching with string yes or no(As this column
#requires yes or no, so cleaning apart from boolean values)
df_c<-transform(
  df_c, Accepting_Online_Order= ifelse(tolower(Accepting_Online_Order)=="yes"|
                                         tolower(Accepting_Online_Order)=="no",
                                       str_to_title(
                                         tolower(Accepting_Online_Order)),
                                       "null"))

df_c<- df_c[!(df_c$Accepting_Online_Order == "null"), ]
#To remove values having null as value(as there are only 16 rows out of 56k rows)
df_c<- df_c[!(is.na(df_c$Accepting_Online_Order)), ]
#to remove null rows from the dataframe

# unique(df_c$Accepting_Online_Order)
#lapply(df_c,function(x) { length(which(is.na(x)))})
df_c$Number_Of_Votes<-as.numeric(df_c$Number_Of_Votes)
```

#Cleaning Restaurant_Location column

```
#Removing Nulls from Restaurant Location column as there are negligible rows
#that is 4 out of 11k
df_c<- df_c[!(is.na(df_c$Restaurant_Location)), ]
```

#cleaning restaurant type column

```
#As the restaurant type column data has different categories clubbed together,
#so replacing nulls with "other" category doesn't make any sense, so removing
#nulls from restaurant type data.
df_c<- df_c[!(is.na(df_c$Restaurant_Type)), ]
lapply(df_c,function(x) { length(which(is.na(x)))})
```

```
## $Restaurant_Name
## [1] 0
##
## $Accepting_Online_Order
## [1] 0
##
## $Facility_To_Book_Table
## [1] 0
##
## $Customer_Rating
## [1] 0
##
## $Number_Of_Votes
## [1] 0
##
## $Restaurant_Location
## [1] 0
##
## $Restaurant_Type
## [1] 0
##
## $Cost_For_Two_People
## [1] 36
```

```
#to check nulls values column wise
```

#data type change for cost for 2 people column

```
df_c$Cost_For_Two_People <- gsub(",","", df_c$Cost_For_Two_People)
#Removing commas from the column values
df_c$Cost_For_Two_People<-as.numeric(df_c$Cost_For_Two_People)
#changing datatype to numeric from char
str(df_c) #Verifying the data type change
```

```
## 'data.frame':    8721 obs. of  8 variables:
##  $ Restaurant_Name       : chr  "Jalsa" "Spice Elephant" "San Churro Cafe" "Addhuri Udupi Bhojana" .
##  $ Accepting_Online_Order: chr  "Yes" "Yes" "Yes" "No" ...
##  $ Facility_To_Book_Table: chr  "Yes" "No" "No" "No" ...
```

