Classifying review data

Load libraries

```
library(tidyverse)
library(tidytext)
library(jsonlite)
library(ggmap)
```

```
Warning: package 'ggmap' was built under R version 4.2.3
```

```
i Google's Terms of Service: <https://mapsplatform.google.com>
  Stadia Maps' Terms of Service: <https://stadiamaps.com/terms-of-service/>
  OpenStreetMap's Tile Usage Policy: <https://operations.osmfoundation.org/policies/tiles/>
i Please cite ggmap if you use it! Use `citation("ggmap")` for details.
```

```
setwd("~/Documents/GitHub/small-business-sentiment-analysis/scripts")

source("ggplot_settings.R")
theme_set(theme_custom())
```

```
Warning: The `size` argument of `element_rect()` is deprecated as of ggplot2 3.4.0.
i Please use the `linewidth` argument instead.
```

Load metadata and review data

```
# loads `metadata` which is metadata of all reviews
load("../review_data/metadata.RData")

# load `reviews` which has all the 2019, 2020 and 2021 reviews
load("../review_data/reviews19_20_21.RData")

reviews_with_sentiment <- read_csv("../review_data/reviews_with_sentiment.csv")
```

```
New names:
Rows: 18087 Columns: 22
-- Column specification
-------------------------------------------------------- Delimiter: "," chr
(14): text, gmap_id, name, address, category, alias, categories, coordi... dbl
(7): ...1, Unnamed: 0, rating, month, year, polarity, subjectivity dttm (1):
time
i Use `spec()` to retrieve the full column specification for this data. i
Specify the column types or set `show_col_types = FALSE` to quiet this message.
* `` -> `...1`
```

Combine with minority-owned business data (not just restaurants)

```
small_businesses <- read_csv("../review_data/SBS_Certified_Business_List_20240309.csv")
small_businesses$name <- small_businesses$Vendor_Formal_Name
small_businesses <- small_businesses |> select(name, First_Name, Last_Name, Business_Descr

joined <- inner_join(small_businesses, metadata)
# 67 common establishments between the two!

asian_gmap_ids <- joined |>
  filter(Ethnicity == "ASIAN") |>
  pull(gmap_id)

# 2,383 rows for ASIAN
small_businesses |> filter(Ethnicity == "ASIAN")

# d |> filter(gmap_id %in% asian_gmap_ids)
```

