Review Data

Sophia Deng

2024-04-28

```r
knitr::opts_chunk$set(fig.width = 8, fig.height = 5)

library(geomtextpath) # for geom_textvline
```

```
## Warning: package 'geomtextpath' was built under R version 4.2.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.2.3
```

```r
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
library(mgcv)
```

```
## Warning: package 'mgcv' was built under R version 4.2.3

## Loading required package: nlme

## This is mgcv 1.9-1. For overview type 'help("mgcv-package")'.
```

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v stringr 1.5.0
## v forcats 1.0.0     v tibble  3.2.1
## v purrr     1.0.2     v tidyr   1.3.0
## v readr     2.1.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::collapse() masks nlme::collapse()
## x dplyr::filter()   masks stats::filter()
## x dplyr::lag()      masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
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.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

Split up combined reviews into 3 categories

```r
load("../review_data/combined_reviews.RData")

asian_reviews <- combined_reviews %>% filter(str_detect(type, "asian"))
asian_reviews <- na.omit(asian_reviews)
write.csv(asian_reviews, "../review_data/all_asian_reviews.csv")

pizza_reviews <- combined_reviews %>% filter(str_detect(type, "pizza"))
pizza_reviews <- na.omit(pizza_reviews)
write.csv(pizza_reviews, "../review_data/all_pizza_reviews.csv")

mexican_reviews <- combined_reviews %>% filter(str_detect(type, "mexican"))
mexican_reviews <- na.omit(mexican_reviews)
write.csv(mexican_reviews, "../review_data/all_mexican_reviews.csv")
```

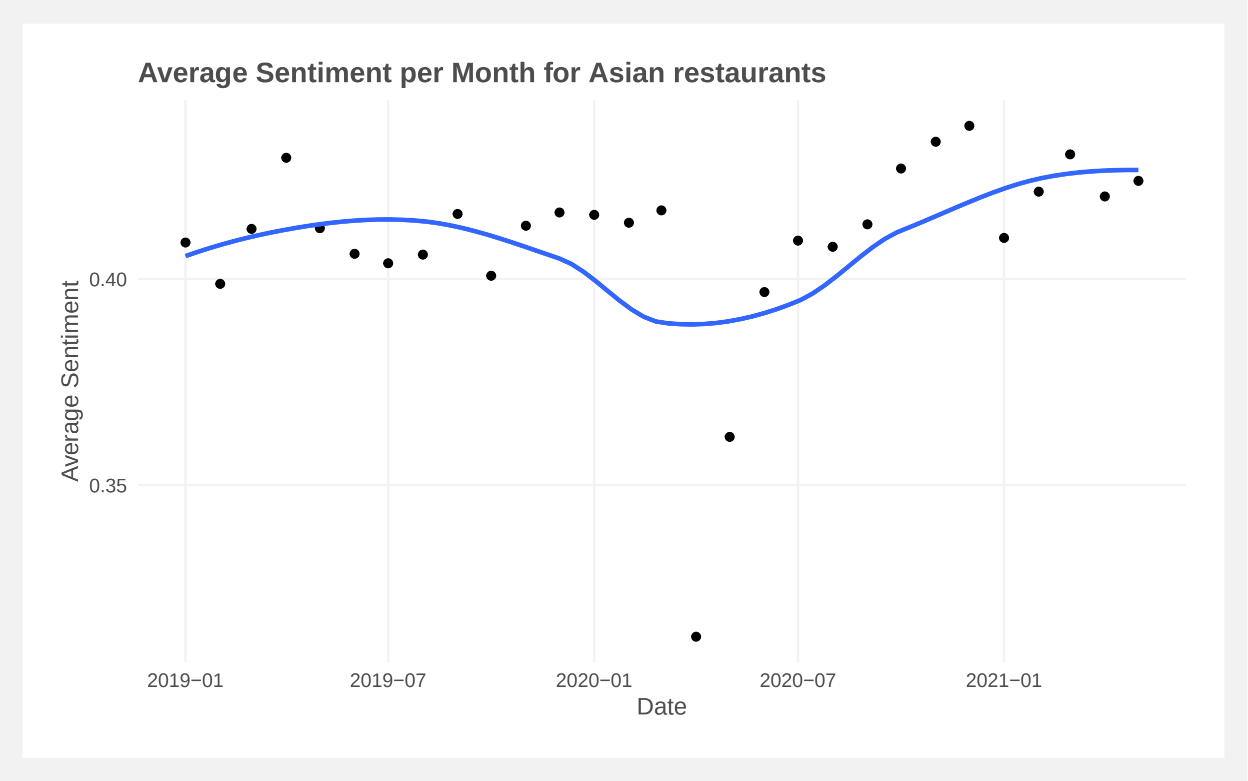
Plot average sentiment of Asian reviews

```r
asian_sentiment_reviews <- read_csv("../sentiment_data/asian_sentiment.csv") |>
  mutate(date = as.Date(paste(year, month, 01), "%Y %m %d")) |>
  filter(time <= as.Date(paste(2021, 06, 01), "%Y %m %d"))
```

```
## New names:
## Rows: 61575 Columns: 24
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (15): text, gmap_id, name, address, category, alias, categories, coordi... dbl
## (7): ...1, Unnamed: 0, rating, month, year, polarity, subjectivity dttm (1):
## time date (1): date
## 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`
```

```r
# Calculate average sentiment per month
asian_avg_sentiment <- aggregate(polarity ~ date, data = asian_sentiment_reviews, FUN = mean)

# Plot the graph
ggplot(asian_avg_sentiment, aes(x = date, y = as.numeric(polarity))) +
  geom_point() +
  geom_smooth(aes(group = 1), method = "loess", se = FALSE) +
  labs(x = "Date", y = "Average Sentiment", title = "Average Sentiment per Month for Asian restaurants"
```

