Zomato

library(tidyverse)  
library(VIM)  
library(GGally)

#Import Data Set  
zomato = read\_csv("zomato.csv")

## Parsed with column specification:  
## cols(  
## url = col\_character(),  
## address = col\_character(),  
## name = col\_character(),  
## online\_order = col\_character(),  
## book\_table = col\_character(),  
## rate = col\_character(),  
## votes = col\_double(),  
## phone = col\_character(),  
## location = col\_character(),  
## rest\_type = col\_character(),  
## dish\_liked = col\_character(),  
## cuisines = col\_character(),  
## `approx\_cost(for two people)` = col\_number(),  
## reviews\_list = col\_character(),  
## menu\_item = col\_character(),  
## `listed\_in(type)` = col\_character(),  
## `listed\_in(city)` = col\_character()  
## )

#Examine Structure  
str(zomato)

## tibble [51,717 x 17] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ url : chr [1:51717] "https://www.zomato.com/bangalore/jalsa-banashankari?context=eyJzZSI6eyJlIjpbNTg2OTQsIjE4Mzc1NDc0IiwiNTkwOTAiLCI"| \_\_truncated\_\_ "https://www.zomato.com/bangalore/spice-elephant-banashankari?context=eyJzZSI6eyJlIjpbIjU4Njk0IiwxODM3NTQ3NCwiNT"| \_\_truncated\_\_ "https://www.zomato.com/SanchurroBangalore?context=eyJzZSI6eyJlIjpbIjU4Njk0IiwiMTgzNzU0NzQiLDU5MDkwLCIxODM4Mjk0N"| \_\_truncated\_\_ "https://www.zomato.com/bangalore/addhuri-udupi-bhojana-banashankari?context=eyJzZSI6eyJlIjpbIjU4Njk0IiwiMTgzNzU"| \_\_truncated\_\_ ...  
## $ address : chr [1:51717] "942, 21st Main Road, 2nd Stage, Banashankari, Bangalore" "2nd Floor, 80 Feet Road, Near Big Bazaar, 6th Block, Kathriguppe, 3rd Stage, Banashankari, Bangalore" "1112, Next to KIMS Medical College, 17th Cross, 2nd Stage, Banashankari, Bangalore" "1st Floor, Annakuteera, 3rd Stage, Banashankari, Bangalore" ...  
## $ name : chr [1:51717] "Jalsa" "Spice Elephant" "San Churro Cafe" "Addhuri Udupi Bhojana" ...  
## $ online\_order : chr [1:51717] "Yes" "Yes" "Yes" "No" ...  
## $ book\_table : chr [1:51717] "Yes" "No" "No" "No" ...  
## $ rate : chr [1:51717] "4.1/5" "4.1/5" "3.8/5" "3.7/5" ...  
## $ votes : num [1:51717] 775 787 918 88 166 ...  
## $ phone : chr [1:51717] "080 42297555\r\n+91 9743772233" "080 41714161" "+91 9663487993" "+91 9620009302" ...  
## $ location : chr [1:51717] "Banashankari" "Banashankari" "Banashankari" "Banashankari" ...  
## $ rest\_type : chr [1:51717] "Casual Dining" "Casual Dining" "Cafe, Casual Dining" "Quick Bites" ...  
## $ dish\_liked : chr [1:51717] "Pasta, Lunch Buffet, Masala Papad, Paneer Lajawab, Tomato Shorba, Dum Biryani, Sweet Corn Soup" "Momos, Lunch Buffet, Chocolate Nirvana, Thai Green Curry, Paneer Tikka, Dum Biryani, Chicken Biryani" "Churros, Cannelloni, Minestrone Soup, Hot Chocolate, Pink Sauce Pasta, Salsa, Veg Supreme Pizza" "Masala Dosa" ...  
## $ cuisines : chr [1:51717] "North Indian, Mughlai, Chinese" "Chinese, North Indian, Thai" "Cafe, Mexican, Italian" "South Indian, North Indian" ...  
## $ approx\_cost(for two people): num [1:51717] 800 800 800 300 600 600 800 600 700 550 ...  
## $ reviews\_list : chr [1:51717] "[('Rated 4.0', 'RATED\\n A beautiful place to dine in.The interiors take you back to the Mughal era. The light"| \_\_truncated\_\_ "[('Rated 4.0', 'RATED\\n Had been here for dinner with family. Turned out to be a good choose suitable for all"| \_\_truncated\_\_ "[('Rated 3.0', \"RATED\\n Ambience is not that good enough and it's not a pocket friendly cafe and the quantit"| \_\_truncated\_\_ "[('Rated 4.0', \"RATED\\n Great food and proper Karnataka style full meals. Been there twice and was fully sat"| \_\_truncated\_\_ ...  
## $ menu\_item : chr [1:51717] "[]" "[]" "[]" "[]" ...  
## $ listed\_in(type) : chr [1:51717] "Buffet" "Buffet" "Buffet" "Buffet" ...  
## $ listed\_in(city) : chr [1:51717] "Banashankari" "Banashankari" "Banashankari" "Banashankari" ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. url = col\_character(),  
## .. address = col\_character(),  
## .. name = col\_character(),  
## .. online\_order = col\_character(),  
## .. book\_table = col\_character(),  
## .. rate = col\_character(),  
## .. votes = col\_double(),  
## .. phone = col\_character(),  
## .. location = col\_character(),  
## .. rest\_type = col\_character(),  
## .. dish\_liked = col\_character(),  
## .. cuisines = col\_character(),  
## .. `approx\_cost(for two people)` = col\_number(),  
## .. reviews\_list = col\_character(),  
## .. menu\_item = col\_character(),  
## .. `listed\_in(type)` = col\_character(),  
## .. `listed\_in(city)` = col\_character()  
## .. )