Combine and select asian, mexican and pizza restaurants

```
# join review and cuisines to get smaller list of only asian restaurants
asian <- read_csv("../review_data/nyc_asian_restaurant.csv") |>
  select(name, alias, categories, coordinates, location)
```

```
Rows: 950 Columns: 17
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (12): id, alias, name, image_url, url, categories, coordinates, transact...
dbl  (4): review_count, rating, phone, distance
lgl  (1): is_closed

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
asian <- inner_join(metadata, asian, by = "name") |>
  distinct(gmap_id, .keep_all = TRUE)
```

```
Warning in inner_join(metadata, asian, by = "name"): Detected an unexpected many-to-many rela
i Row 61277 of `x` matches multiple rows in `y`.
i Row 777 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
# 440 rows after only keeping the first by name
# 533 rows after only keeping the first by gmap_id

d_asian <- inner_join(reviews, asian, by = "gmap_id") |>
  bind_cols(type = "asian")
# 113,116 for 2019, 2020 and 2021


# NOTE: SOPHIA LOOK HERE
# repeat for mexican and pizza restaurants
pizza <- read_csv("../review_data/nyc_pizza_restaurant.csv") |>
  select(name, alias, categories, coordinates, location)
```

```
Rows: 1000 Columns: 17
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (12): id, alias, name, image_url, url, categories, coordinates, transact...
dbl  (4): review_count, rating, phone, distance
lgl  (1): is_closed

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
pizza <- inner_join(metadata, pizza, by = "name") |>
  distinct(gmap_id, .keep_all = TRUE)
```

```
Warning in inner_join(metadata, pizza, by = "name"): Detected an unexpected many-to-many rela
i Row 153 of `x` matches multiple rows in `y`.
i Row 437 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
d_pizza <- inner_join(reviews, pizza, by = "gmap_id") |>
  bind_cols(type = "pizza")

mex <- read_csv("../review_data/nyc_mexican_restaurant.csv") |>
  select(name, alias, categories, coordinates, location)
```

```
Rows: 1000 Columns: 17
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (12): id, alias, name, image_url, url, categories, coordinates, transact...
dbl  (4): review_count, rating, phone, distance
lgl  (1): is_closed

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

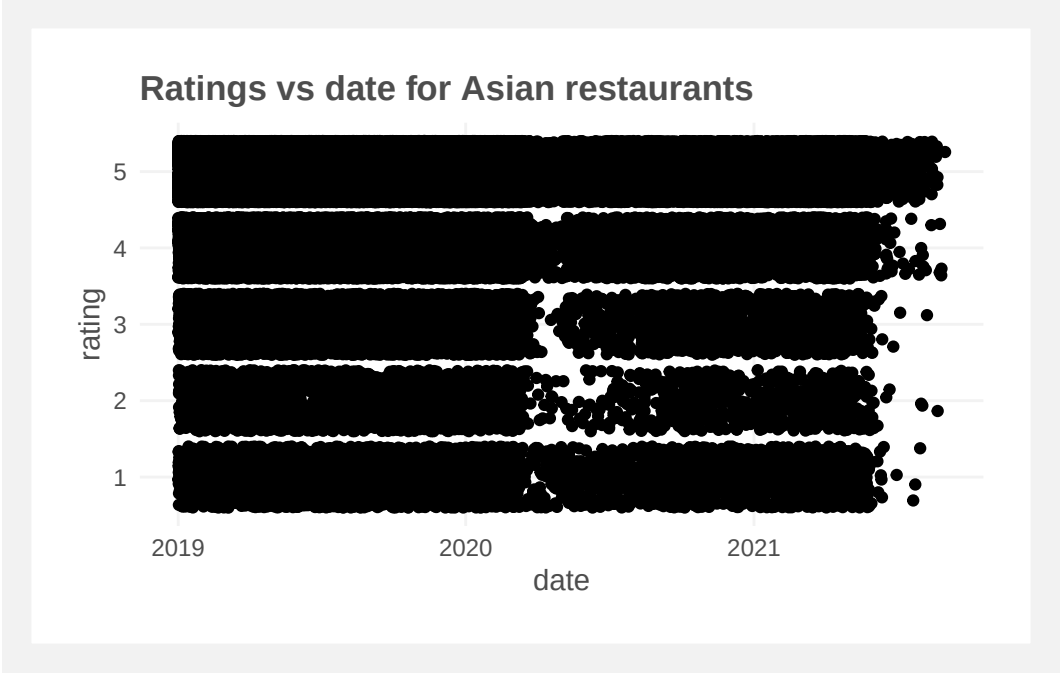
```
mex <- inner_join(metadata, mex, by = "name") |>
  distinct(gmap_id, .keep_all = TRUE)
```

```
Warning in inner_join(metadata, mex, by = "name"): Detected an unexpected many-to-many relati
i Row 3254 of `x` matches multiple rows in `y`.
i Row 757 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```r
d_mex <- inner_join(reviews, mex, by = "gmap_id") |>
  bind_cols(type = "mexican")

combined_reviews <- bind_rows(d_asian, d_pizza, d_mex) |>
  mutate(date = as.Date(paste(year, month, 01), "%Y %m %d"))

# save(combined_reviews, file = "../review_data/combined_reviews.RData")
```

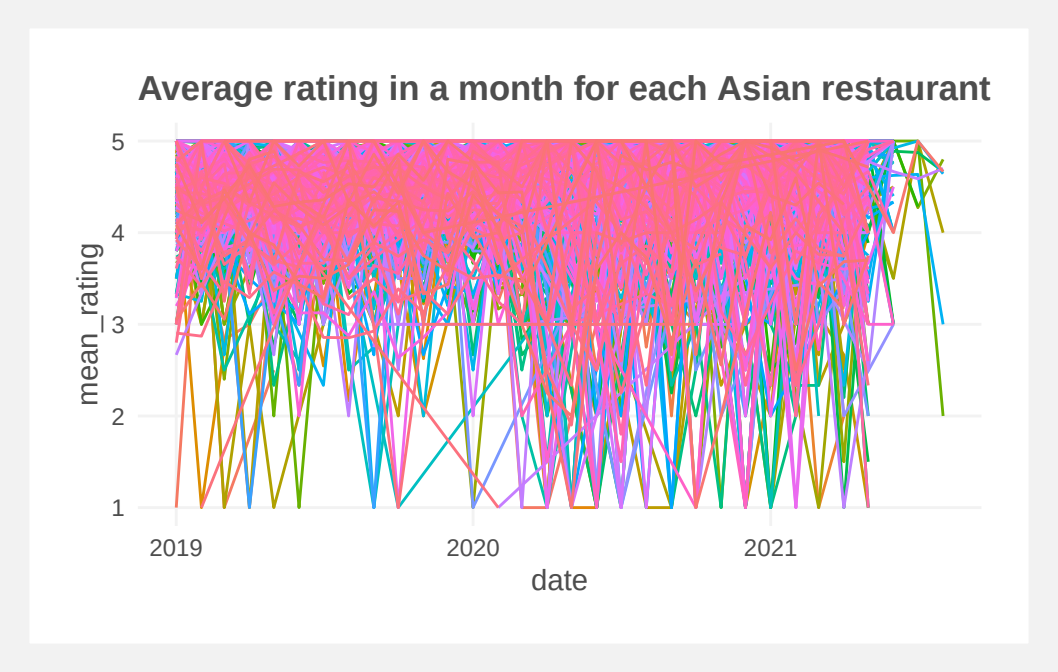
Plotting

