```
library(tidyr)
library(ggplot2)
library(ggthemes)
library(lubridate)
library(leaflet)
library(tm)
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

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

```
library(SnowballC)
```

Preliminary exploratory analysis

```
glimpse(ta)
```

```
## Observations: 179,789
## Variables: 30
## $ userId                    (chr) "0angelal", "0angelal", "0angelal", "...
## $ cityId                    (chr) "g60745", "g60745", "g60745", "g60745...
## $ venueId                   (chr) "d1907605", "d323250", "d3975907", "d...
## $ reviewDate                (chr) "7/14/12 0:00", "7/14/12 0:00", "5/11...
## $ rating                    (int) 3, 4, 5, 4, 4, 5, 4, 5, 4, 5, 4, 4, 5...
## $ votes                     (int) 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0...
## $ reviewText                (chr) "My scallops were overly salty, and t...
## $ service                   (int) 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1...
## $ vibe                      (int) 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0...
## $ desert                    (int) 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0...
## $ bathroom                  (int) 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ drink                     (int) 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0...
## $ cost                      (int) 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0...
## $ music                     (int) 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ location                  (int) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ parking                   (int) 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0...
## $ lunch                     (int) 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0...
## $ breakfast                 (int) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ dinner                    (int) 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0...
## $ reviewerType              (chr) "T", "T", "T", "T", "N", "N", "N", "N...
## $ totalReviewsOfReviewer    (int) 92, 92, 92, 9, 13, 13, 13, 13, 13, 23...
## $ avgHelpfulnessOfReviewer (dbl) 0.348, 0.348, 0.348, 0.111, 0.308, 0....
## $ venue                     (chr) "Deuxave", "Clio", "Swissbakers_Allst...
## $ dollars                   (chr) "$$$$", "$$$$", "$$", "$$$", "None", ...
## $ priceRange                (chr) "30 - 50", "35 - 100", "None", "30 - ...
## $ address                   (chr) "street-address:371 Commonwealth Aven...
## $ ranking                   (int) 4, 390, 999, 346, 613, 43, 108, 39, 3...
## $ cousine                   (chr) "Italian,French,American,Contemporary...
## $ goodFor                   (chr) "['Bar scene', 'Special occasions']",...
## $ city                      (chr) "Boston", "Boston", "Boston", "Boston...
```

```
unique(ta$reviewerType)
```

```
## [1] "T" "N" "L"
```

```
unique(ta$priceRange)
```

```
##   [1] "30 - 50"   "35 - 100"  "None"      "30 - 30"   "20 - 50"
##   [6] "20"        "35 - 80"   "30 - 40"   "18"        "41 - 80"
##  [11] "20-May"    "25 - 100"  "17 - 35"   "21 - 30"   "30"
##  [16] "20-Jun"    "20 - 100"  "29 - 31"   "15 - 30"   "41 - 100"
##  [21] "30-Oct"    "31 - 80"   "15 - 40"   "31 - 50"   "12"
##  [26] "21 - 80"   "20 - 40"   "35 - 45"   "31 - 40"   "26"
##  [31] "20 - 30"   "50 - 80"   "8"         "20-Oct"    "10-May"
##  [36] "13 - 28"   "46 - 46"   "40"        "35"        "15 - 25"
##  [41] "25"        "Oct-60"    "60 - 80"   "30 - 60"   "25 - 120"
##  [46] "20-Aug"    "18 - 30"   "25-Jul"    "25 - 35"   "15 - 50"
##  [51] "24-Oct"    "12-Mar"    "45"        "15-Oct"    "22-Aug"
##  [56] "Nov-34"    "15-May"    "80"        "95 - 95"   "40 - 60"
##  [61] "15 - 48"   "5 - 250"   "16-Jul"    "15 - 80"   "25-Apr"
##  [66] "30 - 80"   "30 - 31"   "10"        "10-Jun"    "28"
##  [71] "15"        "32"        "21 - 40"   "15 - 15"   "20 - 35"
##  [76] "20 - 25"   "25 - 30"   "4-Apr"     "30-Dec"    "Oct-35"
##  [81] "15 - 20"   "16"        "11"        "18-Aug"    "12-Jun"
##  [86] "30 - 35"   "20 - 20"   "17-Jul"    "22 - 45"   "24-Jul"
##  [91] "20-Dec"    "13 - 20"   "14 - 20"   "15-Aug"    "25-Oct"
##  [96] "Oct-50"    "31-May"    "15 - 35"   "6"         "5-Mar"
## [101] "21-Sep"    "12-May"    "22-Dec"    "10-Jul"    "13-Jul"
## [106] "12-Jul"    "30 - 70"   "20-Jul"    "8-Aug"     "30-May"
## [111] "25-Aug"    "50"        "5-May"     "17-Oct"    "17"
## [116] "8-May"     "10-Mar"    "30-Aug"    "23-Jul"    "25 - 50"
## [121] "18-Oct"    "50 - 50"   "20 - 120"  "Oct-40"    "9"
## [126] "20 - 60"   "30-Nov"    "17-Nov"    "9-Sep"     "30-Jun"
## [131] "26-Apr"    "11-Jun"    "10-Oct"    "50 - 60"   "24 - 80"
## [136] "40 - 50"   "25 - 45"   "Jul-45"    "12-Aug"    "20 - 45"
## [141] "May-80"    "25 - 75"   "Oct-55"    "40 - 120"  "25 - 60"
## [146] "198 - 300" "65 - 135"  "35 - 50"   "25 - 40"   "35 - 35"
## [151] "21 - 41"   "21 - 50"   "30 - 100"  "40 - 80"   "21 - 21"
## [156] "95 - 185"  "20-Mar"    "25 - 25"   "15 - 45"   "19 - 38"
## [161] "25 - 80"   "29-Aug"    "13-May"    "16 - 29"   "70 - 80"
## [166] "22"        "31 - 42"   "15-Jun"    "52"        "15 - 29"
## [171] "14 - 30"   "10-Feb"    "75 - 150"  "42"        "28-Dec"
## [176] "19-Oct"    "31 - 35"   "23"        "6-Jun"     "27-Jul"
## [181] "15 - 31"   "22 - 22"   "14"        "31 - 45"   "24-Aug"
## [186] "12-Dec"    "11-May"    "21 - 35"   "24 - 25"   "15-Jul"
## [191] "12-Oct"    "Jul-32"    "16 - 16"   "20-Sep"    "9-Feb"
## [196] "29"        "15 - 75"   "20 - 26"   "2"         "16-May"
```

Transformations on the dataset

```
# Change column types
ta$reviewDate <- mdy_hm(ta$reviewDate)
ta$reviewerType <- as.factor(ta$reviewerType)
ta$dollars <- as.factor(ta$dollars)

# Extract address - main dataset
ta$address_cleaned <- ta$address
ta$address_cleaned <- gsub("street-address:", "", ta$address_cleaned)
ta$address_cleaned <- gsub("locality:", "", ta$address_cleaned)
ta$address_cleaned <- gsub("region:", "", ta$address_cleaned)
ta$address_cleaned <- gsub("postal-code:", "", ta$address_cleaned)
ta$address <- gsub("postal-code: ", "postal-code:", ta$address)
ta$address <- gsub("postal-code:CA ", "postal-code:", ta$address)
ta$address <- gsub("postal-code:UT ", "postal-code:", ta$address)
ta$address <- gsub("postal-code:MA ", "postal-code:", ta$address)
ta$address <- gsub("postal-code:boston, ma", "", ta$address)
ta$address <- gsub("postal-code:bn1 2da", "", ta$address)
ta$address <- gsub("postal-code:CA", "", ta$address)
ta$postal_code <- ifelse(grepl("postal-code:", ta$address), sub(".*postal-code:*([0-9
]{4,5}) *([0-9]{4})?.*", "\\1", ta$address), "")
```

```
## Warning in grepl("postal-code:", ta$address): input string 95744 is invalid
## in this locale
```

```
## Warning in grepl("postal-code:", ta$address): input string 95758 is invalid
## in this locale
```

```
## Warning in grepl("postal-code:", ta$address): input string 102310 is
## invalid in this locale
```

```
## Warning in grepl("postal-code:", ta$address): input string 108418 is
## invalid in this locale
```

```
## Warning in grepl("postal-code:", ta$address): input string 111274 is
## invalid in this locale
```

```
# Checks
# ta$address[518:519]
# ta$postal_code[518:519]
# ta$postal_codeYN[518:519]
ta$Zipcode <- as.numeric(ta$postal_code)
table(ta$Zipcode)
```

```
##
##   210  1603  2101  2108  2109  2110  2111  2113  2114  2115  2116  2118
##   420     1   154  5255  6044  2730  2609 13761  2859  3623 17599  2960
##  2119  2120  2121  2122  2124  2125  2126  2127  2128  2129  2130  2131
##    26    69     5   204    99   241    15   463  1416   676   810   212
##  2132  2134  2135  2136  2138  2145  2171  2182  2189  2199  2205  2210
##   362   881   596   152     4    30     3     7   307  2516     1  6089
##  2211  2215  2228  2272  2445  2446  2459  2467 53005 63015 64116 84010
##     2  3990     3    35     5    16     4    16   133     1     1     5
## 84047 84095 84101 84102 84103 84104 84105 84106 84107 84108 84109 84111
##     1     8  3156   911   132    53   435   606    62   491   458  1947
## 84112 84113 84115 84116 84117 84118 84119 84120 84121 84122 84123 84124
##    46    20   571   741    95    20    92    34   499    37    39   100
## 84128 84129 84132 84144 84148 84150 85115 87510 90272 93933 94015 94019
##     3     9     4     7     5   148    76    10     1     3     2     8
## 94022 94080 94102 94103 94104 94105 94107 94108 94109 94110 94111 94112
##    17    48 12035  4437   840  3851  1925  4906  7648  4033  5602    73
## 94114 94115 94116 94117 94118 94119 94121 94122 94123 94124 94126 94127
##  2520  1966   218  1651  1295   173  1708  1136  3601    53     1   239
## 94128 94129 94130 94131 94132 94133 94134 94142 94143 94150 94158 94233
##    75   191     1   319   140 17099    24   147     3    76    98    10
## 94533 94804 94911 95030 98746
##     3     1    15     5     1
```