```
##  $ Customer_Rating       : num  4.1 4.1 3.8 3.7 3.8 3.8 3.6 4.6 4 4.2 ...
##  $ Number_Of_Votes       : num  775 787 918 88 166 ...
##  $ Restaurant_Location   : chr  "Banashankari" "Banashankari" "Banashankari" "Banashankari" ...
##  $ Restaurant_Type       : chr  "Casual Dining" "Casual Dining" "Cafe, Casual Dining" "Quick Bites"
##  $ Cost_For_Two_People   : num  800 800 800 300 600 600 800 600 700 550 ...
```

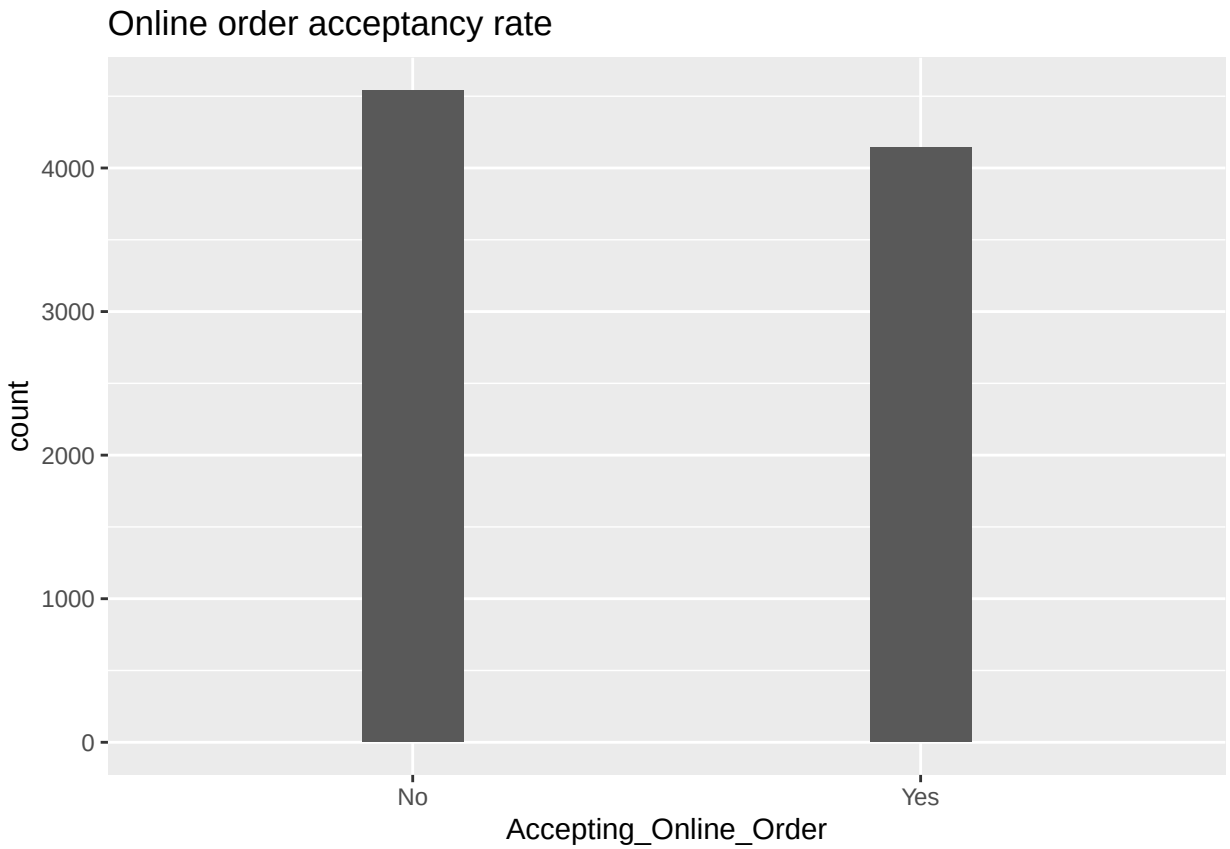
```
df_c<- df_c[!(is.na(df_c$Cost_For_Two_People)), ]
lapply(df_c,function(x) { length(which(is.na(x)))}) #to check nulls values
```

```
## $Restaurant_Name
## [1] 0
##
## $Accepting_Online_Order
## [1] 0
##
## $Facility_To_Book_Table
## [1] 0
##
## $Customer_Rating
## [1] 0
##
## $Number_Of_Votes
## [1] 0
##
## $Restaurant_Location
## [1] 0
##
## $Restaurant_Type
## [1] 0
##
## $Cost_For_Two_People
## [1] 0
```

```
#column wise to verify if every column is free from nulls
```

#(Visualisation)

```
library(ggplot2)
ggplot(df_c,aes(x=Accepting_Online_Order))+geom_bar(width = 0.2)+
  ggtitle("Online order acceptancy rate")
```

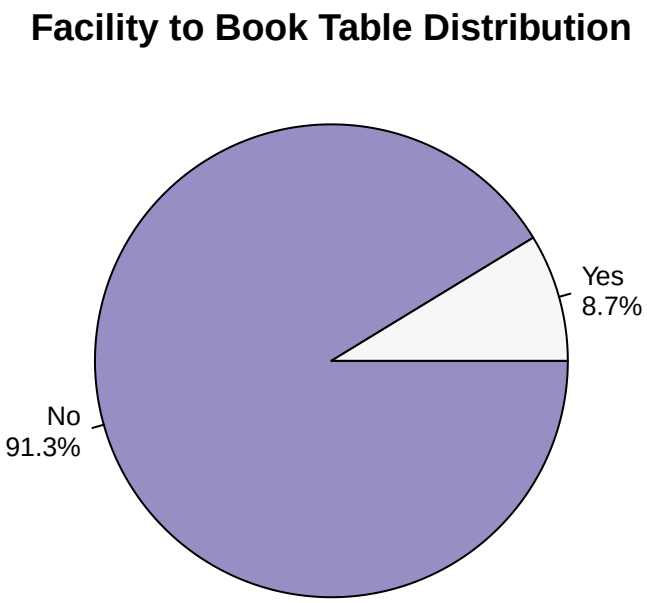


Observations:
Among 8685 Restaurants, approximately 59% of the restaurants does not accept online orders. However, there is a negligible difference of 6.67% between the restaurants those who accept the online orders. So, there is a lot of scope for Zomato to expand and convert that "No" rate to "Yes".