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



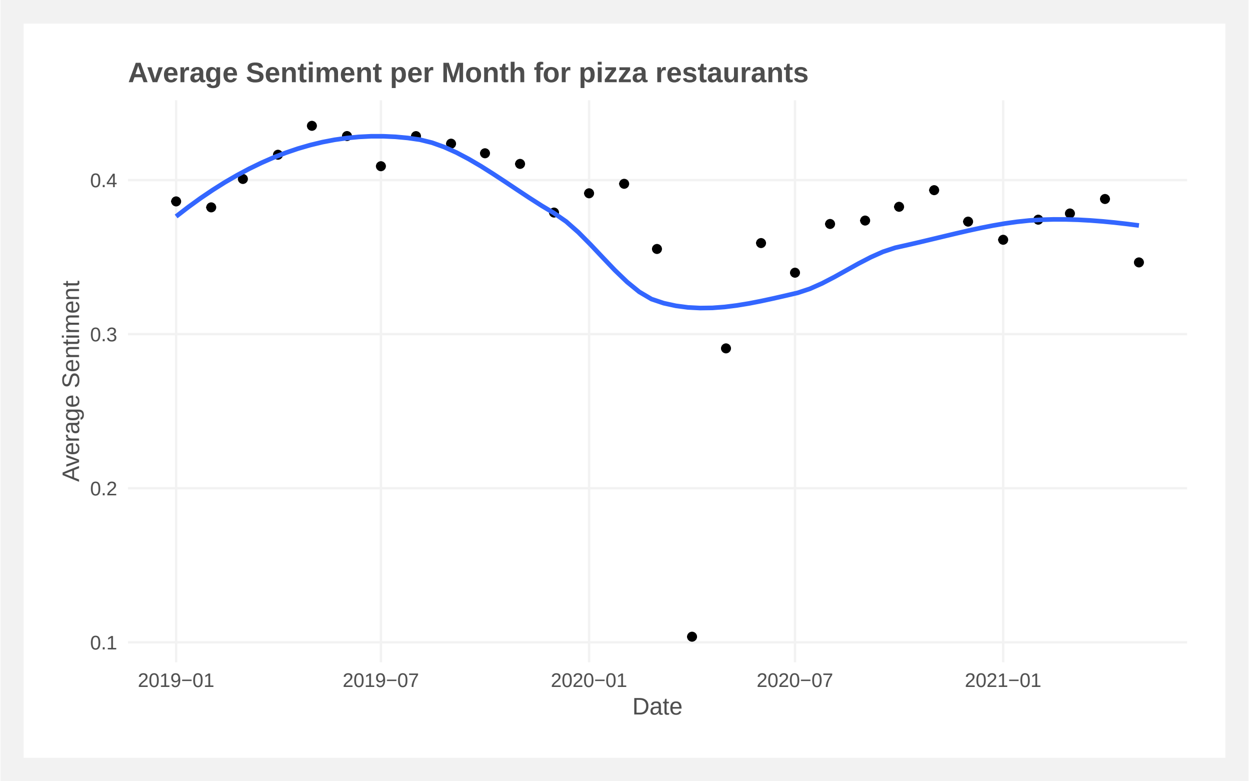
Plot average sentiment of pizza reviews

```
pizza_sentiment_reviews <- read_csv("../sentiment_data/pizza_sentiment.csv") |>
  mutate(date = as.Date(paste(year, month, 01), "%Y %m %d")) |>
  filter(time <= as.Date(paste(2021, 06, 01), "%Y %m %d"))
```

```
## New names:
## Rows: 98114 Columns: 24
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (15): text, gmap_id, name, address, category, alias, categories, coordi... dbl
## (7): ...1, Unnamed: 0, rating, month, year, polarity, subjectivity dttm (1):
## time date (1): date
## 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`
```

```
# Calculate average sentiment per month
pizza_avg_sentiment <- aggregate(polarity ~ date, data = pizza_sentiment_reviews, FUN = mean)

# Plot the graph
ggplot(pizza_avg_sentiment, aes(x = date, y = as.numeric(polarity))) +
  geom_point() +
  geom_smooth(aes(group = 1), method = "loess", se = FALSE) +
  labs(x = "Date", y = "Average Sentiment", title = "Average Sentiment per Month for pizza restaurants")
```