```
### Load zip code database
### Get from http://federalgovernmentzipcodes.us, download .zip titled 'Primary Locat
ion Only'

zipcodeDB <- read.csv("free-zipcode-database-Primary.csv", stringsAsFactors = FALSE,
header = TRUE)
zipcodeDBsub <- select(zipcodeDB, Zipcode, Lat, Long)

### Merge county Lat/Long into main table
taNew <- merge(ta, zipcodeDBsub, by.x = "Zipcode", by.y = "Zipcode", all.x = TRUE)
glimpse(ta)
```

```
## Observations: 179,789
## Variables: 33
## $ userId                    (chr) "0angelal", "0angelal", "0angelal", "...
## $ cityId                    (chr) "g60745", "g60745", "g60745", "g60745...
## $ venueId                   (chr) "d1907605", "d323250", "d3975907", "d...
## $ reviewDate                (time) 2012-07-14, 2012-07-14, 2013-05-11, ...
## $ rating                    (int) 3, 4, 5, 4, 4, 5, 4, 5, 4, 5, 4, 4, 5...
## $ votes                     (int) 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0...
## $ reviewText                (chr) "My scallops were overly salty, and t...
## $ service                   (int) 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1...
## $ vibe                      (int) 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0...
## $ desert                    (int) 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0...
## $ bathroom                  (int) 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ drink                     (int) 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0...
## $ cost                      (int) 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0...
## $ music                     (int) 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ location                  (int) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ parking                   (int) 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0...
## $ lunch                     (int) 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0...
## $ breakfast                 (int) 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ dinner                    (int) 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0...
## $ reviewerType              (fctr) T, T, T, T, N, N, N, N, N, L, L, L, ...
## $ totalReviewsOfReviewer    (int) 92, 92, 92, 9, 13, 13, 13, 13, 13, 23...
## $ avgHelpfulnessOfReviewer (dbl) 0.348, 0.348, 0.348, 0.111, 0.308, 0....
## $ venue                     (chr) "Deuxave", "Clio", "Swissbakers_Allst...
## $ dollars                   (fctr) $$$$, $$$$, $$, $$$, None, $$, None,...
## $ priceRange                (chr) "30 - 50", "35 - 100", "None", "30 - ...
## $ address                   (chr) "street-address:371 Commonwealth Aven...
## $ ranking                   (int) 4, 390, 999, 346, 613, 43, 108, 39, 3...
## $ cousine                   (chr) "Italian,French,American,Contemporary...
## $ goodFor                   (chr) "['Bar scene', 'Special occasions']",...
## $ city                      (chr) "Boston", "Boston", "Boston", "Boston...
## $ address_cleaned           (chr) "371 Commonwealth Avenue, Boston, MA ...
## $ postal_code               (chr) "02115", "02215", "02134", "02210", "...
## $ Zipcode                   (dbl) 2115, 2215, 2134, 2210, 2116, 2113, 2...
```

Summarize individual user reviews down to unique restaurants

```
# Summarize data
taSummarized <- taNew %>% group_by(city, venueId, venue, dollars, ranking, cousine, g
oodFor, address_cleaned, Zipcode, Lat, Long) %>%
  summarise(avgRating = mean(rating), noOfReviews = n())

# Fix column names
# colnames(taSummarized)
taSummarized <- rename(taSummarized, cuisine = cousine)
taSummarized <- rename(taSummarized, zipLon = Long)
taSummarized <- rename(taSummarized, zipLat = Lat)
taSummarized <- rename(taSummarized, zipCode = Zipcode)

# Clean up 'Cuisine' column
taSummarized$cuisine <- gsub(" & ", "&", taSummarized$cuisine)
taSummarized$cuisine <- gsub(",", " ", taSummarized$cuisine)
taSummarized$cuisine <- gsub("Dim Sum", "DimSum", taSummarized$cuisine)
taSummarized$cuisine <- gsub("Hong Kong", "HongKong", taSummarized$cuisine)
taSummarized$cuisine <- gsub("Tea Room", "TeaRoom", taSummarized$cuisine)
taSummarized$cuisine <- gsub("Ice Cream", "IceCream", taSummarized$cuisine)
```

Subset to individual cities

```
taSF <- filter(taSummarized, zipLat != "null", city == "San Francisco")
taSLC <- filter(taSummarized, zipLat != "null", city == "Salt Lake City")
taBos <- filter(taSummarized, zipLat != "null", city == "Boston")
```

Set up Leaflet

```
# glimpse(taSF)

# Set 'dollar' column levels and colors
levels(taSF$dollars) <- c("$", "$$","$$$","$$$$","None")
table(taSF$dollars)
```

```
##
##    $   $$  $$$ $$$$ None
##  128  498  306  140 1772
```

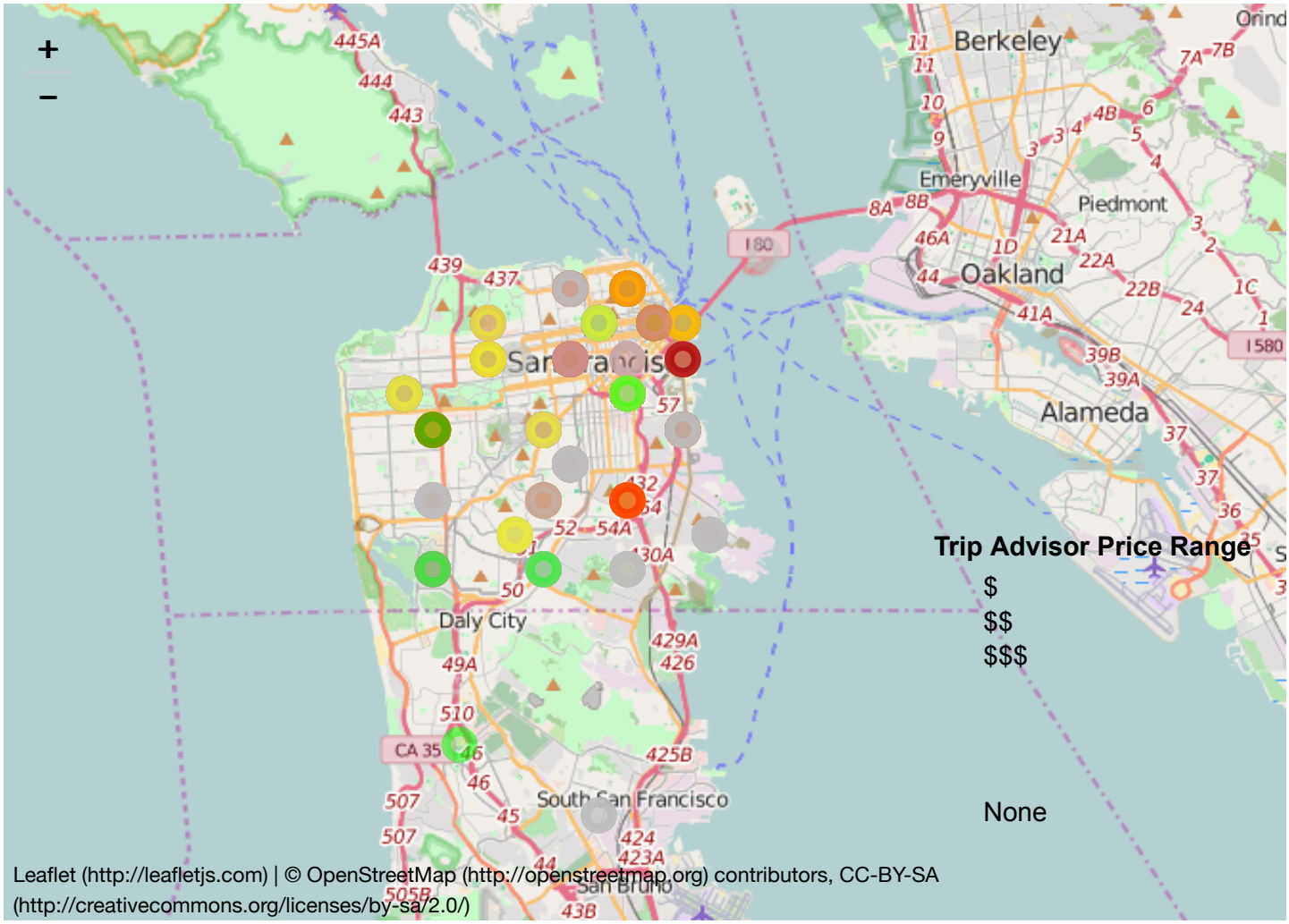
```
pal <- colorFactor(c("green", "yellow", "red", "darkred","gray"), domain = c("$", "$$
","$$$","$$$$","None"))