```
colors <- rev(RColorBrewer::brewer.pal(name="PuOr",n=3))
yn <- table(df_c$Facility_To_Book_Table)

angles <- cumsum(yn) / sum(yn) * 2 * pi
angles <- append(angles, angles[1])
# Plot the pie chart
percentages <- round(100 * yn / sum(yn), 1)
labels <- paste(names(yn), "\n", percentages, "%", sep="")
pie(yn, labels = labels, radius=1, col=colors,
    main="Facility to Book Table Distribution", cex=0.8, font=13, init.angle=0,
    clockwise=TRUE, density=NULL)

for (i in 1:length(yn)) {
  x <- 1.5 + cos(angles[i] + angles[i+1] / 2) * 1.7
  y <- 1.5 + sin(angles[i] + angles[i+1] / 2) * 1.7
  label <- paste(names(yn)[i], "\n", round(100 * yn[i] / sum(yn), 1), "%",
                 sep="")
  text(x, y, label, cex=0.8)
}
```



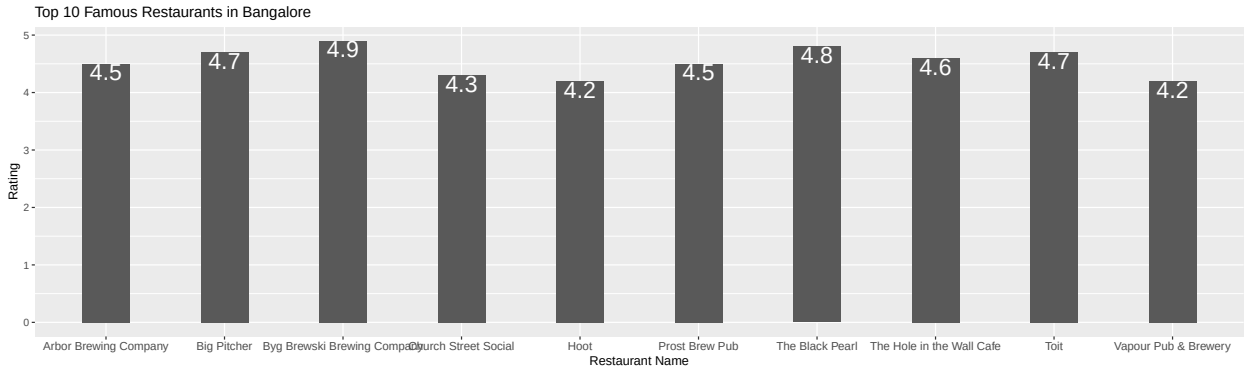
Observations:
Only 9 percent of the restaurants are facilitating to book a table online, whereas majority of the restaurants are not offering to book a table. The Online delivery app Zomato can make a better plan of action to expand their coverage on online bookings. So that the orders at restaurants can be balanced by both online and offline.

#Top 10 famous Restaurants

```
library(ggplot2)

Rest_sorted<-df_c[order(-df_c$Number_Of_Votes,-df_c$Customer_Rating),]
top10 <- head(Rest_sorted, 10)
ggplot(top10, aes(x = as.factor(Restaurant_Name), y = Customer_Rating)) +
  geom_bar(stat = "summary",width=0.4)+
ggtitle("Top 10 Famous Restaurants in Bangalore")+
  labs(x="Restaurant Name",y="Rating",size=121)+
  geom_text(label=top10$Customer_Rating,size=7,vjust=1,color="white")+
  theme(axis.text.x = element_text(size=10))
```