#Review data types  
summary(zomato)

## url address name online\_order   
## Length:51717 Length:51717 Length:51717 Length:51717   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## book\_table rate votes phone   
## Length:51717 Length:51717 Min. : 0.0 Length:51717   
## Class :character Class :character 1st Qu.: 7.0 Class :character   
## Mode :character Mode :character Median : 41.0 Mode :character   
## Mean : 283.7   
## 3rd Qu.: 198.0   
## Max. :16832.0   
##   
## location rest\_type dish\_liked cuisines   
## Length:51717 Length:51717 Length:51717 Length:51717   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## approx\_cost(for two people) reviews\_list menu\_item   
## Min. : 40.0 Length:51717 Length:51717   
## 1st Qu.: 300.0 Class :character Class :character   
## Median : 400.0 Mode :character Mode :character   
## Mean : 555.4   
## 3rd Qu.: 650.0   
## Max. :6000.0   
## NA's :346   
## listed\_in(type) listed\_in(city)   
## Length:51717 Length:51717   
## Class :character Class :character   
## Mode :character Mode :character   
##   
##   
##   
##

#Glimpse first entries in each column  
glimpse(zomato)

## Rows: 51,717  
## Columns: 17  
## $ url <chr> "https://www.zomato.com/bangalore/jal...  
## $ address <chr> "942, 21st Main Road, 2nd Stage, Bana...  
## $ name <chr> "Jalsa", "Spice Elephant", "San Churr...  
## $ online\_order <chr> "Yes", "Yes", "Yes", "No", "No", "Yes...  
## $ book\_table <chr> "Yes", "No", "No", "No", "No", "No", ...  
## $ rate <chr> "4.1/5", "4.1/5", "3.8/5", "3.7/5", "...  
## $ votes <dbl> 775, 787, 918, 88, 166, 286, 8, 2556,...  
## $ phone <chr> "080 42297555\r\n+91 9743772233", "08...  
## $ location <chr> "Banashankari", "Banashankari", "Bana...  
## $ rest\_type <chr> "Casual Dining", "Casual Dining", "Ca...  
## $ dish\_liked <chr> "Pasta, Lunch Buffet, Masala Papad, P...  
## $ cuisines <chr> "North Indian, Mughlai, Chinese", "Ch...  
## $ `approx\_cost(for two people)` <dbl> 800, 800, 800, 300, 600, 600, 800, 60...  
## $ reviews\_list <chr> "[('Rated 4.0', 'RATED\\n A beautifu...  
## $ menu\_item <chr> "[]", "[]", "[]", "[]", "[]", "[]", "...  
## $ `listed\_in(type)` <chr> "Buffet", "Buffet", "Buffet", "Buffet...  
## $ `listed\_in(city)` <chr> "Banashankari", "Banashankari", "Bana...

#Reduce Data Set to Features of Interest  
#URL and Phone Not Likely to Contribute to Rate  
#Address somewhat duplicative of Location/listed\_in(city)  
#Menu\_Item Duplicative of Dish\_Liked  
#Reviews\_List Duplicative of Rate  
zomato\_reduced = zomato %>% select(-url, -phone, -address, -reviews\_list, -menu\_item)  
str(zomato\_reduced)

## tibble [51,717 x 12] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ name : chr [1:51717] "Jalsa" "Spice Elephant" "San Churro Cafe" "Addhuri Udupi Bhojana" ...  
## $ online\_order : chr [1:51717] "Yes" "Yes" "Yes" "No" ...  
## $ book\_table : chr [1:51717] "Yes" "No" "No" "No" ...  
## $ rate : chr [1:51717] "4.1/5" "4.1/5" "3.8/5" "3.7/5" ...  
## $ votes : num [1:51717] 775 787 918 88 166 ...  
## $ location : chr [1:51717] "Banashankari" "Banashankari" "Banashankari" "Banashankari" ...  
## $ rest\_type : chr [1:51717] "Casual Dining" "Casual Dining" "Cafe, Casual Dining" "Quick Bites" ...  
## $ dish\_liked : chr [1:51717] "Pasta, Lunch Buffet, Masala Papad, Paneer Lajawab, Tomato Shorba, Dum Biryani, Sweet Corn Soup" "Momos, Lunch Buffet, Chocolate Nirvana, Thai Green Curry, Paneer Tikka, Dum Biryani, Chicken Biryani" "Churros, Cannelloni, Minestrone Soup, Hot Chocolate, Pink Sauce Pasta, Salsa, Veg Supreme Pizza" "Masala Dosa" ...  
## $ cuisines : chr [1:51717] "North Indian, Mughlai, Chinese" "Chinese, North Indian, Thai" "Cafe, Mexican, Italian" "South Indian, North Indian" ...  
## $ approx\_cost(for two people): num [1:51717] 800 800 800 300 600 600 800 600 700 550 ...  
## $ listed\_in(type) : chr [1:51717] "Buffet" "Buffet" "Buffet" "Buffet" ...  
## $ listed\_in(city) : chr [1:51717] "Banashankari" "Banashankari" "Banashankari" "Banashankari" ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. url = col\_character(),  
## .. address = col\_character(),  
## .. name = col\_character(),  
## .. online\_order = col\_character(),  
## .. book\_table = col\_character(),  
## .. rate = col\_character(),  
## .. votes = col\_double(),  
## .. phone = col\_character(),  
## .. location = col\_character(),  
## .. rest\_type = col\_character(),  
## .. dish\_liked = col\_character(),  
## .. cuisines = col\_character(),  
## .. `approx\_cost(for two people)` = col\_number(),  
## .. reviews\_list = col\_character(),  
## .. menu\_item = col\_character(),  
## .. `listed\_in(type)` = col\_character(),  
## .. `listed\_in(city)` = col\_character()  
## .. )