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



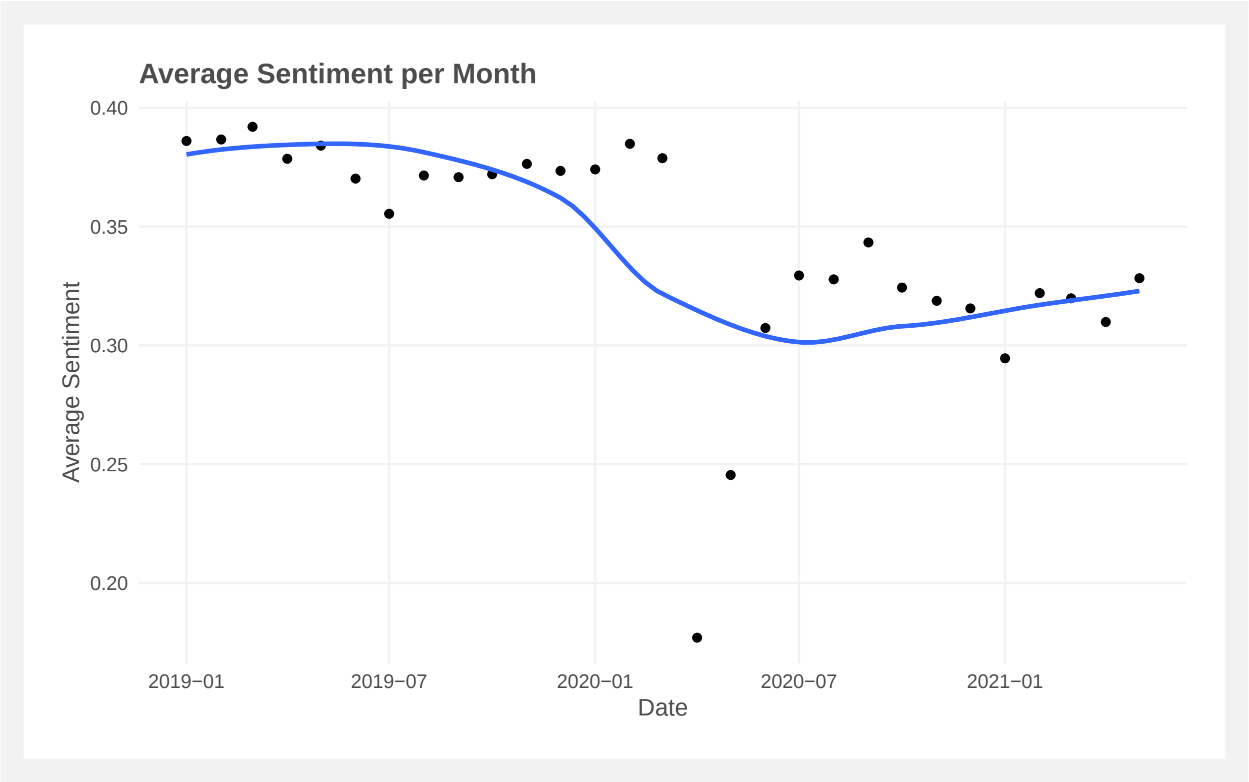
Plot average sentiment of Mexican reviews

```
mexican_sentiment_reviews <- read_csv("../sentiment_data/mexican_sentiment.csv") |>
  mutate(date = as.Date(paste(year, month, 01), "%Y %m %d")) |>
  filter(time <= as.Date(paste(2021, 06, 01), "%Y %m %d"))
```

```
## New names:
## Rows: 119039 Columns: 24
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (15): text, gmap_id, name, address, category, alias, categories, coordi... dbl
## (7): ...1, Unnamed: 0, rating, month, year, polarity, subjectivity dttm (1):
## time date (1): date
## 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`
```

```
# Calculate average sentiment per month
mexican_avg_sentiment <- aggregate(polarity ~ date, data = mexican_sentiment_reviews, FUN = mean)

# Plot the graph
ggplot(mexican_avg_sentiment, aes(x = date, y = as.numeric(polarity))) +
  geom_point() +
  geom_smooth(aes(group = 1), method = "loess", se = FALSE) +
  labs(x = "Date", y = "Average Sentiment", title = "Average Sentiment per Month")
```