```r
covid_start <- as.Date(paste(2020, 03, 01), "%Y %m %d")
stopah_start <- as.Date(paste(2021, 03, 01), "%Y %m %d")

ggplot(d_asian, aes(x = time, y = rating)) +
  geom_jitter() +
  labs(x = "date", title = "Ratings vs date for Asian restaurants")
```



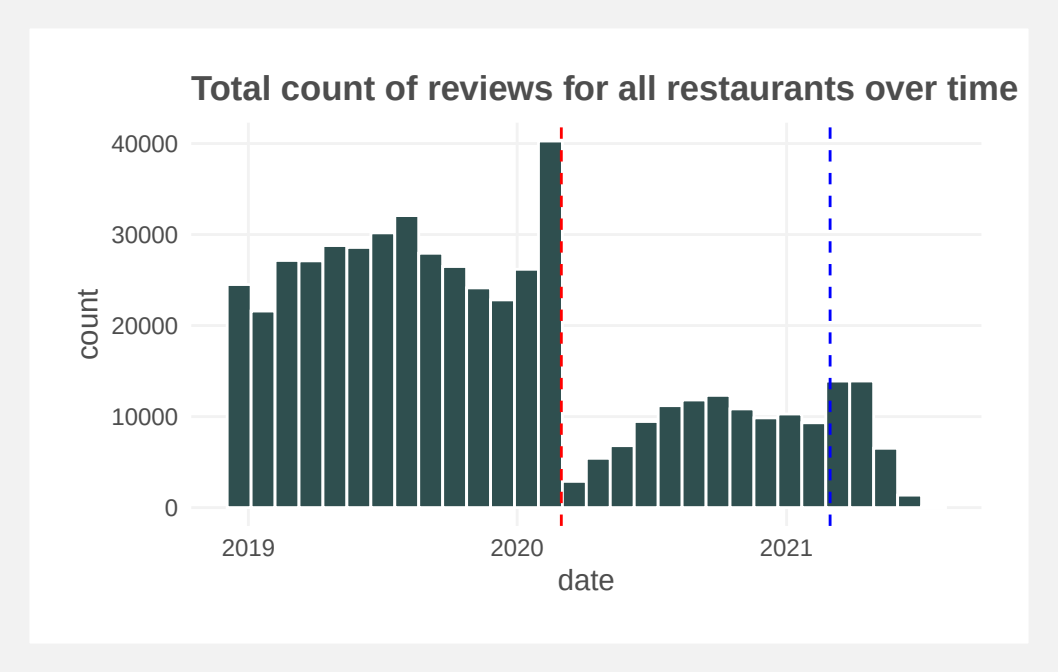

```
combined_reviews |>
  filter(type == "asian") |>
  group_by(gmap_id, month, year) |>
  summarize(mean_rating = mean(rating)) |>
  mutate(date = as.Date(paste(year, month, 01), "%Y %m %d")) |>
  ggplot(aes(x = date, y = mean_rating, color = gmap_id)) +
  geom_line() +
  theme(legend.position = "none") +
  labs(title = "Average rating in a month for each Asian restaurant")
```