summary(zomato\_reduced)

## name online\_order book\_table rate   
## Length:51717 Length:51717 Length:51717 Length:51717   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## votes location rest\_type dish\_liked   
## Min. : 0.0 Length:51717 Length:51717 Length:51717   
## 1st Qu.: 7.0 Class :character Class :character Class :character   
## Median : 41.0 Mode :character Mode :character Mode :character   
## Mean : 283.7   
## 3rd Qu.: 198.0   
## Max. :16832.0   
##   
## cuisines approx\_cost(for two people) listed\_in(type)   
## Length:51717 Min. : 40.0 Length:51717   
## Class :character 1st Qu.: 300.0 Class :character   
## Mode :character Median : 400.0 Mode :character   
## Mean : 555.4   
## 3rd Qu.: 650.0   
## Max. :6000.0   
## NA's :346   
## listed\_in(city)   
## Length:51717   
## Class :character   
## Mode :character   
##   
##   
##   
##

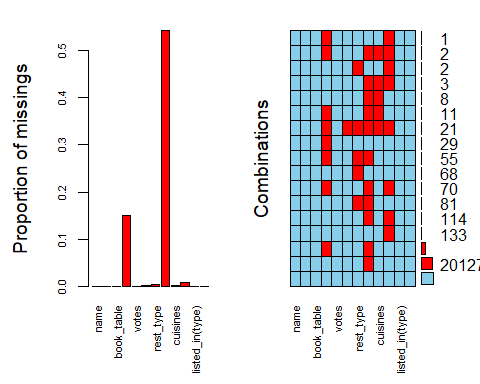
glimpse(zomato\_reduced)

## Rows: 51,717  
## Columns: 12  
## $ name <chr> "Jalsa", "Spice Elephant", "San Churr...  
## $ online\_order <chr> "Yes", "Yes", "Yes", "No", "No", "Yes...  
## $ book\_table <chr> "Yes", "No", "No", "No", "No", "No", ...  
## $ rate <chr> "4.1/5", "4.1/5", "3.8/5", "3.7/5", "...  
## $ votes <dbl> 775, 787, 918, 88, 166, 286, 8, 2556,...  
## $ location <chr> "Banashankari", "Banashankari", "Bana...  
## $ rest\_type <chr> "Casual Dining", "Casual Dining", "Ca...  
## $ dish\_liked <chr> "Pasta, Lunch Buffet, Masala Papad, P...  
## $ cuisines <chr> "North Indian, Mughlai, Chinese", "Ch...  
## $ `approx\_cost(for two people)` <dbl> 800, 800, 800, 300, 600, 600, 800, 60...  
## $ `listed\_in(type)` <chr> "Buffet", "Buffet", "Buffet", "Buffet...  
## $ `listed\_in(city)` <chr> "Banashankari", "Banashankari", "Bana...

#Row-wise deletion of Null Values for rate, rest\_type, `approx\_cost(for two people)`,location, and cuisines  
colSums(is.na(zomato\_reduced))

## name online\_order   
## 0 0   
## book\_table rate   
## 0 7775   
## votes location   
## 0 21   
## rest\_type dish\_liked   
## 227 28078   
## cuisines approx\_cost(for two people)   
## 45 346   
## listed\_in(type) listed\_in(city)   
## 0 0

vim\_plot = aggr(zomato\_reduced, numbers = TRUE, prop = c(TRUE, FALSE),cex.axis=.7)

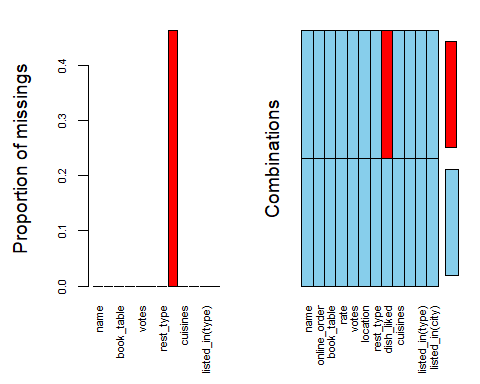


zomato\_reduced = zomato\_reduced %>% drop\_na(rest\_type, `approx\_cost(for two people)`,location, cuisines, rate)  
#dish\_liked contains a large amount of missing values (Worthy of Its Own Analysis?)  
  