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

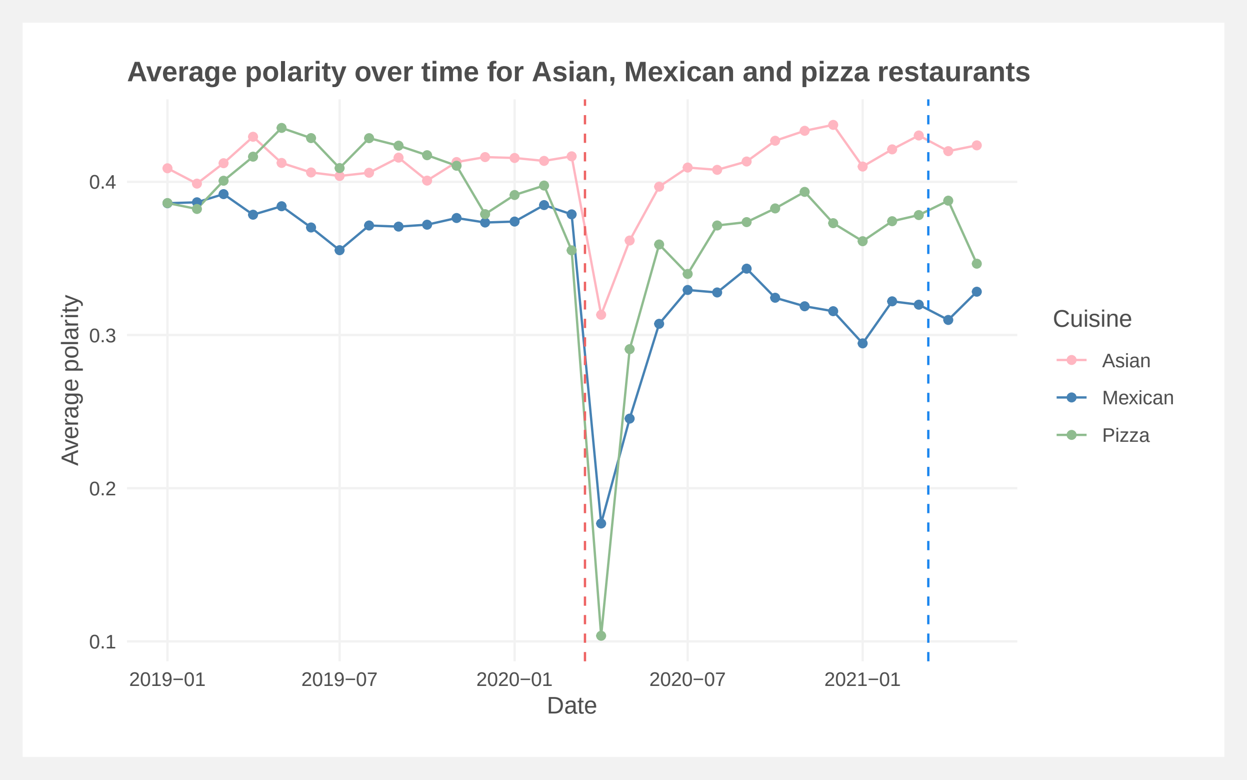


```r
# REMOVING LAST 3 MONTHS OF 2021 DATA

# create combined sentiment reviews
combined_reviews <-
  bind_rows(asian_sentiment_reviews, mexican_sentiment_reviews, pizza_sentiment_reviews) |>
  mutate(date = as.Date(paste(year, month, 01), "%Y %m %d")) |>
  filter(time <= as.Date(paste(2021, 06, 01), "%Y %m %d"))
```