# San Francisco Leaflet map based on County lat/lon
mean(taSF$zipLat)
```

```
## [1] 37.77058
```

```
mean(taSF$zipLon)
```

```
## [1] -122.3959
```

```
taSF_map <- leaflet() %>%
  addTiles() %>%
  setView(-122.4036, 37.75059, zoom = 11) %>%
  addCircleMarkers(data = taSF, lng = ~ zipLon, lat = ~ zipLat, radius = 7,
      color = ~ pal(dollars), popup = ~ paste(venue, " - ",
      round(avgRating, digits = 2), " (", dollars , ")", sep = "")) %>%
  addLegend("bottomright", pal = pal, values = taSF$dollars, title = "Trip Advisor Pr
ice Range", opacity = 1)
taSF_map
```



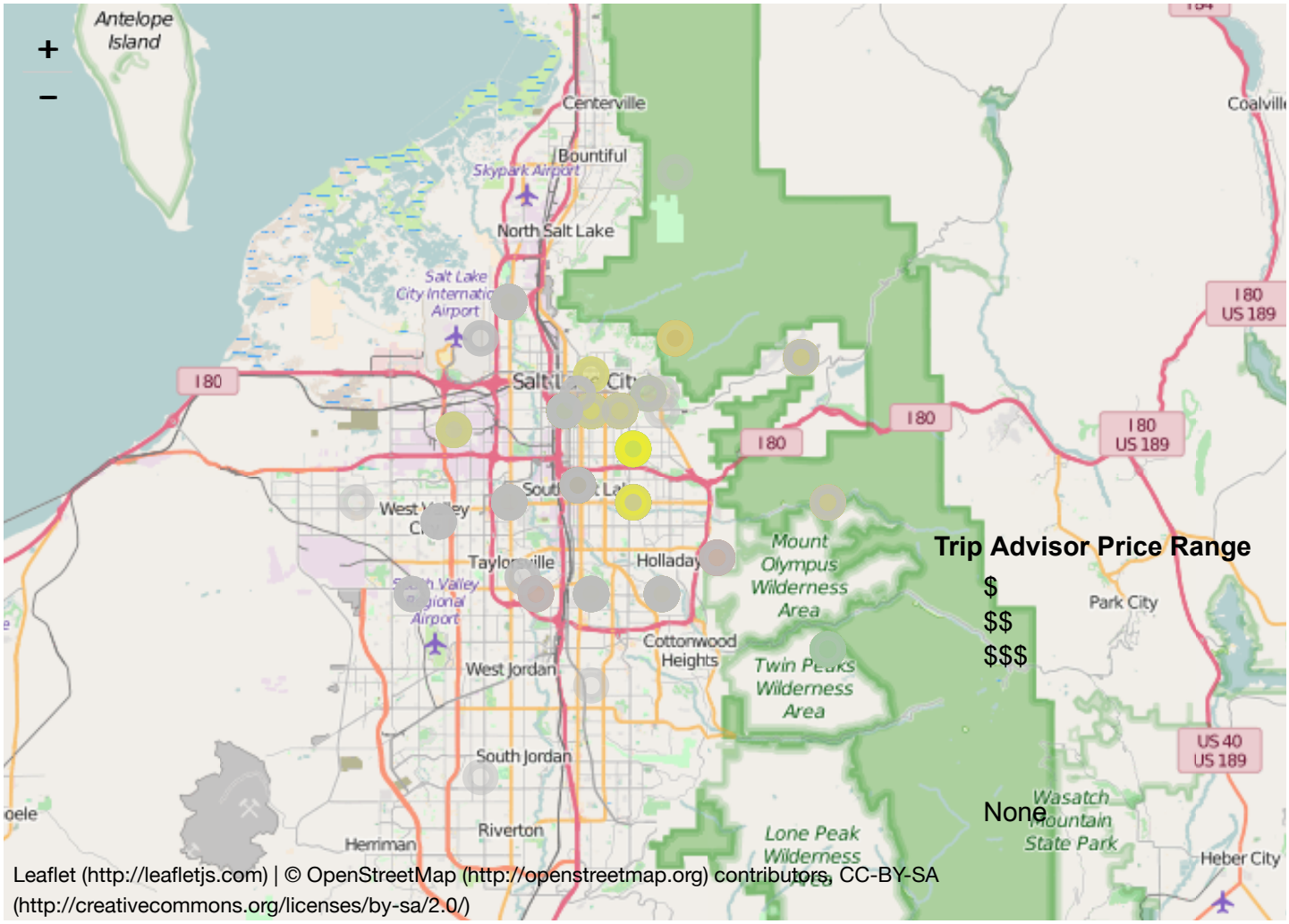

```
# Salt Lake City Leaflet map based on County lat/lon
mean(taSLC$zipLat)
```

```
## [1] 40.72247
```

```
mean(taSLC$zipLon)
```

```
## [1] -111.8423
```

```
taSLC_map <- leaflet() %>%
  addTiles() %>%
  setView(-111.8423, 40.72247, zoom = 10) %>%
  addCircleMarkers(data = taSLC, lng = ~ zipLon, lat = ~ zipLat, radius = 7,
      color = ~ pal(dollars), popup = ~ paste(venue, " - ",
      round(avgRating, digits = 2), " (", dollars , ")", sep = "")) %>%
  addLegend("bottomright", pal = pal, values = taSLC$dollars, title = "Trip Advisor P
rice Range", opacity = 1)
taSLC_map
```



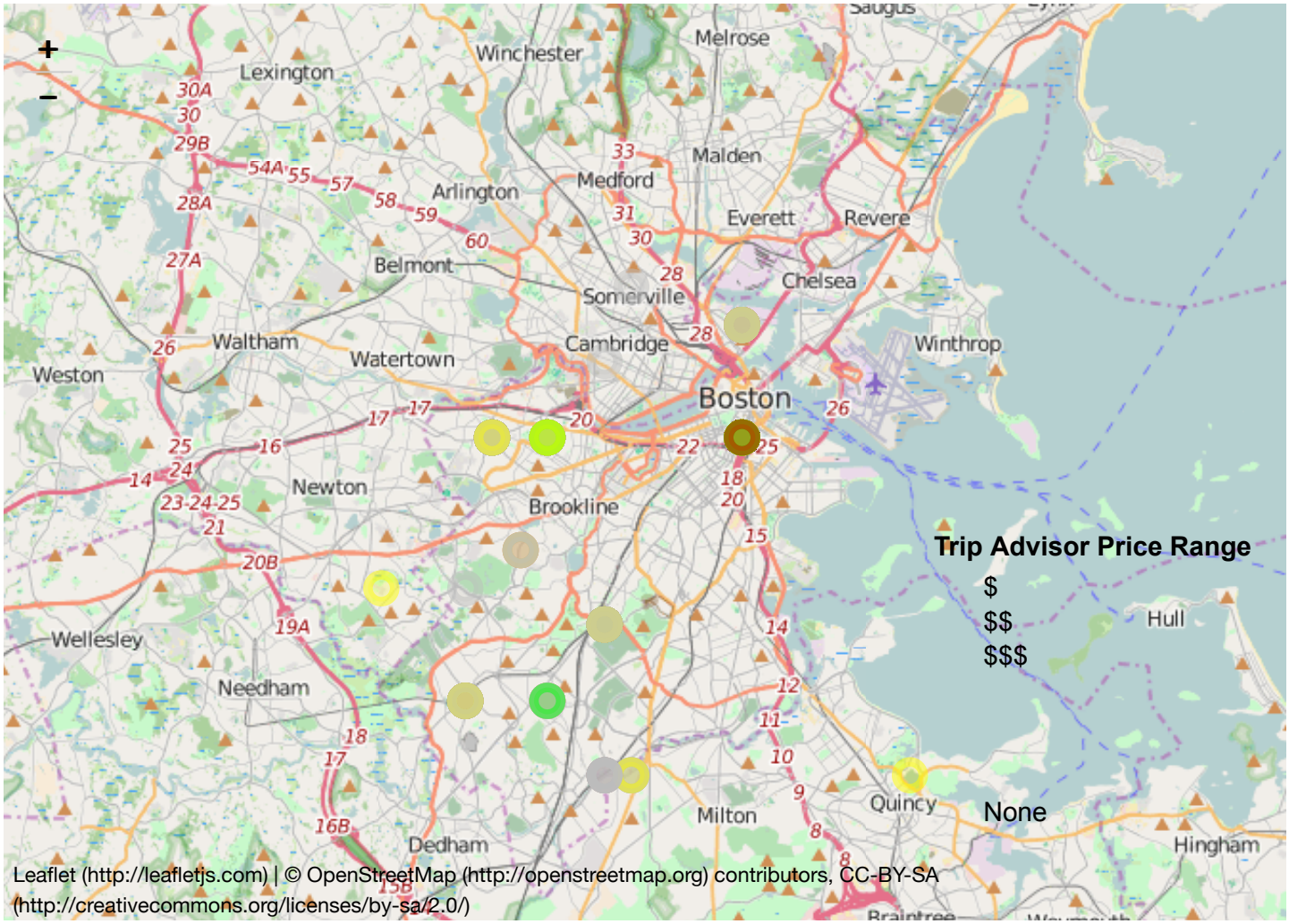

```
# Boston Leaflet map based on County lat/lon
mean(taBos$zipLat)
```

```
## [1] 42.34353
```

```
mean(taBos$zipLon)
```

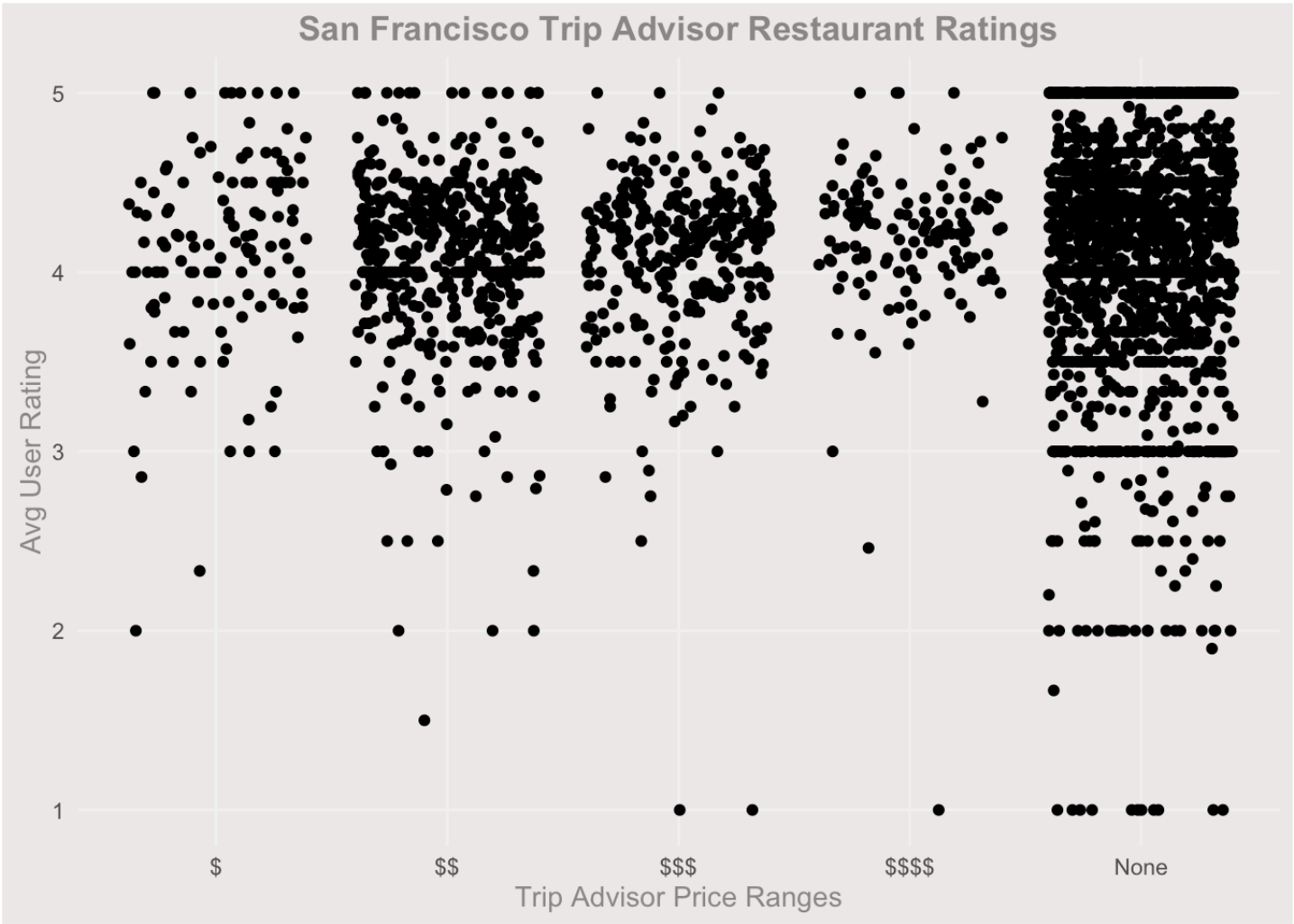
```
## [1] -71.09505
```