#Investigate missing information  
colSums(is.na(zomato\_reduced))

## name online\_order   
## 0 0   
## book\_table rate   
## 0 0   
## votes location   
## 0 0   
## rest\_type dish\_liked   
## 0 20127   
## cuisines approx\_cost(for two people)   
## 0 0   
## listed\_in(type) listed\_in(city)   
## 0 0

vim\_plot = aggr(zomato\_reduced, numbers = TRUE, prop = c(TRUE, FALSE),cex.axis=.7)

## Warning in plot.aggr(res, ...): not enough horizontal space to display  
## frequencies



#High degree of missingness in dish\_liked, rate (How To Handle?)  
#Could potentially drop nulls for features with lesser degree of missingness

#Investigate Response Variable(s)  
unique(zomato\_reduced$rate)

## [1] "4.1/5" "3.8/5" "3.7/5" "3.6/5" "4.6/5" "4.0/5" "4.2/5" "3.9/5"   
## [9] "3.1/5" "3.0/5" "3.2/5" "3.3/5" "2.8/5" "4.4/5" "4.3/5" "NEW"   
## [17] "2.9/5" "3.5/5" "2.6/5" "3.8 /5" "3.4/5" "4.5/5" "2.5/5" "2.7/5"   
## [25] "4.7/5" "2.4/5" "2.2/5" "2.3/5" "3.4 /5" "-" "3.6 /5" "4.8/5"   
## [33] "3.9 /5" "4.2 /5" "4.0 /5" "4.1 /5" "3.7 /5" "3.1 /5" "2.9 /5" "3.3 /5"  
## [41] "2.8 /5" "3.5 /5" "2.7 /5" "2.5 /5" "3.2 /5" "2.6 /5" "4.5 /5" "4.3 /5"  
## [49] "4.4 /5" "4.9/5" "2.1/5" "2.0/5" "1.8/5" "4.6 /5" "4.9 /5" "3.0 /5"  
## [57] "4.8 /5" "2.3 /5" "4.7 /5" "2.4 /5" "2.1 /5" "2.2 /5" "2.0 /5" "1.8 /5"

#Rate contains "NEW" and "-"that should be removed  
zomato\_reduced = zomato\_reduced %>%  
 filter(zomato\_reduced$rate != "-")  
zomato\_reduced = zomato\_reduced %>%  
 filter(zomato\_reduced$rate != "NEW")  
unique(zomato\_reduced$rate)

## [1] "4.1/5" "3.8/5" "3.7/5" "3.6/5" "4.6/5" "4.0/5" "4.2/5" "3.9/5"   
## [9] "3.1/5" "3.0/5" "3.2/5" "3.3/5" "2.8/5" "4.4/5" "4.3/5" "2.9/5"   
## [17] "3.5/5" "2.6/5" "3.8 /5" "3.4/5" "4.5/5" "2.5/5" "2.7/5" "4.7/5"   
## [25] "2.4/5" "2.2/5" "2.3/5" "3.4 /5" "3.6 /5" "4.8/5" "3.9 /5" "4.2 /5"  
## [33] "4.0 /5" "4.1 /5" "3.7 /5" "3.1 /5" "2.9 /5" "3.3 /5" "2.8 /5" "3.5 /5"  
## [41] "2.7 /5" "2.5 /5" "3.2 /5" "2.6 /5" "4.5 /5" "4.3 /5" "4.4 /5" "4.9/5"   
## [49] "2.1/5" "2.0/5" "1.8/5" "4.6 /5" "4.9 /5" "3.0 /5" "4.8 /5" "2.3 /5"  
## [57] "4.7 /5" "2.4 /5" "2.1 /5" "2.2 /5" "2.0 /5" "1.8 /5"

#Remove the "/5" from rating  
zomato\_reduced = zomato\_reduced %>%  
 mutate(rate = as.character(lapply(rate, gsub, pattern = "/5", replacement = ""))) %>%  
 mutate(rate = as.numeric(rate))  
str(zomato\_reduced)