Plot average polarity

```r
covid_start <- as.Date(paste(2020, 03, 15), "%Y %m %d")
stopah_start <- as.Date(paste(2021, 03, 11), "%Y %m %d")

combined_reviews |>
  mutate(type = factor(str_to_title(type), levels = c("Asian", "Mexican", "Pizza"))) |>
  group_by(type, month, year) |>
  summarise(mean_polarity = mean(polarity)) |>
  mutate(date = as.Date(paste(year, month, 01), "%Y %m %d")) |>
  filter(date <= as.Date(paste(2021, 06, 01), "%Y %m %d")) |>
  ggplot(aes(x = date, y = mean_polarity, color = type)) +
  geom_line() +
  geom_point() +
  scale_color_manual(values = c("lightpink", "steelblue", "darkseagreen")) +
  geom_vline(xintercept = covid_start, color = "indianred2", lty = "dashed") +
  geom_vline(xintercept = stopah_start, color = "dodgerblue2", lty = "dashed") +
  labs(title = "Average polarity over time for Asian, Mexican and pizza restaurants", color = "Cuisine"
```

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

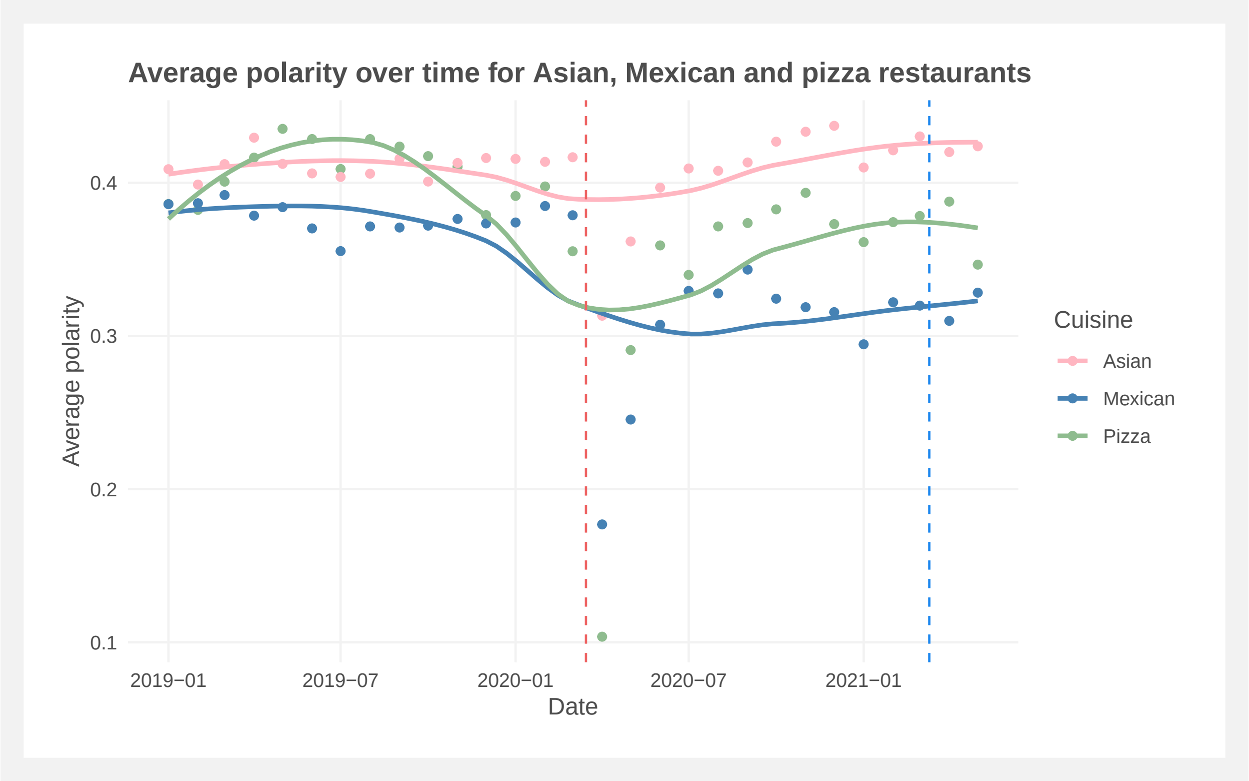


```r
# ggsave(filename = "../figures/average_polarity.png", width = 8, height = 5)
```

Plot average polarity with smoothed line

```r
avg_sentiment <- bind_rows(pizza_avg_sentiment, asian_avg_sentiment, mexican_avg_sentiment) |>
  mutate(type = rep(c("pizza", "asian", "mexican"), each = 29))

ggplot(avg_sentiment) +
  geom_point(aes(x = date, y = polarity, color = type)) +
  geom_smooth(aes(x = date, y = polarity, color = type), method = "loess", se = F) +
  scale_color_manual(values = c("lightpink", "steelblue", "darkseagreen"), labels = c("Asian", "Mexican