```
`summarise()` has grouped output by 'gmap_id', 'month'. You can override using
the `.groups` argument.
```

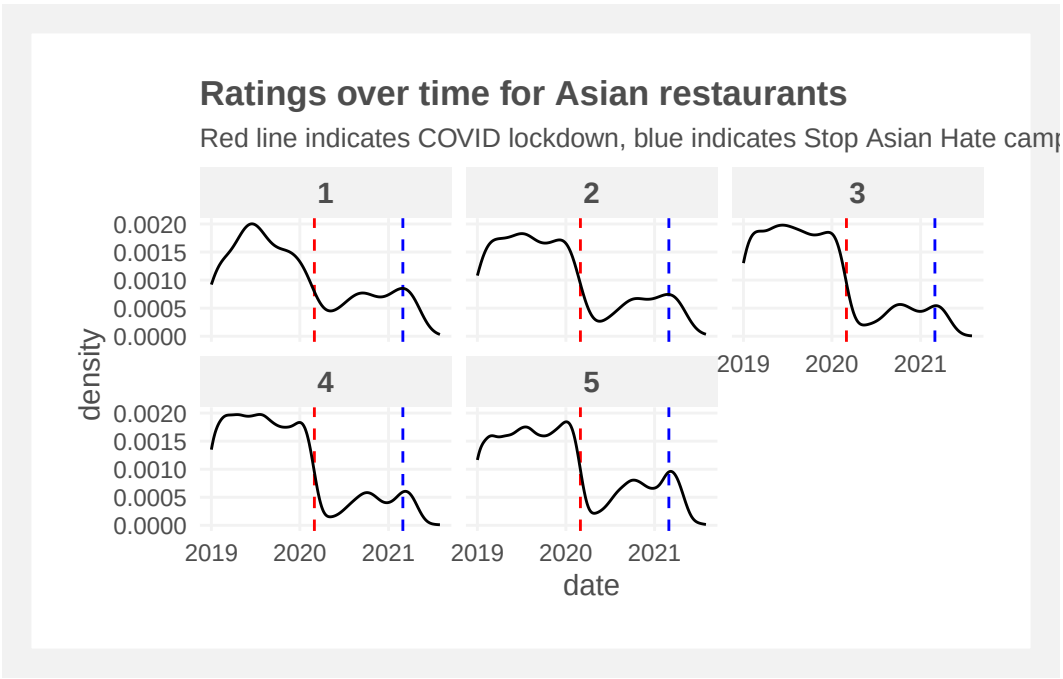



```
# messy plot - no real trend

combined_reviews |>
  group_by(gmap_id, month, year) |>
  ggplot(aes(x = date)) +
  geom_histogram(color = "white", fill = "darkslategray") +
  geom_vline(xintercept = covid_start, color = "red", lty = "dashed") +
  geom_vline(xintercept = stopah_start, color = "blue", lty = "dashed") +
  labs(title = "Total count of reviews for all restaurants over time")
```

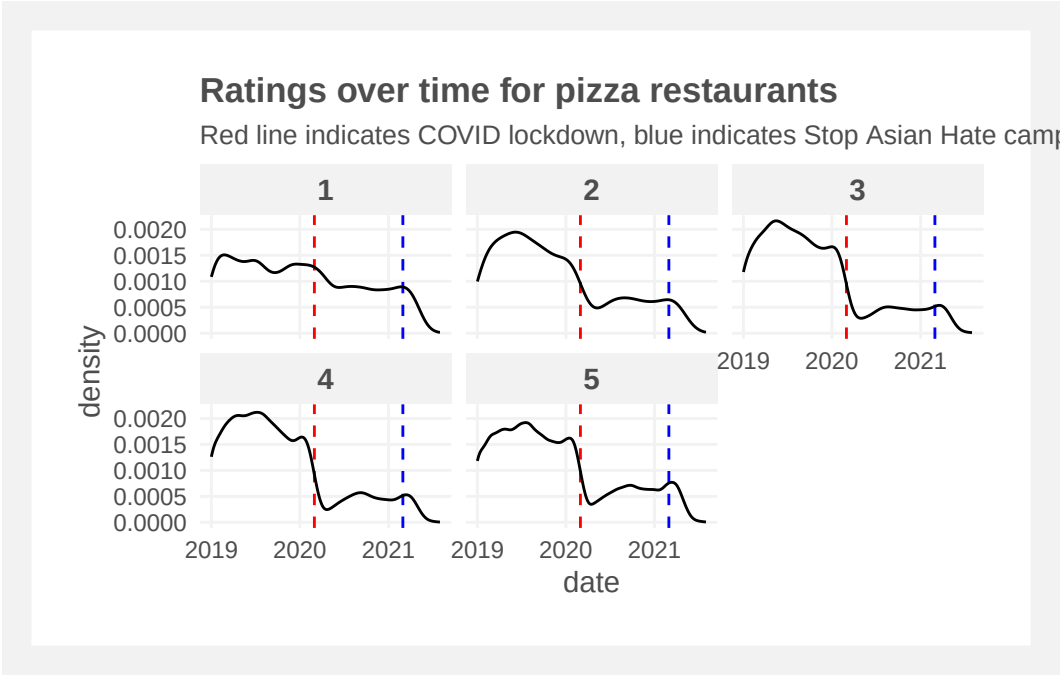
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



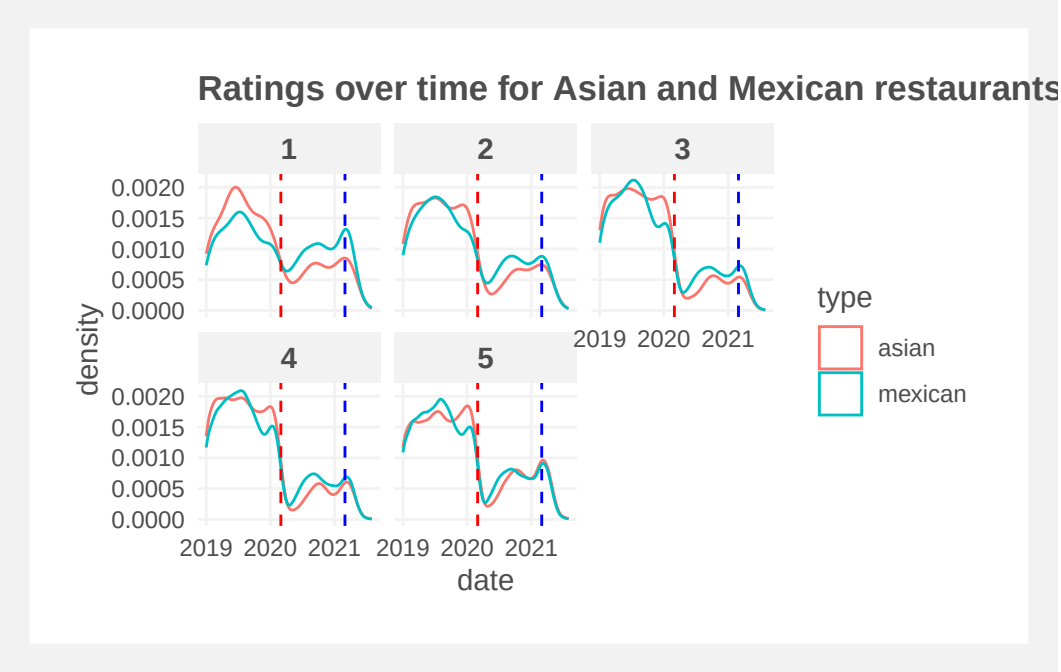