## tibble [41,263 x 12] (S3: tbl\_df/tbl/data.frame)  
## $ name : chr [1:41263] "Jalsa" "Spice Elephant" "San Churro Cafe" "Addhuri Udupi Bhojana" ...  
## $ online\_order : chr [1:41263] "Yes" "Yes" "Yes" "No" ...  
## $ book\_table : chr [1:41263] "Yes" "No" "No" "No" ...  
## $ rate : num [1:41263] 4.1 4.1 3.8 3.7 3.8 3.8 3.6 4.6 4 4.2 ...  
## $ votes : num [1:41263] 775 787 918 88 166 ...  
## $ location : chr [1:41263] "Banashankari" "Banashankari" "Banashankari" "Banashankari" ...  
## $ rest\_type : chr [1:41263] "Casual Dining" "Casual Dining" "Cafe, Casual Dining" "Quick Bites" ...  
## $ dish\_liked : chr [1:41263] "Pasta, Lunch Buffet, Masala Papad, Paneer Lajawab, Tomato Shorba, Dum Biryani, Sweet Corn Soup" "Momos, Lunch Buffet, Chocolate Nirvana, Thai Green Curry, Paneer Tikka, Dum Biryani, Chicken Biryani" "Churros, Cannelloni, Minestrone Soup, Hot Chocolate, Pink Sauce Pasta, Salsa, Veg Supreme Pizza" "Masala Dosa" ...  
## $ cuisines : chr [1:41263] "North Indian, Mughlai, Chinese" "Chinese, North Indian, Thai" "Cafe, Mexican, Italian" "South Indian, North Indian" ...  
## $ approx\_cost(for two people): num [1:41263] 800 800 800 300 600 600 800 600 700 550 ...  
## $ listed\_in(type) : chr [1:41263] "Buffet" "Buffet" "Buffet" "Buffet" ...  
## $ listed\_in(city) : chr [1:41263] "Banashankari" "Banashankari" "Banashankari" "Banashankari" ...

summary(zomato\_reduced)

## name online\_order book\_table rate   
## Length:41263 Length:41263 Length:41263 Min. :1.800   
## Class :character Class :character Class :character 1st Qu.:3.400   
## Mode :character Mode :character Mode :character Median :3.700   
## Mean :3.702   
## 3rd Qu.:4.000   
## Max. :4.900   
## votes location rest\_type dish\_liked   
## Min. : 0.0 Length:41263 Length:41263 Length:41263   
## 1st Qu.: 21.0 Class :character Class :character Class :character   
## Median : 73.0 Mode :character Mode :character Mode :character   
## Mean : 352.7   
## 3rd Qu.: 277.0   
## Max. :16832.0   
## cuisines approx\_cost(for two people) listed\_in(type)   
## Length:41263 Min. : 40.0 Length:41263   
## Class :character 1st Qu.: 300.0 Class :character   
## Mode :character Median : 500.0 Mode :character   
## Mean : 603.9   
## 3rd Qu.: 750.0   
## Max. :6000.0   
## listed\_in(city)   
## Length:41263   
## Class :character   
## Mode :character   
##   
##   
##

glimpse(zomato\_reduced)

## Rows: 41,263  
## Columns: 12  
## $ name <chr> "Jalsa", "Spice Elephant", "San Churr...  
## $ online\_order <chr> "Yes", "Yes", "Yes", "No", "No", "Yes...  
## $ book\_table <chr> "Yes", "No", "No", "No", "No", "No", ...  
## $ rate <dbl> 4.1, 4.1, 3.8, 3.7, 3.8, 3.8, 3.6, 4....  
## $ votes <dbl> 775, 787, 918, 88, 166, 286, 8, 2556,...  
## $ location <chr> "Banashankari", "Banashankari", "Bana...  
## $ rest\_type <chr> "Casual Dining", "Casual Dining", "Ca...  
## $ dish\_liked <chr> "Pasta, Lunch Buffet, Masala Papad, P...  
## $ cuisines <chr> "North Indian, Mughlai, Chinese", "Ch...  
## $ `approx\_cost(for two people)` <dbl> 800, 800, 800, 300, 600, 600, 800, 60...  
## $ `listed\_in(type)` <chr> "Buffet", "Buffet", "Buffet", "Buffet...  
## $ `listed\_in(city)` <chr> "Banashankari", "Banashankari", "Bana...

#Investigate duplicate information  
#51717 observations, close to 12,000 restaurants  
count(as.data.frame(unique(zomato\_reduced$name)))

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 6602

#Duplicates by name and address not really duplicates (multiple locations, multiple reviews)  
count(as.data.frame(unique(zomato\_reduced$online\_order)))

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 2

count(as.data.frame(unique(zomato\_reduced$book\_table)))

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 2

#Find variables that can be changed to factors  
zomato\_reduced = zomato\_reduced %>%  
 mutate(online\_order = as.factor(online\_order)) %>%   
 mutate(book\_table = as.factor(book\_table))  
  
  
##Count Unique Values in Key Columns  
count(as.data.frame(unique(zomato\_reduced$location)))

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 92

count(as.data.frame(unique(zomato\_reduced$`listed\_in(city)`)))

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 30

#Location vs. Listed\_In(City): Listed\_In(city) more manageable  
  
count(as.data.frame(unique(zomato\_reduced$rest\_type)))

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 87

count(as.data.frame(unique(zomato\_reduced$`listed\_in(type)`)))

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 7

#Rest\_Type vs. Listed\_In(Type): Listed\_In(type) more manageable  
  
  
count(as.data.frame(unique(zomato\_reduced$dish\_liked)))