  geom_vline(xintercept = covid_start, color = "indianred2", lty = "dashed") +
  geom_vline(xintercept = stopah_start, color = "dodgerblue2", lty = "dashed") +
  labs(x = "Date", y = "Average polarity", title = "Average polarity over time for Asian, Mexican and p
```

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



```
# ggsave("../figures/average_monthly_sentiment.png", width = 8, height = 5)
```

GAM with Date 0 as 3/11/2020 (start of covid)

```
combined_reviews <- combined_reviews |> mutate(covid_date = as.numeric(lubridate::as_date(time) - lubri
asian_sentiment <- combined_reviews %>% filter(type == "asian")
mexican_sentiment <- combined_reviews %>% filter(type == "mexican")
pizza_sentiment <- combined_reviews %>% filter(type == "pizza")

# using gam model with te (tensor) which allows for an interaction
gam_asian <- mgcv::gam(polarity ~ 1 + te(rating, covid_date, k = c(4, 10)), data = asian_sentiment)
summary(gam_asian)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## polarity ~ 1 + te(rating, covid_date, k = c(4, 10))
##
## Parametric coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.413287   0.001117     370   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                       edf Ref.df     F p-value
## te(rating,covid_date) 22.4  26.91 748.1  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.247   Deviance explained = 24.7%
## GCV = 0.076595  Scale est. = 0.076566  n = 61360
```

```
gam_mexican <- mgcv::gam(polarity ~ 1 + te(rating, covid_date, k = c(4, 10)), data = mexican_sentiment)
summary(gam_mexican)
```

