Level 2:

Task 1: Table Booking and Online Delivery

Determine the percentage of restaurants that offer table booking and online delivery. Compare the average ratings of restaurants with table booking and those without. Analyze the availability of online delivery among restaurants with different price ranges.

```
library(base)
t1<-df$Has.Online.delivery
Online_percent<-prop.table(table(t1))
Table_booking<-prop.table(table(df$Has.Table.booking))
print("Online Delivery")
```

```
## [1] "Online Delivery"
```

```
a1<-Online_percent['Yes']*100
a1
```

```
##      Yes
## 25.66223
```

```
print("Table Booking")
```

```
## [1] "Table Booking"
```

```
a2<-Table_booking['Yes']*100
a2
```

```
##      Yes
## 12.12438
```

```
avg_rating<-aggregate(Aggregate.rating~Has.Table.booking,data=df,FUN = mean)
print("Average Rating Of Restaurants")
```

```
## [1] "Average Rating Of Restaurants"
```

```
avg_rating
```

```
##   Has.Table.booking Aggregate.rating
## 1                No         2.559359
## 2               Yes         3.441969
```

```
Online_delivery_availability<-
aggregate(Has.Online.delivery~Price.range,data=df,FUN=function(x)
mean(x=='Yes')*100)

print("Availability Of Online Delivery with Different Price Ranges")
```

```
## [1] "Availability Of Online Delivery with Different Price Ranges"
```

```
Online_delivery_availability
```

```
##   Price.range Has.Online.delivery
## 1           1           15.774077
## 2           2           41.310633
## 3           3           29.190341
## 4           4            9.044369
```

Task 2: Price Range Analysis

Determine the most common price range among all the restaurants. Calculate the average rating for each price range. Identify the color that represents the highest average rating among different price ranges.

```
tab<-table(df$Price.range)
print("Most Common Price Range")

## [1] "Most Common Price Range"

names(tab[which.max(tab)])

## [1] "1"

print("Average Rating for each Price Range")

## [1] "Average Rating for each Price Range"

library(dplyr)
avg_rating_diff_price_range<-df %>% group_by(price_range=df$Price.range) %>%
summarize( Average_Rating=mean(Aggregate.rating))

avg_rating_diff_price_range

## # A tibble: 4 × 2
##   price_range Average_Rating
##         <int>          <dbl>
## 1           1           2.00
## 2           2           2.94
## 3           3           3.68
## 4           4           3.82

highest_avg_rating<-avg_rating_diff_price_range %>%
filter(Average_Rating==max(Average_Rating))

color_highest_price_avg_rate<-df %>% group_by(Rating.color) %>%
filter(Price.range==highest_avg_rating$price_range) %>% summarise(count=n())
print("Color that represents the highest
average rating among different price ranges")

## [1] "Color that represents the highest\naverage rating among different
price ranges"

color_highest_price_avg_rate
```

```
## # A tibble: 6 × 2
##   Rating.color count
##   <chr>        <int>
## 1 Dark Green      74
## 2 Green          194
## 3 Orange         101
## 4 Red              6
## 5 White           11
## 6 Yellow         200
```

Task 3: Feature Engineering

Extract additional features from the existing columns, such as the length of the restaurant name or address. Create new features like “Has Table Booking” or “Has Online Delivery” by encoding categorical variables.

```
df['Length_of_Restaurant_name']<-nchar(df$Restaurant.Name)
df['Length_of_Restaurant_Address']<-nchar(df$Address)
print("Length of Restaurant Address")
```

```
## [1] "Length of Restaurant Address"
```

```
head(df$Length_of_Restaurant_Address)
```

```
## [1] 71 67 56 70 64 71
```

```
print("Length of Restaurant Name")
```

```
## [1] "Length of Restaurant Name"
```

```
head(df$Length_of_Restaurant_name)
```

```
## [1] 16 16 22  4 11 12
```

```
df['Encode_Has_Table_Booking']=as.numeric(factor(df$Has.Table.booking))
print("Encoded Restaurant_Has_Table_Booking")
```

```
## [1] "Encoded Restaurant_Has_Table_Booking"
```

```
head(df$Encode_Has_Table_Booking)
```

```
## [1] 2 2 2 1 2 1
```

```
df['Encode_Has_Online_Delivery']=as.numeric(factor(df$Has.Online.delivery))
print("Encoded Restaurant_Has_Online_Delivery")
```

```
## [1] "Encoded Restaurant_Has_Online_Delivery"
```

```
head(df$Encode_Has_Online_Delivery)
```

```
## [1] 1 1 1 1 1 1
```