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 5195

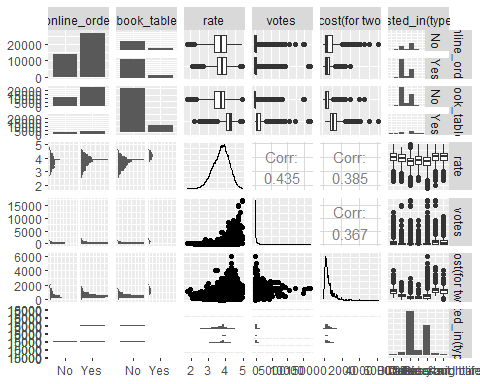
count(as.data.frame(unique(zomato\_reduced$cuisines)))

## # A tibble: 1 x 1  
## n  
## <int>  
## 1 2367

#Dish\_Liked, and Cuisines Have Many Unique Values  
#Worth finding most common and popular in each category

zomato\_corr = zomato\_reduced %>%  
 select(-location,-rest\_type, -dish\_liked, -cuisines, -`listed\_in(city)`, -name)  
ggpairs(zomato\_corr)

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



#Dish-Liked Analysis  
  
library(tm)

## Warning: package 'tm' was built under R version 4.0.5

## Loading required package: NLP

## Warning: package 'NLP' was built under R version 4.0.3

##   
## Attaching package: 'NLP'

## The following object is masked from 'package:ggplot2':  
##   
## annotate

library(SnowballC)

## Warning: package 'SnowballC' was built under R version 4.0.3

library(wordcloud)

## Warning: package 'wordcloud' was built under R version 4.0.5

## Loading required package: RColorBrewer

## Warning: package 'RColorBrewer' was built under R version 4.0.3

library(RColorBrewer)  
  
zomato\_dishliked = zomato %>% dplyr::select(dish\_liked) %>%  
 drop\_na(dish\_liked)  
str(zomato\_dishliked)

## tibble [23,639 x 1] (S3: tbl\_df/tbl/data.frame)  
## $ dish\_liked: chr [1:23639] "Pasta, Lunch Buffet, Masala Papad, Paneer Lajawab, Tomato Shorba, Dum Biryani, Sweet Corn Soup" "Momos, Lunch Buffet, Chocolate Nirvana, Thai Green Curry, Paneer Tikka, Dum Biryani, Chicken Biryani" "Churros, Cannelloni, Minestrone Soup, Hot Chocolate, Pink Sauce Pasta, Salsa, Veg Supreme Pizza" "Masala Dosa" ...

summary(zomato\_dishliked)

## dish\_liked   
## Length:23639   
## Class :character   
## Mode :character

glimpse(zomato\_dishliked)

## Rows: 23,639  
## Columns: 1  
## $ dish\_liked <chr> "Pasta, Lunch Buffet, Masala Papad, Paneer Lajawab, Toma...

###Export to CSV to Convert to TXT file externally  
write.csv(zomato\_dishliked,  
 file = 'zomato\_dishliked.csv',  
 row.names = FALSE,  
 na = '-9999')  
  
#Text Mining  
text = readLines('zomato\_dishliked.txt')  
TextDoc = Corpus(VectorSource(text))  
  
toSpace = content\_transformer(function (x , pattern ) gsub(pattern, " ", x))  
TextDoc = tm\_map(TextDoc, toSpace, "/")

## Warning in tm\_map.SimpleCorpus(TextDoc, toSpace, "/"): transformation drops  
## documents

TextDoc = tm\_map(TextDoc, toSpace, "@")

## Warning in tm\_map.SimpleCorpus(TextDoc, toSpace, "@"): transformation drops  
## documents

TextDoc = tm\_map(TextDoc, toSpace, "\\|")

## Warning in tm\_map.SimpleCorpus(TextDoc, toSpace, "\\|"): transformation drops  
## documents

# Convert the text to lower case  
TextDoc = tm\_map(TextDoc, content\_transformer(tolower))

## Warning in tm\_map.SimpleCorpus(TextDoc, content\_transformer(tolower)):  
## transformation drops documents

# Remove numbers  
TextDoc = tm\_map(TextDoc, removeNumbers)

## Warning in tm\_map.SimpleCorpus(TextDoc, removeNumbers): transformation drops  
## documents

# Remove english common stopwords  
TextDoc = tm\_map(TextDoc, removeWords, stopwords("english"))

## Warning in tm\_map.SimpleCorpus(TextDoc, removeWords, stopwords("english")):  
## transformation drops documents

TextDoc = tm\_map(TextDoc, removeWords, c("restaur"))

## Warning in tm\_map.SimpleCorpus(TextDoc, removeWords, c("restaur")):  
## transformation drops documents

# Remove punctuations  
TextDoc = tm\_map(TextDoc, removePunctuation)

## Warning in tm\_map.SimpleCorpus(TextDoc, removePunctuation): transformation drops  
## documents

# Eliminate extra white spaces  
TextDoc = tm\_map(TextDoc, stripWhitespace)

## Warning in tm\_map.SimpleCorpus(TextDoc, stripWhitespace): transformation drops  
## documents

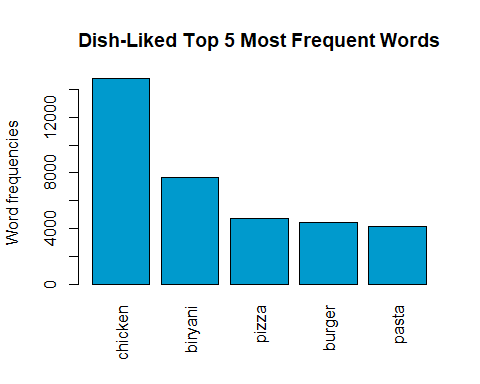
# Text stemming - which reduces words to their root form  
TextDoc = tm\_map(TextDoc, stemDocument)