```
d_asian |>
  group_by(gmap_id, month, year, rating) |>
  mutate(date = as.Date(paste(year, month, 01), "%Y %m %d")) |>
  ggplot(aes(x = date)) +
  geom_vline(xintercept = covid_start, color = "red", lty = "dashed") +
  geom_vline(xintercept = stopah_start, color = "blue", lty = "dashed") +
  geom_density() +
  facet_wrap(vars(rating)) +
  labs(title = "Ratings over time for Asian restaurants", subtitle = "Red line indicates C
```




```
d_pizza |>
  group_by(gmap_id, month, year, rating) |>
  mutate(date = as.Date(paste(year, month, 01), "%Y %m %d")) |>
  ggplot(aes(x = date)) +
  geom_vline(xintercept = covid_start, color = "red", lty = "dashed") +
  geom_vline(xintercept = stopah_start, color = "blue", lty = "dashed") +
  geom_density() +
  facet_wrap(vars(rating)) +
  labs(title = "Ratings over time for pizza restaurants", subtitle = "Red line indicates C
```

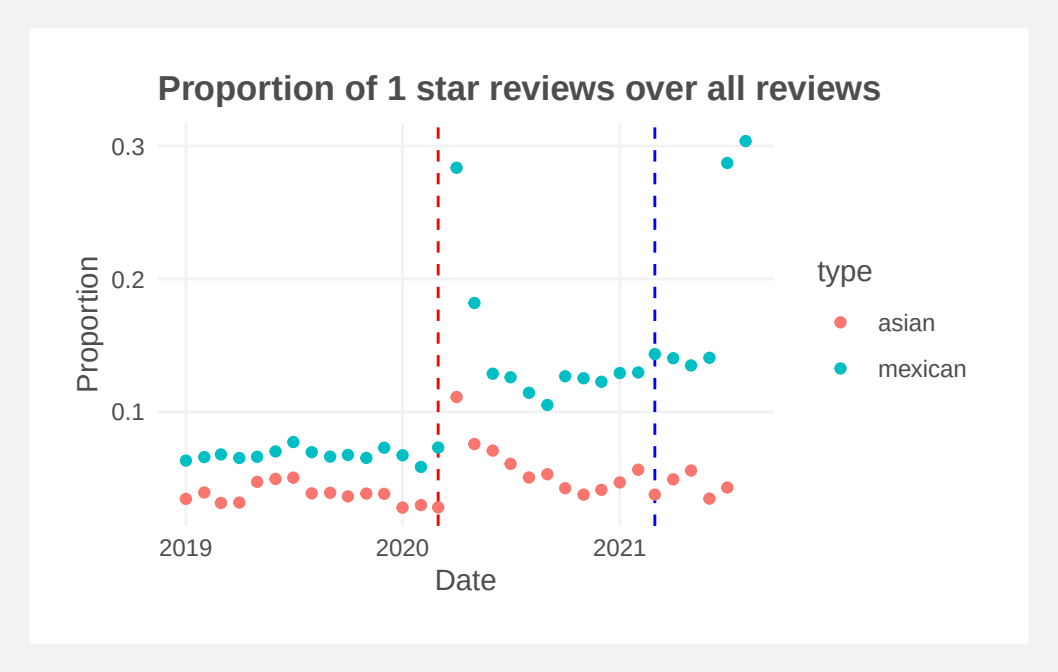