```
## No summary function supplied, defaulting to `mean_se()`
```



Observations:
The above bar graph gives information about the top 10 Restaurants in the city, Bangalore based on user rating and ordered by number of votes. On an average, the Restaurant above 4.2 rating is listed in the top 10.

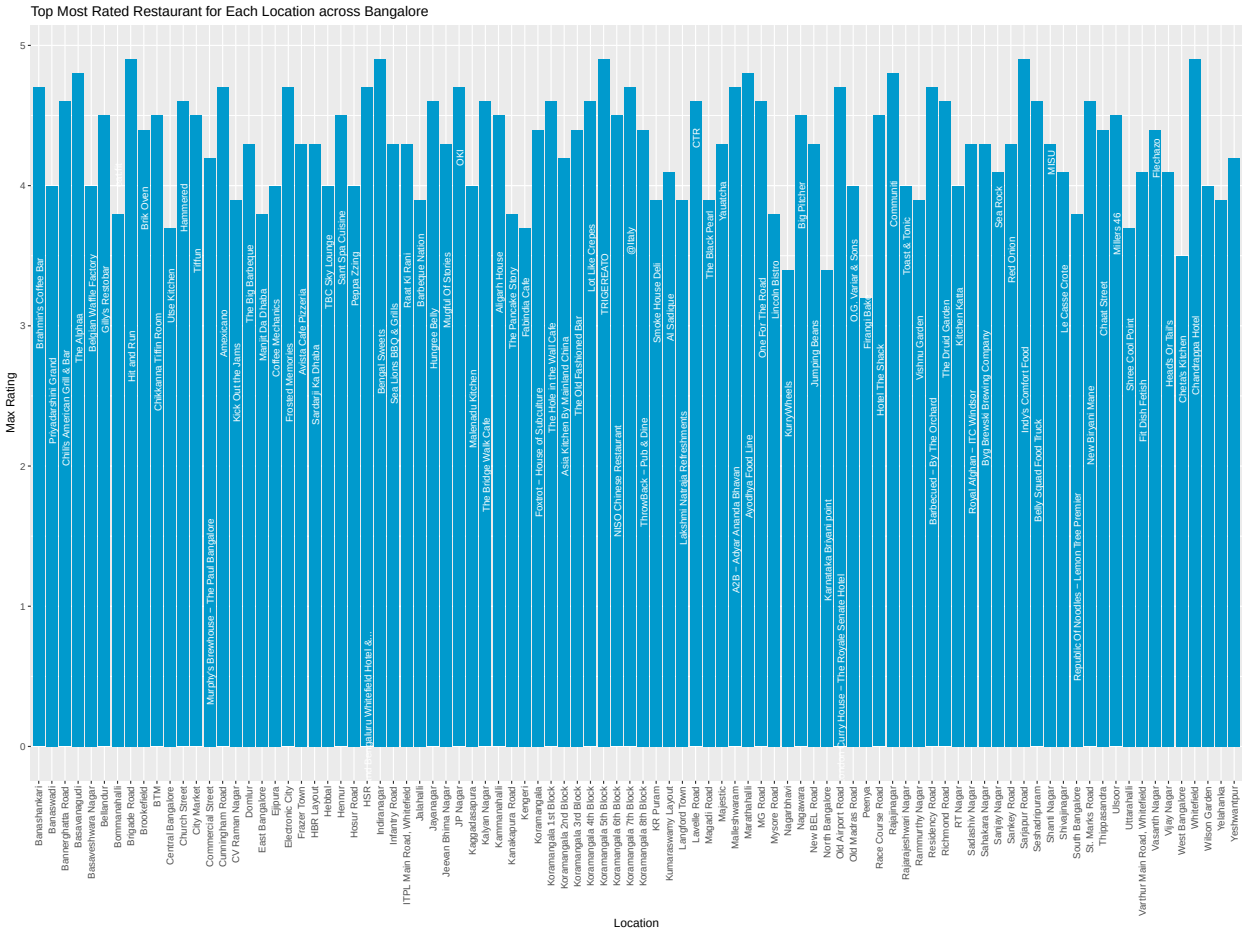
```
#Correlation between Customer Rating and the Cost for Two People
df_cc<-df_c[(df_c$Customer_Rating!=-1),]
ggplot(df_cc,aes(x=Customer_Rating,y=Cost_For_Two_People))+geom_point()+
  geom_smooth(method=lm,color="darkred",fill="blue")+
  ggtitle("Impact of Customer Rating on the Price")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Observations:
There is a positive correlation between the Customer rating and the Cost. Most of the data points lie in between 3.6-4.6 rating. Restaurants with low rating are not charging high amount compared to those with high rating.

```
# Load the required libraries
library(ggplot2)
library(dplyr)
dflr <- df_c[order(df_c$Number_Of_Votes, decreasing = TRUE),]
dflr<-dflr[!(dflr$Customer_Rating == -1),]
# Group the data frame by location and calculate the max of the rating column
df_grouped <- dflr %>%
  group_by(Restaurant_Location) %>%
  summarize(max_rating = max(Customer_Rating),
            top_rated_restaurant = Restaurant_Name[which.max(Customer_Rating)])

# Plot the graph
ggplot(df_grouped, aes(x = Restaurant_Location, y = max_rating,
                       label = top_rated_restaurant)) +
  geom_bar(stat = "identity", fill = "deepskyblue3") +
  geom_text(vjust = -5,hjust=3, color = "white",angle=90,size=3) +
  xlab("Location") +
  ylab("Max Rating")+
  ggtitle("Top Most Rated Restaurant for Each Location across Bangalore") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```


Top Most Rated Restaurant for Each Location across Bangalore
Max Rating
Location
0
1
2
3
4
5
Brahmin's Coffee Bar
Priyadarshini Grand
Chili's American Grill & Bar
The Alpha
Bohlan Waffle Factory
Glily's Restobar
Hi Land Run
Brik Oven
Brookefield
Chikkanna's Tiffin Room
Vivek Kitchen
Hammered
Tiffun
Murphy's Brewhouse – The Paul Bangalore
Amexicano
Kick Out the Jams
The Big Barbeque
Manjit Da Dhaba
Coffee Mechanics
Frosted Memories
Avesta Cafe Pizzeria
Sardarji Ka Dhaba
TBC Sky Lounge
Soni Spa Cuisine
Pepsa Zaring
Bengal Sweets