```
taBos_map <- leaflet() %>%
  addTiles() %>%
  setView(-71.09502, 42.34353, zoom = 11) %>%
  addCircleMarkers(data = taBos, lng = ~ zipLon, lat = ~ zipLat, radius = 7,
      color = ~ pal(dollars), popup = ~ paste(venue, " - ",
      round(avgRating, digits = 2), " (", dollars , ")", sep = "")) %>%
  addLegend("bottomright", pal = pal, values = taBos$dollars, title = "Trip Advisor P
rice Range", opacity = 1)
taBos_map
```

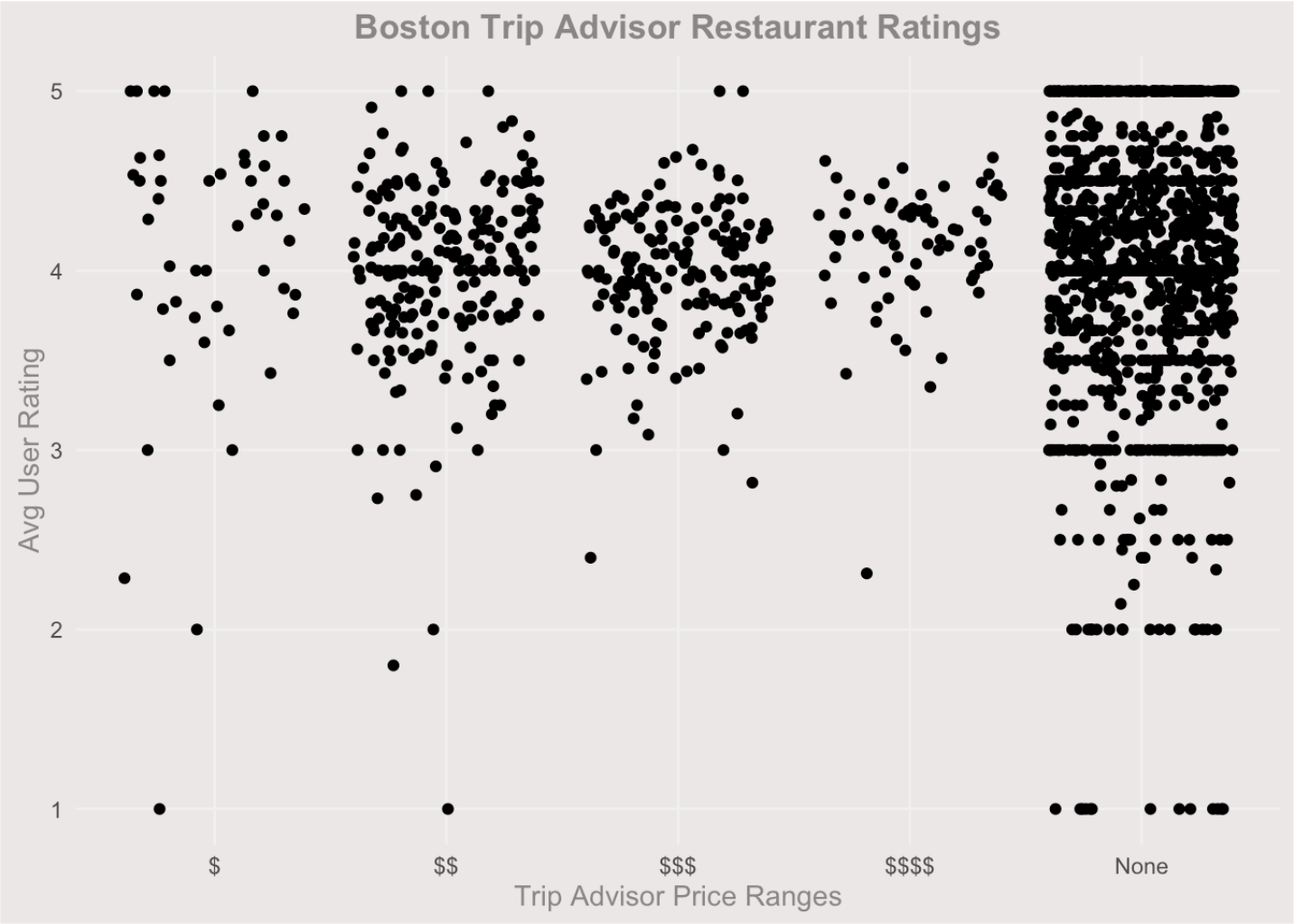


Charts and Averages

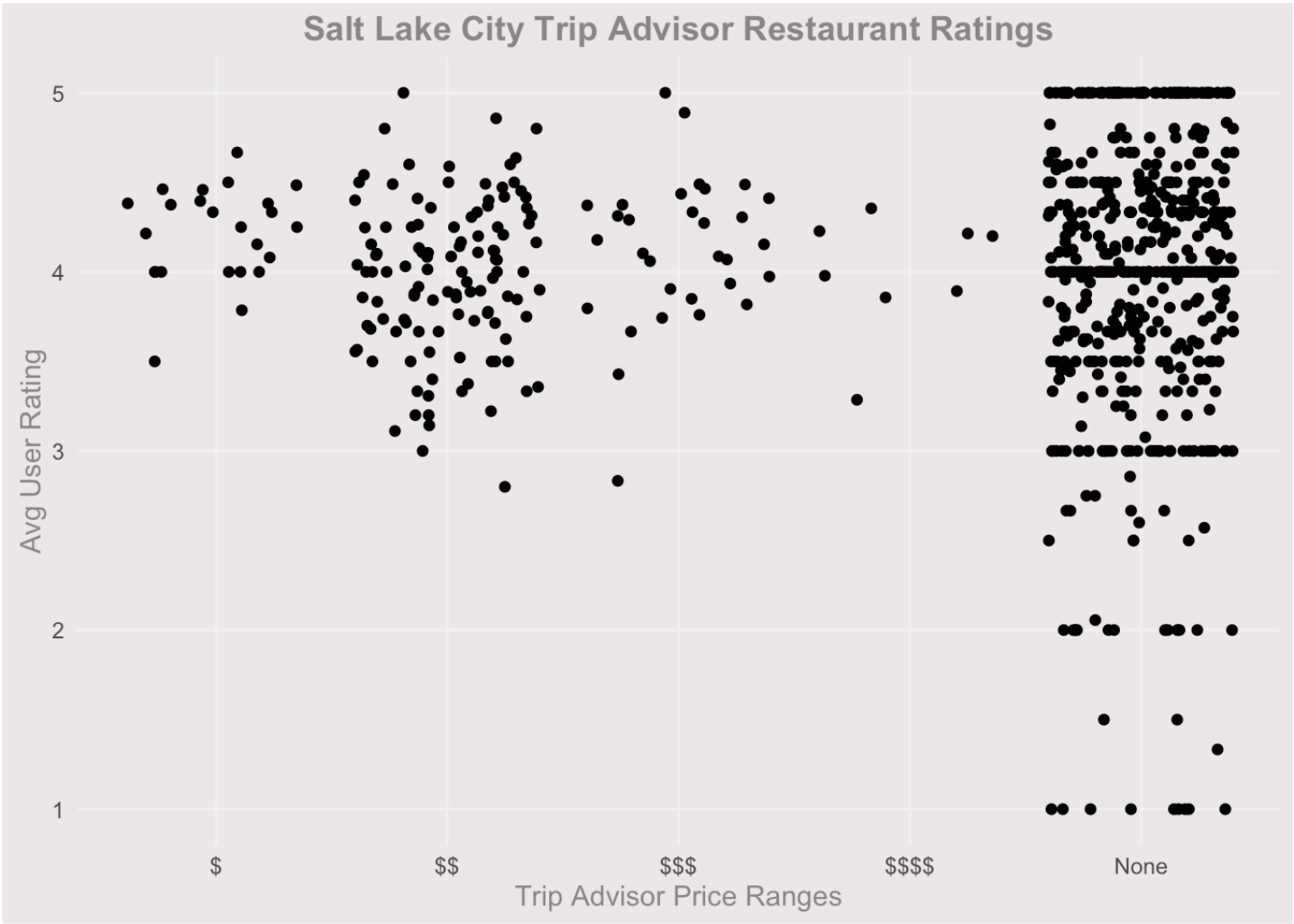
```
# Scatterplot
ggplot(data = taSF, aes(x = dollars, y = avgRating)) + geom_jitter() + labs(x = "Trip
Advisor Price Ranges", y = "Avg User Rating", title = "San Francisco Trip Advisor Res
taurant Ratings") + theme(
  axis.ticks =              element_blank(),
  axis.title =              element_text(color="snow4"),
  legend.position =         "bottom",
  legend.background =       element_blank(),
  legend.key =              element_blank(),
  panel.background =        element_blank(),
  panel.border =            element_blank(),
  panel.grid.major =        element_line(color="gray95"),
  panel.grid.minor =        element_blank(),
  plot.background =         element_rect(fill="snow2"),
  plot.title =              element_text(color="snow4", face = "bold"),
  strip.background =        element_rect(fill = "snow2"),
  strip.text =              element_text(size = rel(1.3), face = "bold")
)
```