```
combined_reviews |>
  filter(type != "pizza") |>
  mutate(date = as.Date(paste(year, month, 01), "%Y %m %d")) |>
  ggplot(aes(x = date, color = type)) +
  geom_density() +
  geom_vline(xintercept = covid_start, color = "red", lty = "dashed") +
  geom_vline(xintercept = stopah_start, color = "blue", lty = "dashed") +
  facet_wrap(vars(rating)) +
  labs(title = "Ratings over time for Asian and Mexican restaurants")
```



did people get more unhappy with asian restaurants, and did they get less unhappy over the campaign

```
# trying to answer whether the drop for asian restaurants is more or the same for pizza/me
# proportion of 1 star reviews over all reviews
combined_reviews |>
  filter(type == "asian" | type == "mexican") |>
  count(rating, date, type) |>
  group_by(date, type) |>
  mutate(prop = prop.table(n)) |>
  arrange(date) |>
  filter(rating == 1) |>
  ggplot() +
  geom_vline(xintercept = covid_start, color = "red", lty = "dashed") +
  geom_vline(xintercept = stopah_start, color = "blue", lty = "dashed") +
  geom_point(aes(x = date, y = prop, color = type)) +
  labs(title = "Proportion of 1 star reviews over all reviews", x = "Date", y = "Proportio
```




```
combined_reviews |>
  filter(type == "asian", month == "04", year == "2020", rating < 3, !is.na(text)) |>
  pull(text) |>
  sample(10)
```

```
 [1] "They are the best and take very good care of their customers.\nAnd the food is to die
 [2] "Awful "curb side to go" experience. 2nd time in a month and unacceptable. Staff proper
 [3] "Awful "curbside" experience!  Schedule a pickup for 6:30pm and 6:45 still didn't recei
 [4] "Worst sushi I had in The city, and I eat a lot of sushi! super salty, and cheap tastin
 [5] "We just ordered from Grubhub and it never came. They say they are open but maybe they
 [6] "People giving this place great scores must not really know their ramen... The plates a
 [7] "Boneless crispy chicken was dry with no meat :/ service was great though"
 [8] "Don't order from here if you are doing take out. The food takes around 3 hours to come
 [9] "Been standing here for almost an hour now waiting on someone else's order not recommen
[10] "Mediocre overprice food in hells kitchen. There are so many options around, no need to
```

Sentiment analysis

```
review_text <- unnest_tokens(d_asian, word, text)

lexicon <- tidytext::get_sentiments("bing")
# remove the double rows for this word so that inner_join works
lexicon <- lexicon |> filter(!(word %in% c("envious", "enviously", "enviousness")))

review_stm <- review_text |>
  group_by(time, gmap_id) |>
  inner_join(lexicon, by = "word") |>
  mutate(score = ifelse(sentiment == "negative", 0, 1))

mean_stm <- review_stm |>
  group_by(time, gmap_id) |>
  summarize(mean_sentiment = mean(score, na.rm = T), mean_rating = mean(rating))

ggplot(mean_stm, aes(x = mean_rating, y = mean_sentiment)) +
  geom_point()
# sentiment aligns with rating? not really
# Doesn't show anything interesting
```

Plot restaurants by coordinates