Sea Lords BBQ & Grills
Rauf Ki Rani
Barbeque Nation
Hungree Belly
Mirchi Of Stories
OKI
Malleshwaram
The Bridge Walk Cafe
Aligarh House
The Pancake Story
Fabindia Cafe
Foorin - House of Sanctuaire
The Hole in the Wall Cafe
Asia Kitchen By Mainland China
The Old Fashioned Bar
Let Us Cooper
TRIGGERATO
@Italy
NISO Chinese Restaurant
Throwback - Pub & Dine
Smoke House Deli
Al Sadique
Lakshmi Narain Refreshments
CTR
The Black Pearl
Yauatcha
A2B - Adyar Ananda Bhavan
Ayodhya Food Line
One Pot The Rotel
Lincoln Bistro
KetoWheels
Jumping Beans
Big Pitcher
Karnataka Bayani point
Dirty House – The Royale Senate Hotel
O.G. Vadar & Sons
Firangi Bake
Hotel The Shack
Communit
Toast & Tonic
Vishnu Garden
The Druid Garden
Barbecued - By The Orchard
Kitchen Kitta
Royal Afghan – ITC Windsor
Byg Brewski Brewing Company
Sea Rock
Red Onion
Indi's Comfort Food
Belly Squad Food Truck
MISU
Le Casse Crote
Republic Of Noodles – Lemon Tree Premier
New Biryani Mane
Chaat Street
Millers 46
Street Cool Point
Fit Dish Fetish
Feichazo
Hearts On Tarts
Cheles Kitchen
Chandulpur Hotel
Banashankari
Banaswadi
Bannerghatta Road
Basavanagudi
Basaveshwara Nagar
Bellandur
Bommanahalli
Brigade Road
Brookefield
BTM
Central Bangalore
Church Street
City Market
Commercial Street
Cunningham Road
CV Raman Nagar
Domlur
East Bangalore
Ejipura
Electronic City
Frazer Town
HBR Layout
Hebbal
Hennur
Hosur Road
HSR
Indiranagar
Infantry Road
ITPL Main Road, Whitefield
Jalahalli
Jayanagar
Jeevan Bhima Nagar
JP Nagar
Kaggadasapura
Kalyan Nagar
Kammanahalli
Kanakapura Road
Kengeri
Koramangala
Koramangala 1st Block
Koramangala 2nd Block
Koramangala 3rd Block
Koramangala 4th Block
Koramangala 5th Block
Koramangala 6th Block
Koramangala 7th Block
Koramangala 8th Block
KR Puram
Kumaraswamy Layout
Langford Town
Lavelle Road
Magadi Road
Majestic
Malleshwaram
Marathahalli
MG Road
Mysore Road
Nagarbhavi
Nagawara
New BEL Road
North Bangalore
Old Airport Road
Old Madras Road
Peenya
Race Course Road
Rajajinagar
Rajarajeshwari Nagar
Rammurthy Nagar
Residency Road
Richmond Road
RT Nagar
Sadashiv Nagar
Sahakara Nagar
Sanjay Nagar
Seshadripuram
Shanti Nagar
Shivajinagar
South Bangalore
St. Marks Road
Thippasandra
Ulsoor
Uttarahalli
Varthur Main Road, Whitefield
Vasanth Nagar
Vijay Nagar
West Bangalore
Whitefield
Wilson Garden
Yelahanka
Yeshwanthpur