```
ggplot(data = taBos, aes(x = dollars, y = avgRating)) + geom_jitter() + labs(x = "Tri
p Advisor Price Ranges", y = "Avg User Rating", title = "Boston Trip Advisor Restaura
nt Ratings") + theme(
  axis.ticks =              element_blank(),
  axis.title =              element_text(color="snow4"),
  legend.position =         "bottom",
  legend.background =       element_blank(),
  legend.key =              element_blank(),
  panel.background =        element_blank(),
  panel.border =            element_blank(),
  panel.grid.major =        element_line(color="gray95"),
  panel.grid.minor =        element_blank(),
  plot.background =         element_rect(fill="snow2"),
  plot.title =              element_text(color="snow4", face = "bold"),
  strip.background =        element_rect(fill = "snow2"),
  strip.text =              element_text(size = rel(1.3), face = "bold")
)
```

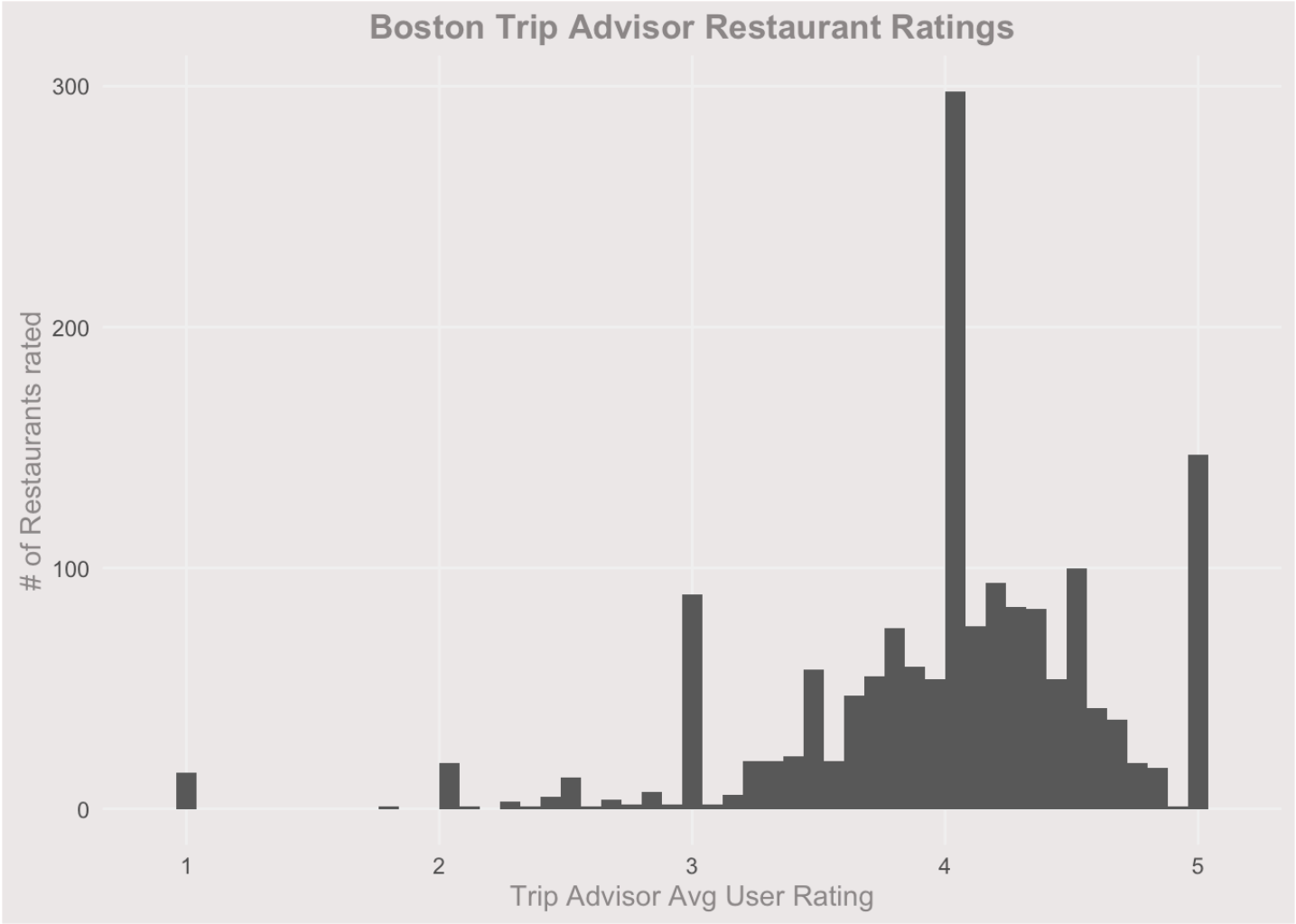


```
ggplot(data = taSLC, aes(x = dollars, y = avgRating)) + geom_jitter() + labs(x = "Tri
p Advisor Price Ranges", y = "Avg User Rating", title = "Salt Lake City Trip Advisor
Restaurant Ratings") + theme(
  axis.ticks =              element_blank(),
  axis.title =              element_text(color="snow4"),
  legend.position =         "bottom",
  legend.background =       element_blank(),
  legend.key =              element_blank(),
  panel.background =        element_blank(),
  panel.border =            element_blank(),
  panel.grid.major =        element_line(color="gray95"),
  panel.grid.minor =        element_blank(),
  plot.background =         element_rect(fill="snow2"),
  plot.title =              element_text(color="snow4", face = "bold"),
  strip.background =        element_rect(fill = "snow2"),
  strip.text =              element_text(size = rel(1.3), face = "bold")
)
```

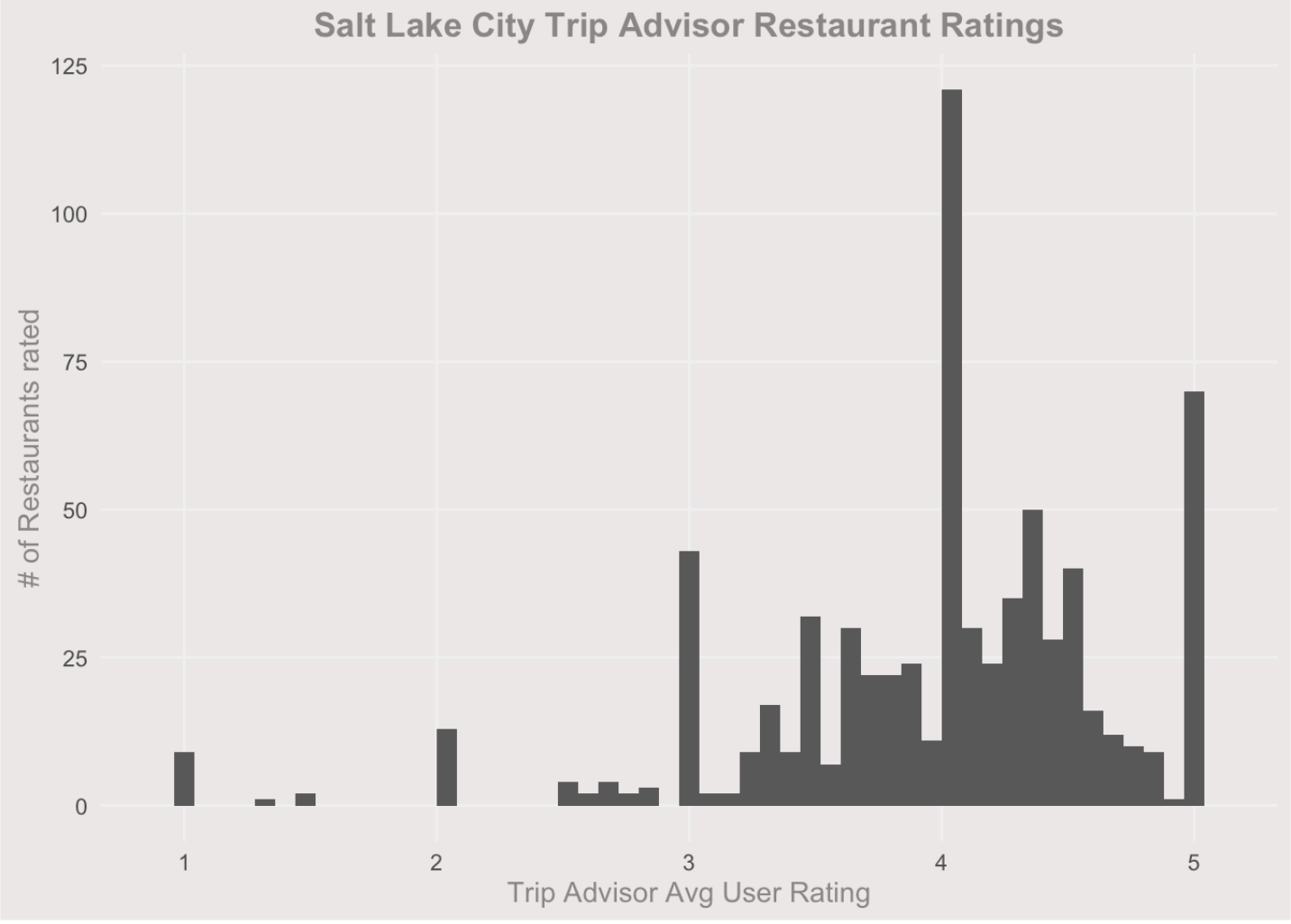


```
# Histogram
ggplot(data = taSF, aes(x = avgRating)) + geom_histogram(bins = 50) + labs(x = "Trip
Advisor Avg User Rating", y = "# of Restaurants rated", title = "San Francisco Trip A
dvisor Restaurant Ratings") + theme(
  axis.ticks =              element_blank(),
  axis.title =              element_text(color="snow4"),
  legend.position =         "bottom",
  legend.background =       element_blank(),
  legend.key =              element_blank(),
  panel.background =        element_blank(),
  panel.border =            element_blank(),
  panel.grid.major =        element_line(color="gray95"),
  panel.grid.minor =        element_blank(),
  plot.background =         element_rect(fill="snow2"),
  plot.title =              element_text(color="snow4", face = "bold"),
  strip.background =        element_rect(fill = "snow2"),
  strip.text =              element_text(size = rel(1.3), face = "bold")
)
```