## Warning in tm\_map.SimpleCorpus(TextDoc, stemDocument): transformation drops  
## documents

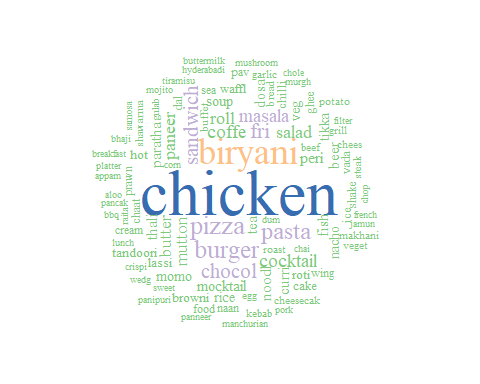
# Build a term-document matrix  
TextDoc\_dtm = TermDocumentMatrix(TextDoc)  
dtm\_m = as.matrix(TextDoc\_dtm)  
# Sort by descending value of frequency  
dtm\_v = sort(rowSums(dtm\_m),decreasing=TRUE)  
dtm\_d = data.frame(word = names(dtm\_v),freq=dtm\_v)  
# Display the top 5 most frequent words  
head(dtm\_d, 5)

## word freq  
## chicken chicken 14772  
## biryani biryani 7641  
## pizza pizza 4686  
## burger burger 4429  
## pasta pasta 4141

barplot(dtm\_d[1:5,]$freq, las = 3, names.arg = dtm\_d[1:5,]$word,  
 col ="deepskyblue3", main ="Dish-Liked Top 5 Most Frequent Words",  
 ylab = "Word frequencies")



#generate word cloud  
set.seed(123)  
wordcloud(words   
 = dtm\_d$word, freq = dtm\_d$freq, min.freq = 30,  
 max.words=100, random.order=FALSE, rot.per=0.40, family="serif",  
 colors=brewer.pal(5, "Accent"))



#Cuisines Analysis  
zomato\_cuisines = zomato\_reduced %>%  
 dplyr::select(cuisines)  
str(zomato\_cuisines)

## tibble [41,263 x 1] (S3: tbl\_df/tbl/data.frame)  
## $ cuisines: chr [1:41263] "North Indian, Mughlai, Chinese" "Chinese, North Indian, Thai" "Cafe, Mexican, Italian" "South Indian, North Indian" ...

summary(zomato\_cuisines)

## cuisines   
## Length:41263   
## Class :character   
## Mode :character

glimpse(zomato\_cuisines)

## Rows: 41,263  
## Columns: 1  
## $ cuisines <chr> "North Indian, Mughlai, Chinese", "Chinese, North Indian, ...

###Export to CSV to Convert to TXT file externally  
write.csv(zomato\_cuisines,  
 file = 'zomato\_cuisines.csv',  
 row.names = FALSE,  
 na = '-9999')  
  
#Text Mining  
text = readLines('zomato\_cuisines.txt')  
TextDoc = Corpus(VectorSource(text))  
  
toSpace = content\_transformer(function (x , pattern ) gsub(pattern, " ", x))  
TextDoc = tm\_map(TextDoc, toSpace, "/")

## Warning in tm\_map.SimpleCorpus(TextDoc, toSpace, "/"): transformation drops  
## documents

TextDoc = tm\_map(TextDoc, toSpace, "@")

## Warning in tm\_map.SimpleCorpus(TextDoc, toSpace, "@"): transformation drops  
## documents

TextDoc = tm\_map(TextDoc, toSpace, "\\|")

## Warning in tm\_map.SimpleCorpus(TextDoc, toSpace, "\\|"): transformation drops  
## documents

# Convert the text to lower case  
TextDoc = tm\_map(TextDoc, content\_transformer(tolower))

## Warning in tm\_map.SimpleCorpus(TextDoc, content\_transformer(tolower)):  
## transformation drops documents

# Remove numbers  
TextDoc = tm\_map(TextDoc, removeNumbers)

## Warning in tm\_map.SimpleCorpus(TextDoc, removeNumbers): transformation drops  
## documents

# Remove english common stopwords  
TextDoc = tm\_map(TextDoc, removeWords, stopwords("english"))

## Warning in tm\_map.SimpleCorpus(TextDoc, removeWords, stopwords("english")):  
## transformation drops documents

TextDoc = tm\_map(TextDoc, removeWords, c("food"))