```
restaurant_locations <- combined_reviews |>
  distinct(gmap_id, .keep_all = TRUE) |>
  select(-time, -rating, -text, -month, -year, -date) |>
  rowwise() |>
  mutate(
    # coordinates had the wrong quote marks for JSON parsing using from JSON
    # have to substitute
    coordinates = gsub('["{*}"]', "", coordinates),
    coordinates = gsub("'", '"', coordinates),
    coordinates = paste0("{", coordinates, "}"),
    lat = fromJSON(coordinates)$latitude,
    lon = fromJSON(coordinates)$longitude,
    zipcode = as.numeric(substr(address, (nchar(address) - 5 + 1), nchar(address)))
  ) |>
  filter(!is.na(zipcode))
```

```
Warning: There were 43 warnings in `mutate()`.
The first warning was:
i In argument: `zipcode = as.numeric(substr(address, (nchar(address) - 5 + 1),
  nchar(address)))`.
i In row 11.
Caused by warning:
! NAs introduced by coercion
i Run `dplyr::last_dplyr_warnings()` to see the 42 remaining warnings.
```

```r
# write_csv(restaurant_locations, file = "../review_data/restaurant_locations_by_type.csv"


register_google(key = read_lines("../google-api-key"))
nyc_map_12 <- qmap("new-york-city", color = "bw", zoom = 12)
```

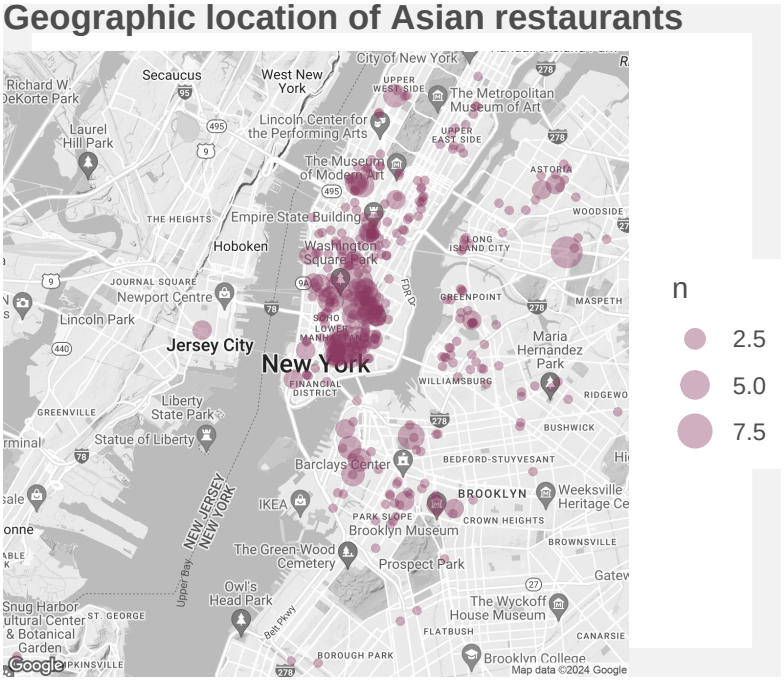
```
i <https://maps.googleapis.com/maps/api/staticmap?center=new-york-city&zoom=12&size=640x640&s

i <https://maps.googleapis.com/maps/api/geocode/json?address=new-york-city&key=xxx>
```

```r
# nyc_map <- qmap("new-york-city", color = "bw", zoom=11)
pd_asian <- restaurant_locations |>
  filter(type == "asian") |>
  group_by(lat, lon) |>
  summarize(n = n())
```

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

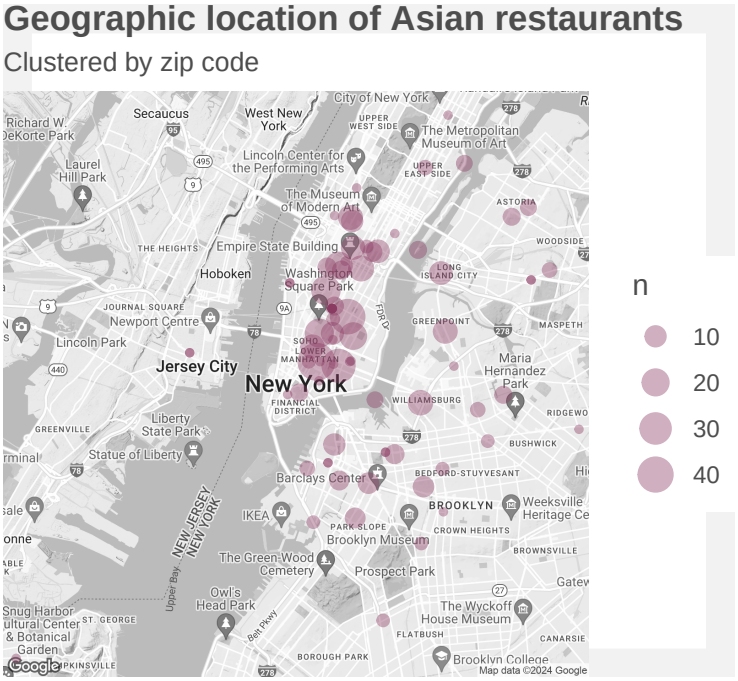
```r
pd_asian_zip <- restaurant_locations |>
  filter(type == "asian") |>
  group_by(zipcode) |>
  summarize(lat = mean(lat), lon = mean(lon), n = n())