```
ggplot(data = taBos, aes(x = avgRating)) + geom_histogram(bins = 50) + labs(x = "Trip
Advisor Avg User Rating", y = "# of Restaurants rated", title = "Boston Trip Advisor
Restaurant Ratings") + theme(
  axis.ticks =              element_blank(),
  axis.title =              element_text(color="snow4"),
  legend.position =         "bottom",
  legend.background =       element_blank(),
  legend.key =              element_blank(),
  panel.background =        element_blank(),
  panel.border =            element_blank(),
  panel.grid.major =        element_line(color="gray95"),
  panel.grid.minor =        element_blank(),
  plot.background =         element_rect(fill="snow2"),
  plot.title =              element_text(color="snow4", face = "bold"),
  strip.background =        element_rect(fill = "snow2"),
  strip.text =              element_text(size = rel(1.3), face = "bold")
)
```



```
ggplot(data = taSLC, aes(x = avgRating)) + geom_histogram(bins = 50) + labs(x = "Trip
Advisor Avg User Rating", y = "# of Restaurants rated", title = "Salt Lake City Trip
Advisor Restaurant Ratings") + theme(
  axis.ticks =              element_blank(),
  axis.title =              element_text(color="snow4"),
  legend.position =         "bottom",
  legend.background =       element_blank(),
  legend.key =              element_blank(),
  panel.background =        element_blank(),
  panel.border =            element_blank(),
  panel.grid.major =        element_line(color="gray95"),
  panel.grid.minor =        element_blank(),
  plot.background =         element_rect(fill="snow2"),
  plot.title =              element_text(color="snow4", face = "bold"),
  strip.background =        element_rect(fill = "snow2"),
  strip.text =              element_text(size = rel(1.3), face = "bold")
)
```



Bag of Words technique - SF only

```
corpusSF <- Corpus(VectorSource(taSF$cuisine))
corpusSF[[1]]$content
```

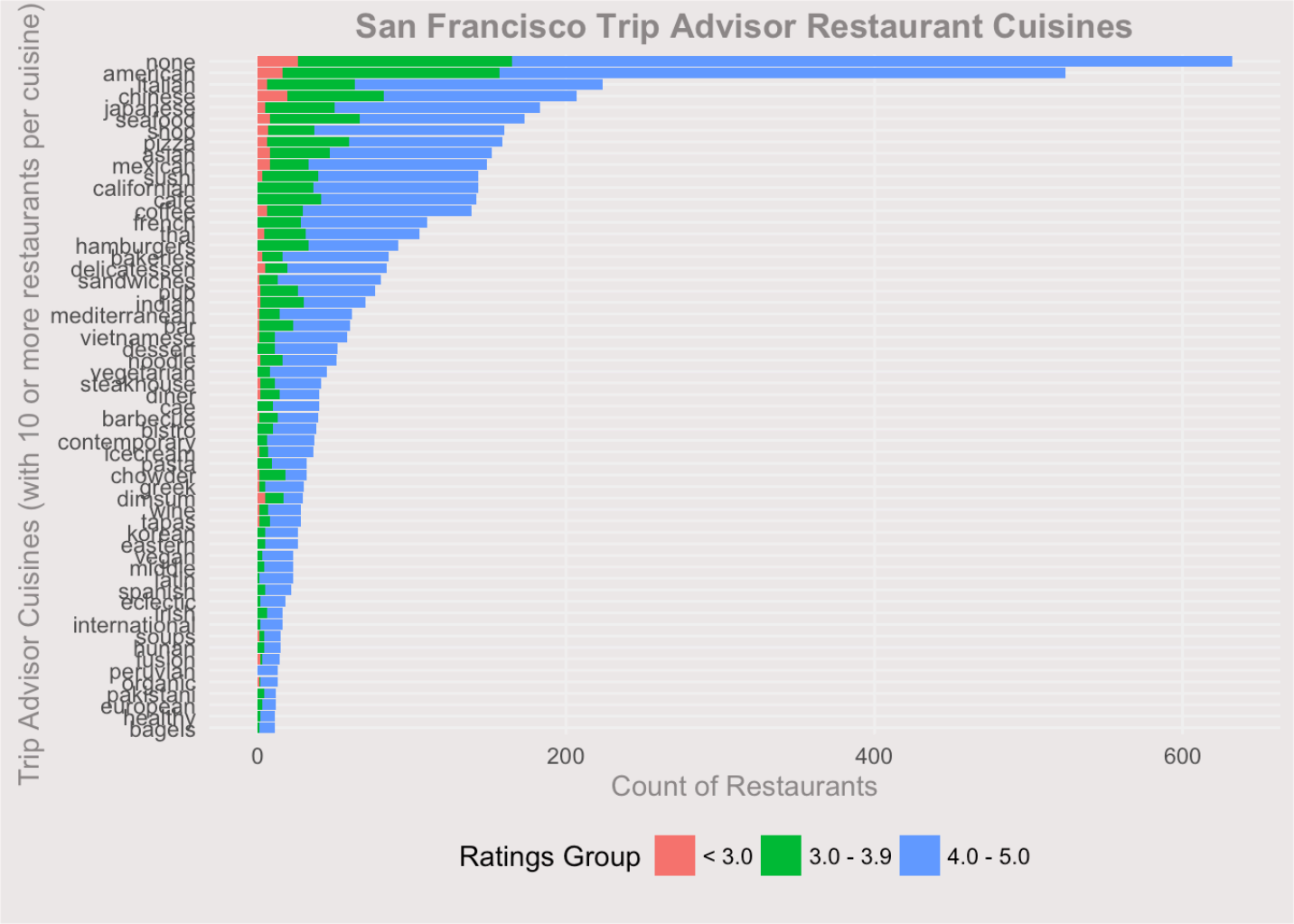
```
## [1] "Mediterranean"
```

```
corpusSF <- tm_map(corpusSF, PlainTextDocument)
freqSFCuis <- DocumentTermMatrix(corpusSF)
findFreqTerms(freqSFCuis, lowfreq = 20)
```

```
##  [1] "american"      "asian"         "bakeries"      "bar"
##  [5] "barbecue"      "bistro"        "cae"           "cafe"
##  [9] "californian"   "chinese"       "chowder"       "coffee"
## [13] "contemporary"  "delicatessen"  "dessert"       "dimsum"
## [17] "diner"         "eastern"       "french"        "greek"
## [21] "hamburgers"    "icecream"      "indian"        "italian"
## [25] "japanese"      "korean"        "latin"         "mediterranean"
## [29] "mexican"       "middle"        "none"          "noodle"
## [33] "pasta"         "pizza"         "pub"           "sandwiches"
## [37] "seafood"       "shop"          "spanish"       "steakhouse"
## [41] "sushi"         "tapas"         "thai"          "vegan"
## [45] "vegetarian"    "vietnamese"    "wine"
```

```
taSFCuis = as.data.frame(as.matrix(freqSFCuis), stringsAsFactors = FALSE)

# Arrange into a dataframe
rownames(taSFCuis) <- seq(1,nrow(taSFCuis),1)
colnames(taSFCuis) <- make.names(colnames(taSFCuis))
taSFCuis <- cbind(taSF, taSFCuis)
taSFCuisLower <- taSFCuis %>% filter(avgRating < 3.0)
taSFCuisGood <- taSFCuis %>% filter(avgRating >= 3.0 & avgRating < 4.0)
taSFCuisGreat <- taSFCuis %>% filter(avgRating >= 4.0 & avgRating <= 5.0)

# nrow(taSFCuisGreat) + nrow(taSFCuisGood) + nrow(taSFCuisLower)

# Summarize cuisines by restaurant count, and rating groups
listSFCuis <- as.matrix(colnames(taSFCuis[14:ncol(taSFCuis)]))
listSFCuisSum <- ''
listRatedLower <- ''
listRatedGood <- ''
listRatedGreat <- ''
for (i in 14:ncol(taSFCuis)) {listSFCuisSum[i - 13] <- length(which(taSFCuis[,i] > 0)
)}
for (i in 14:ncol(taSFCuisLower)) {listRatedLower[i - 13] <- length(which(taSFCuisLow
er[,i] > 0))}
for (i in 14:ncol(taSFCuisGood)) {listRatedGood[i - 13] <- length(which(taSFCuisGood[
,i] > 0))}
for (i in 14:ncol(taSFCuisGreat)) {listRatedGreat[i - 13] <- length(which(taSFCuisGre
```

```
at[,i] > 0))}

# Clean up data columns and prepare for charting
listSFCuis <- cbind(listSFCuis, listSFCuisSum, listRatedLower, listRatedGood, listRat
edGreat)
listSFCuis <- as.data.frame(listSFCuis, stringsAsFactors = FALSE)
colnames(listSFCuis) <- c("cuisine","countOfRestaurant", "lowerthan3", "between3_4",
"between4_5")
listSFCuis$countOfRestaurant <- as.numeric(listSFCuis$countOfRestaurant)
listSFCuis$lowerthan3 <- as.numeric(listSFCuis$lowerthan3)
listSFCuis$between3_4 <- as.numeric(listSFCuis$between3_4)
listSFCuis$between4_5 <- as.numeric(listSFCuis$between4_5)
rm(listSFCuisSum, listRatedLower, listRatedGood, listRatedGreat)

# Arrange factors
listSFCuis <- arrange(listSFCuis, desc(countOfRestaurant))
listSFCuis$cuisine <- factor(listSFCuis$cuisine, levels = listSFCuis$cuisine[order(li
stSFCuis$countOfRestaurant)])

# Set up for stacked bar chart
listSFCuis2 <- listSFCuis %>% gather("ratingGrp", "restCount", 3:5)
listSFCuis2$ratingGrp <- gsub("lowerthan3","< 3.0", listSFCuis2$ratingGrp)
listSFCuis2$ratingGrp <- gsub("between3_4","3.0 - 3.9", listSFCuis2$ratingGrp)
listSFCuis2$ratingGrp <- gsub("between4_5","4.0 - 5.0", listSFCuis2$ratingGrp)