## Warning in tm\_map.SimpleCorpus(TextDoc, removeWords, c("food")): transformation  
## drops documents

# Remove punctuations  
TextDoc = tm\_map(TextDoc, removePunctuation)

## Warning in tm\_map.SimpleCorpus(TextDoc, removePunctuation): transformation drops  
## documents

# Eliminate extra white spaces  
TextDoc = tm\_map(TextDoc, stripWhitespace)

## Warning in tm\_map.SimpleCorpus(TextDoc, stripWhitespace): transformation drops  
## documents

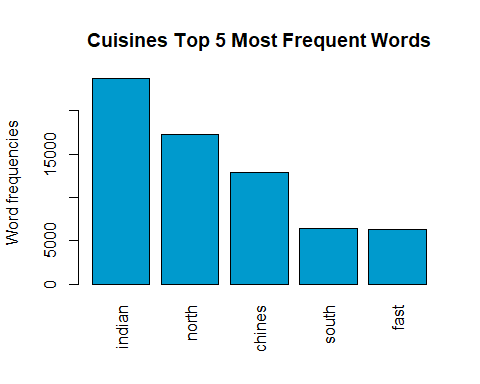
# Text stemming - which reduces words to their root form  
TextDoc = tm\_map(TextDoc, stemDocument)

## Warning in tm\_map.SimpleCorpus(TextDoc, stemDocument): transformation drops  
## documents

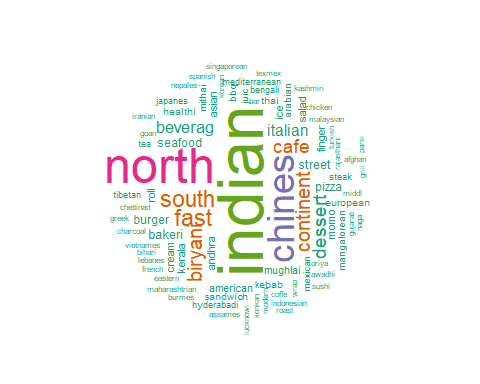
# Build a term-document matrix  
TextDoc\_dtm = TermDocumentMatrix(TextDoc)  
dtm\_m = as.matrix(TextDoc\_dtm)  
# Sort by descending value of frequency  
dtm\_v = sort(rowSums(dtm\_m),decreasing=TRUE)  
dtm\_d = data.frame(word = names(dtm\_v),freq=dtm\_v)  
# Display the top 5 most frequent words  
head(dtm\_d, 5)

## word freq  
## indian indian 23714  
## north north 17249  
## chines chines 12859  
## south south 6367  
## fast fast 6324

barplot(dtm\_d[1:5,]$freq, las = 3, names.arg = dtm\_d[1:5,]$word,  
 col = "deepskyblue3", main ="Cuisines Top 5 Most Frequent Words",  
 ylab = "Word frequencies")



#generate word cloud  
set.seed(123)  
wordcloud(words = dtm\_d$word, freq = dtm\_d$freq, min.freq = 20,  
 max.words=100, random.order=FALSE, rot.per=0.40,   
 colors=brewer.pal(5, "Dark2"))



#zomato\_reduced$DL\_Chicken <- ifelse(str\_contains(zomato\_reduced$dish\_liked, c("Chicken"), logic = "or", switch=FALSE), 1, 0)  
zomato\_reduced$DL\_Chicken = ifelse(grepl("Chicken",zomato\_reduced$dish\_liked),1, 0)  
zomato\_reduced$DL\_Biryani = ifelse(grepl("Biryani",zomato\_reduced$dish\_liked),1, 0)  
zomato\_reduced$DL\_Pizza = ifelse(grepl("Pizza",zomato\_reduced$dish\_liked),1, 0)  
zomato\_reduced$DL\_Burger = ifelse(grepl("Burger",zomato\_reduced$dish\_liked),1, 0)  
zomato\_reduced$DL\_Pasta = ifelse(grepl("Pasta",zomato\_reduced$dish\_liked),1, 0)  
zomato\_reduced$DL\_Sand = ifelse(grepl("Sandwich",zomato\_reduced$dish\_liked),1, 0)  
zomato\_reduced$DL\_Choc = ifelse(grepl("Chocol",zomato\_reduced$dish\_liked),1, 0)  
zomato\_reduced$DL\_Masala = ifelse(grepl("Masala",zomato\_reduced$dish\_liked),1, 0)  
zomato\_reduced$DL\_Fri = ifelse(grepl("Fri",zomato\_reduced$dish\_liked),1, 0)  
  
  
zomato\_reduced$Cuis\_NI = ifelse(grepl("North Indian",zomato\_reduced$cuisines),1, 0)  
zomato\_reduced$Cuis\_SI = ifelse(grepl("South Indian",zomato\_reduced$cuisines),1, 0)  
zomato\_reduced$Cuis\_CH = ifelse(grepl("Chines",zomato\_reduced$cuisines),1, 0)  
zomato\_reduced$Cuis\_FA = ifelse(grepl("Fast",zomato\_reduced$cuisines),1, 0)  
zomato\_reduced$Cuis\_Cont = ifelse(grepl("Continent",zomato\_reduced$cuisines),1, 0)  
zomato\_reduced$Cuis\_Caf = ifelse(grepl("Cafe",zomato\_reduced$cuisines),1, 0)  
  
  
  
view(zomato\_reduced)  
  
write.csv(zomato\_reduced,  
 file = 'zomato\_final.csv',  
 row.names = FALSE,  
 na = '-9999')