nyc_map_12 + geom_point(
  data = pd_asian, aes(x = lon, y = lat, size = n),
  show.legend = T, col = alpha("hotpink4", 0.4)
) + labs(title = "Geographic location of Asian restaurants")
```

```
Warning: Removed 17 rows containing missing values (`geom_point()`).
```




```
nyc_map_12 + geom_point(
  data = pd_asian_zip, aes(x = lon, y = lat, size = n),
  show.legend = T, col = alpha("hotpink4", 0.4)
) + labs(title = "Geographic location of Asian restaurants", subtitle = "Clustered by zip
```

```
Warning: Removed 6 rows containing missing values (`geom_point()`).
```

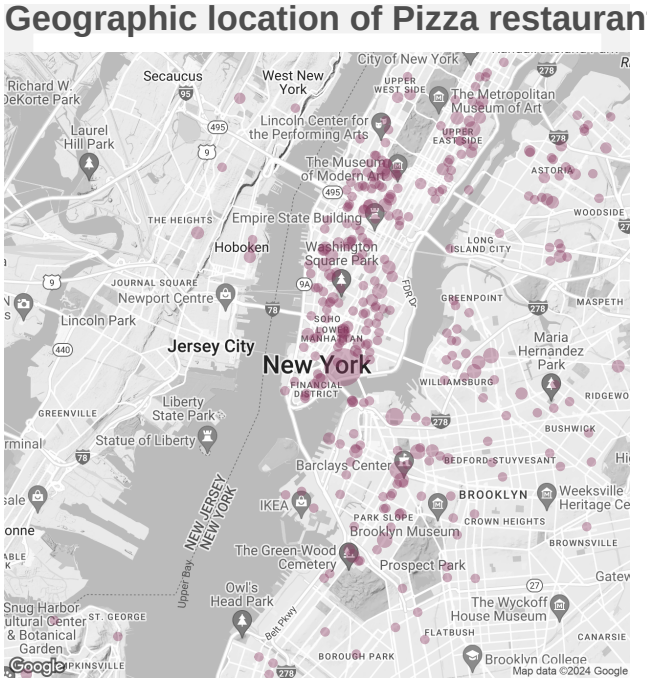



```
pd <- restaurant_locations |>
  filter(type == "pizza") |>
  group_by(lat, lon) |>
  summarize(n = n())
```

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

```
nyc_map_12 + geom_point(data = pd, aes(x = lon, y = lat, size = n), show.legend = FALSE, c
```

```
Warning: Removed 54 rows containing missing values (`geom_point()`).
```



see if there's a spike in racist reviews?????

BUT HOW!??!?!!

Work with polarity and subjectivity

```
# average polarity and subjectivity by restaurant
d <- reviews_with_sentiment |>
  mutate(date = as.Date(paste(year, month, 01), "%Y %m %d")) |>
  group_by(name, date) |>
  mutate(
    avg_polarity = mean(polarity, na.rm = TRUE),
    avg_subjectivity = mean(subjectivity, na.rm = TRUE)
  ) |>
  select(name, avg_polarity, avg_subjectivity) |>
  unique()
```

```
Adding missing grouping variables: `date`
```

```r
d <- d |>
  ungroup() |>
  mutate(
    scaled_avg_polarity = scale(avg_polarity)[, 1],
    scaled_avg_subjectivity = scale(avg_subjectivity)[, 1]
  )

d |>
  pivot_longer(starts_with("scaled"), names_to = "property", values_to = "value") |>
  select(name, property, value) |>
  ggplot() +
  geom_density(aes(value, color = property)) +
  labs(title = "Scaled average polarity and\nsubjectivity for Asian restaurants")
```