# Cuisine and # of restaurants
ggplot(data = filter(listSFCuis2, countOfRestaurant >= 10), aes(x = cuisine, y = rest
Count, fill = ratingGrp)) + geom_bar(stat = "identity") + coord_flip() + labs(y = "Co
unt of Restaurants", x = "Trip Advisor Cuisines (with 10 or more restaurants per cuis
ine)", title = "San Francisco Trip Advisor Restaurant Cuisines", fill = "Ratings Grou
p") + theme(
  axis.ticks =              element_blank(),
  axis.title =              element_text(color="snow4"),
  legend.position =         "bottom",
  legend.background =       element_blank(),
  legend.key =              element_blank(),
  panel.background =        element_blank(),
  panel.border =            element_blank(),
  panel.grid.major =        element_line(color="gray95"),
  panel.grid.minor =        element_blank(),
  plot.background =         element_rect(fill="snow2"),
  plot.title =              element_text(color="snow4", face = "bold"),
  strip.background =        element_rect(fill = "snow2"),
  strip.text =              element_text(size = rel(1.3), face = "bold")
)
```



Bag of Words technique - SLC only

```
corpusSLC <- Corpus(VectorSource(taSLC$cuisine))
corpusSLC[[1]]$content
```

```
## [1] "Chinese Japanese Asian TeaRoom"
```

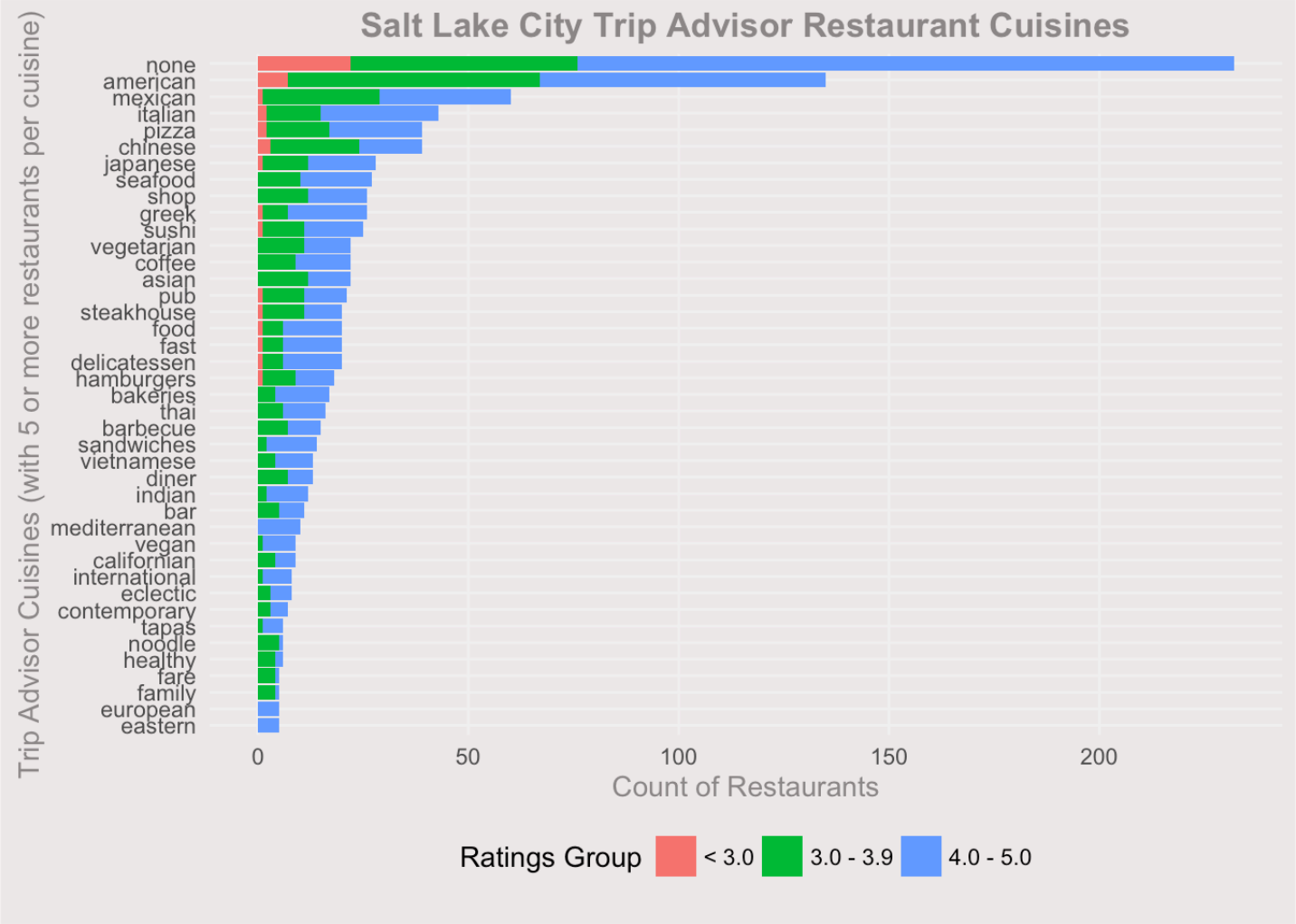
```
corpusSLC <- tm_map(corpusSLC, PlainTextDocument)
freqSLCCuis <- DocumentTermMatrix(corpusSLC)
findFreqTerms(freqSLCCuis, lowfreq = 20)
```

```
##  [1] "american"     "asian"        "chinese"      "coffee"
##  [5] "delicatessen" "fast"         "food"         "greek"
##  [9] "italian"      "japanese"     "mexican"      "none"
## [13] "pizza"        "pub"          "seafood"      "shop"
## [17] "steakhouse"   "sushi"        "vegetarian"
```

```
taSLCCuis = as.data.frame(as.matrix(freqSLCCuis), stringsASLCactors = FALSE)
```

```
# Arrange into a dataframe
rownames(taSLCCuis) <- seq(1,nrow(taSLCCuis),1)
colnames(taSLCCuis) <- make.names(colnames(taSLCCuis))
taSLCCuis <- cbind(taSLC, taSLCCuis)
taSLCCuisLower <- taSLCCuis %>% filter(avgRating < 3.0)
taSLCCuisGood <- taSLCCuis %>% filter(avgRating >= 3.0 & avgRating < 4.0)
taSLCCuisGreat <- taSLCCuis %>% filter(avgRating >= 4.0 & avgRating <= 5.0)

# nrow(taSLCCuisGreat) + nrow(taSLCCuisGood) + nrow(taSLCCuisLower)

# Summarize cuisines by restaurant count, and rating groups
listSLCCuis <- as.matrix(colnames(taSLCCuis[14:ncol(taSLCCuis)]))
listSLCCuisSum <- ''
listRatedLower <- ''
listRatedGood <- ''
listRatedGreat <- ''
for (i in 14:ncol(taSLCCuis)) {listSLCCuisSum[i - 13] <- length(which(taSLCCuis[,i] >
0))}
for (i in 14:ncol(taSLCCuisLower)) {listRatedLower[i - 13] <- length(which(taSLCCuisL
ower[,i] > 0))}
for (i in 14:ncol(taSLCCuisGood)) {listRatedGood[i - 13] <- length(which(taSLCCuisGoo
d[,i] > 0))}
for (i in 14:ncol(taSLCCuisGreat)) {listRatedGreat[i - 13] <- length(which(taSLCCuisG
reat[,i] > 0))}

# length(which(taSLCCuisLower[,99] > 0))

# Clean up data columns and prepare for charting
listSLCCuis <- cbind(listSLCCuis, listSLCCuisSum, listRatedLower, listRatedGood, list
RatedGreat)
listSLCCuis <- as.data.frame(listSLCCuis, stringsAsFactors = FALSE)
colnames(listSLCCuis) <- c("cuisine","countOfRestaurant", "lowerthan3", "between3_4",
"between4_5")
listSLCCuis$countOfRestaurant <- as.numeric(listSLCCuis$countOfRestaurant)
listSLCCuis$lowerthan3 <- as.numeric(listSLCCuis$lowerthan3)
listSLCCuis$between3_4 <- as.numeric(listSLCCuis$between3_4)
listSLCCuis$between4_5 <- as.numeric(listSLCCuis$between4_5)
rm(listSLCCuisSum, listRatedLower, listRatedGood, listRatedGreat)

# Arrange factors
listSLCCuis <- arrange(listSLCCuis, desc(countOfRestaurant))
listSLCCuis$cuisine <- factor(listSLCCuis$cuisine, levels = listSLCCuis$cuisine[order
(listSLCCuis$countOfRestaurant)])

# Set up for stacked bar chart
listSLCCuis2 <- listSLCCuis %>% gather("ratingGrp", "restCount", 3:5)
listSLCCuis2$ratingGrp <- gsub("lowerthan3","< 3.0", listSLCCuis2$ratingGrp)
listSLCCuis2$ratingGrp <- gsub("between3_4","3.0 - 3.9", listSLCCuis2$ratingGrp)
listSLCCuis2$ratingGrp <- gsub("between4_5","4.0 - 5.0", listSLCCuis2$ratingGrp)
```

```
# Cuisine and # of restaurants
ggplot(data = filter(listSLCCuis2, countOfRestaurant >= 5), aes(x = cuisine, y = rest
Count, fill = ratingGrp)) + geom_bar(stat = "identity") + coord_flip() + labs(y = "Co
unt of Restaurants", x = "Trip Advisor Cuisines (with 5 or more restaurants per cuisi
ne)", title = "Salt Lake City Trip Advisor Restaurant Cuisines", fill = "Ratings Grou
p") + theme(
  axis.ticks =              element_blank(),
  axis.title =              element_text(color="snow4"),
  legend.position =         "bottom",
  legend.background =       element_blank(),
  legend.key =              element_blank(),
  panel.background =        element_blank(),
  panel.border =            element_blank(),
  panel.grid.major =        element_line(color="gray95"),
  panel.grid.minor =        element_blank(),
  plot.background =         element_rect(fill="snow2"),
  plot.title =              element_text(color="snow4", face = "bold"),
  strip.background =        element_rect(fill = "snow2"),
  strip.text =              element_text(size = rel(1.3), face = "bold")
)
```



Bag of Words technique - Bos only

```
corpusBos <- Corpus(VectorSource(taBos$cuisine))
corpusBos[[1]]$content
```

```
## [1] "Italian Pizza Pizza&Pasta"
```

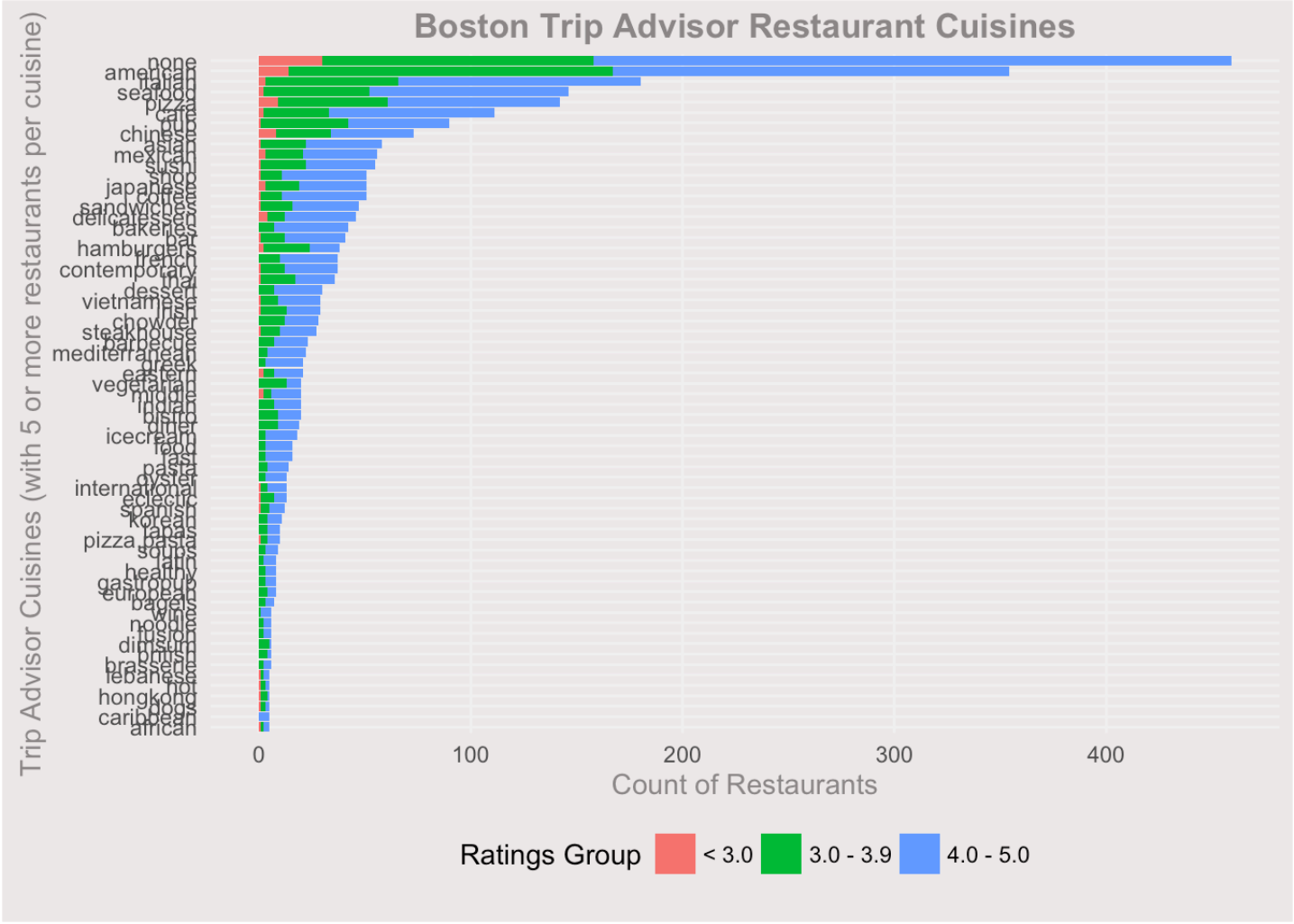
```
corpusBos <- tm_map(corpusBos, PlainTextDocument)
freqBosCuis <- DocumentTermMatrix(corpusBos)
findFreqTerms(freqBosCuis, lowfreq = 20)
```

```
##  [1] "american"      "asian"         "bakeries"      "bar"
##  [5] "barbecue"      "bistro"        "cafe"          "chinese"
##  [9] "chowder"       "coffee"        "contemporary"  "delicatessen"
## [13] "dessert"       "eastern"       "french"        "greek"
## [17] "hamburgers"    "indian"        "irish"         "italian"
## [21] "japanese"      "mediterranean" "mexican"       "middle"
## [25] "none"          "pizza"         "pub"           "sandwiches"
## [29] "seafood"       "shop"          "steakhouse"    "sushi"
## [33] "thai"          "vegetarian"    "vietnamese"
```

```
taBosCuis = as.data.frame(as.matrix(freqBosCuis), stringsABosactors = FALSE)

# Arrange into a dataframe
rownames(taBosCuis) <- seq(1,nrow(taBosCuis),1)
colnames(taBosCuis) <- make.names(colnames(taBosCuis))
taBosCuis <- cbind(taBos, taBosCuis)
taBosCuisLower <- taBosCuis %>% filter(avgRating < 3.0)
taBosCuisGood <- taBosCuis %>% filter(avgRating >= 3.0 & avgRating < 4.0)
taBosCuisGreat <- taBosCuis %>% filter(avgRating >= 4.0 & avgRating <= 5.0)

# nrow(taBosCuisGreat) + nrow(taBosCuisGood) + nrow(taBosCuisLower)

# Summarize cuisines by restaurant count, and rating groups
listBosCuis <- as.matrix(colnames(taBosCuis[14:ncol(taBosCuis)]))
listBosCuisSum <- ''
listRatedLower <- ''
listRatedGood <- ''
listRatedGreat <- ''
for (i in 14:ncol(taBosCuis)) {listBosCuisSum[i - 13] <- length(which(taBosCuis[,i] >
0))}
for (i in 14:ncol(taBosCuisLower)) {listRatedLower[i - 13] <- length(which(taBosCuisL
ower[,i] > 0))}
for (i in 14:ncol(taBosCuisGood)) {listRatedGood[i - 13] <- length(which(taBosCuisGoo
d[,i] > 0))}
for (i in 14:ncol(taBosCuisGreat)) {listRatedGreat[i - 13] <- length(which(taBosCuisG
reat[,i] > 0))}

# length(which(taBosCuisLower[,99] > 0))

# Clean up data columns and prepare for charting
listBosCuis <- cbind(listBosCuis, listBosCuisSum, listRatedLower, listRatedGood, list
RatedGreat)
listBosCuis <- as.data.frame(listBosCuis, stringsAsFactors = FALSE)
colnames(listBosCuis) <- c("cuisine","countOfRestaurant", "lowerthan3", "between3_4",
"between4_5")
listBosCuis$countOfRestaurant <- as.numeric(listBosCuis$countOfRestaurant)
listBosCuis$lowerthan3 <- as.numeric(listBosCuis$lowerthan3)
listBosCuis$between3_4 <- as.numeric(listBosCuis$between3_4)
listBosCuis$between4_5 <- as.numeric(listBosCuis$between4_5)
rm(listBosCuisSum, listRatedLower, listRatedGood, listRatedGreat)

# Arrange factors
listBosCuis <- arrange(listBosCuis, desc(countOfRestaurant))
listBosCuis$cuisine <- factor(listBosCuis$cuisine, levels = listBosCuis$cuisine[order
(listBosCuis$countOfRestaurant)])
glimpse(listBosCuis)
```

```
## Observations: 119
## Variables: 5
## $ cuisine           (fctr) none, american, italian, seafood, pizza, ca...
## $ countOfRestaurant (dbl) 459, 354, 180, 146, 142, 111, 90, 73, 58, 56...
## $ lowerthan3        (dbl) 30, 14, 3, 2, 9, 2, 1, 8, 1, 3, 1, 1, 3, 1, ...
## $ between3_4        (dbl) 128, 153, 63, 50, 52, 31, 41, 26, 21, 18, 21...
## $ between4_5        (dbl) 301, 187, 114, 94, 81, 78, 48, 39, 36, 35, 3...
```

```
# Set up for stacked bar chart
listBosCuis2 <- listBosCuis %>% gather("ratingGrp", "restCount", 3:5)
listBosCuis2$ratingGrp <- gsub("lowerthan3","< 3.0", listBosCuis2$ratingGrp)
listBosCuis2$ratingGrp <- gsub("between3_4","3.0 - 3.9", listBosCuis2$ratingGrp)
listBosCuis2$ratingGrp <- gsub("between4_5","4.0 - 5.0", listBosCuis2$ratingGrp)

# Cuisine and # of restaurants
ggplot(data = filter(listBosCuis2, countOfRestaurant >= 5), aes(x = cuisine, y = rest
Count, fill = ratingGrp)) + geom_bar(stat = "identity") + coord_flip() + labs(y = "Co
unt of Restaurants", x = "Trip Advisor Cuisines (with 5 or more restaurants per cuisi
ne)", title = "Boston Trip Advisor Restaurant Cuisines", fill = "Ratings Group") + th
eme(
  axis.ticks =              element_blank(),
  axis.title =              element_text(color="snow4"),
  legend.position =         "bottom",
  legend.background =       element_blank(),
  legend.key =              element_blank(),
  panel.background =        element_blank(),
  panel.border =            element_blank(),
  panel.grid.major =        element_line(color="gray95"),
  panel.grid.minor =        element_blank(),
  plot.background =         element_rect(fill="snow2"),
  plot.title =              element_text(color="snow4", face = "bold"),
  strip.background =        element_rect(fill = "snow2"),
  strip.text =              element_text(size = rel(1.3), face = "bold")
)
```


Boston Trip Advisor Restaurant Cuisines
Trip Advisor Cuisines (with 5 or more restaurants per cuisine)
none
american
italian
seafood
pizza
cafe
pub
chinese
asian
mexican
steak
sushi
shop
japanese
coffee
sandwiches
delicatessen
bakeries
bar
hamburgers
french
contemporary
italy
thai
dessert
vietnamese
irish
chowder
steakhouse
barbecue
mediterranean
greek
eastern
vegetarian
middle
indian
bistro
diner
icecream
food
fast
pasta
oyster
international
eclectic
spanish
korean
tapas
pizza pasta
sushi
soups
bbq
healthy
gastropub
european
bagels
wine
noodle
fusion
dimsum
british
brasserie
lebanese
thai
hongkong
tapas
caribbean
african
0
100
200
300
400
Count of Restaurants
Ratings Group
< 3.0
3.0 - 3.9
4.0 - 5.0