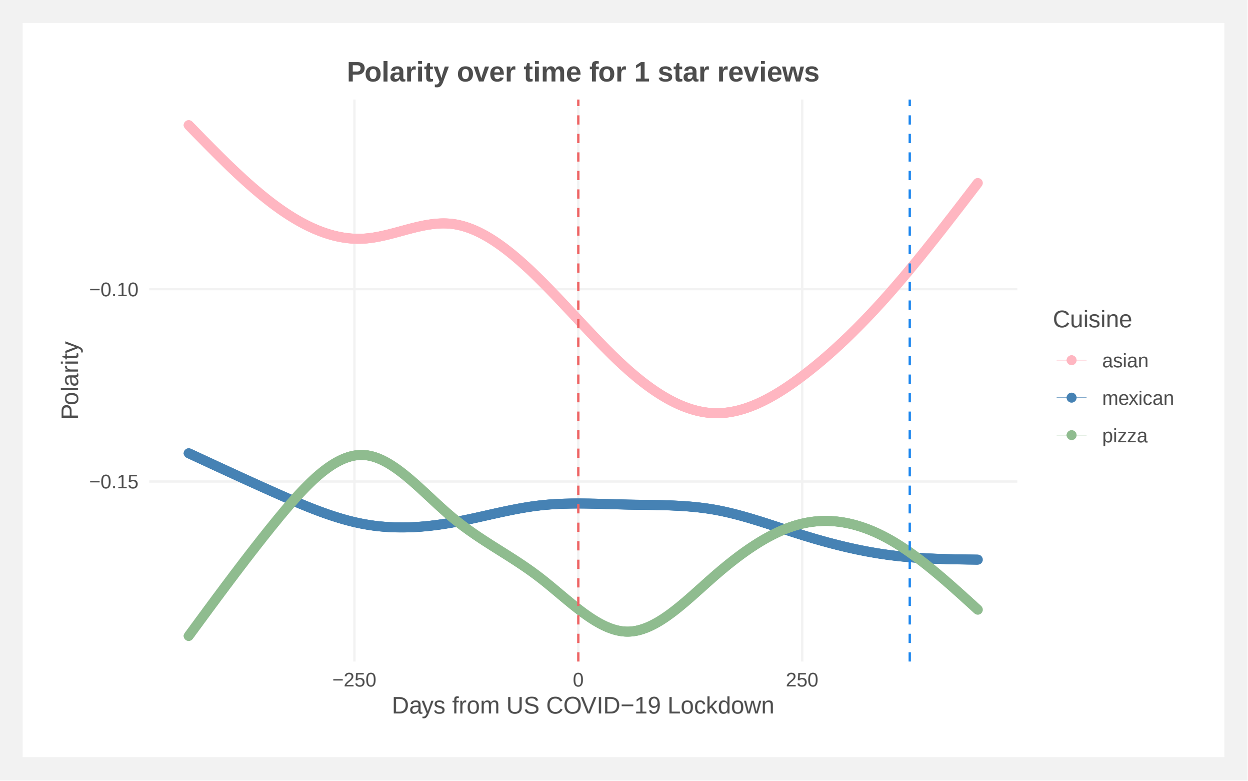
```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## polarity ~ 1 + te(rating, covid_date, k = c(4, 10))
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.3574188  0.0009014   396.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                         edf Ref.df    F p-value
## te(rating,covid_date) 20.43   24.4 2953  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.378   Deviance explained = 37.8%
## GCV = 0.096332  Scale est. = 0.096314  n = 118530
```

```
gam_pizza <- mgcv::gam(polarity ~ 1 + te(rating, covid_date, k = c(4, 10)), data = pizza_sentiment)
summary(gam_pizza)
```

```
##
## Family: gaussian
## Link function: identity
##
## Formula:
## polarity ~ 1 + te(rating, covid_date, k = c(4, 10))
##
## Parametric coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.3932125  0.0009831     400   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                         edf Ref.df    F p-value
## te(rating,covid_date) 26.67  31.58 1955  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.387   Deviance explained = 38.7%
## GCV = 0.094566  Scale est. = 0.09454   n = 97818
```

Comparing 1 star reviews

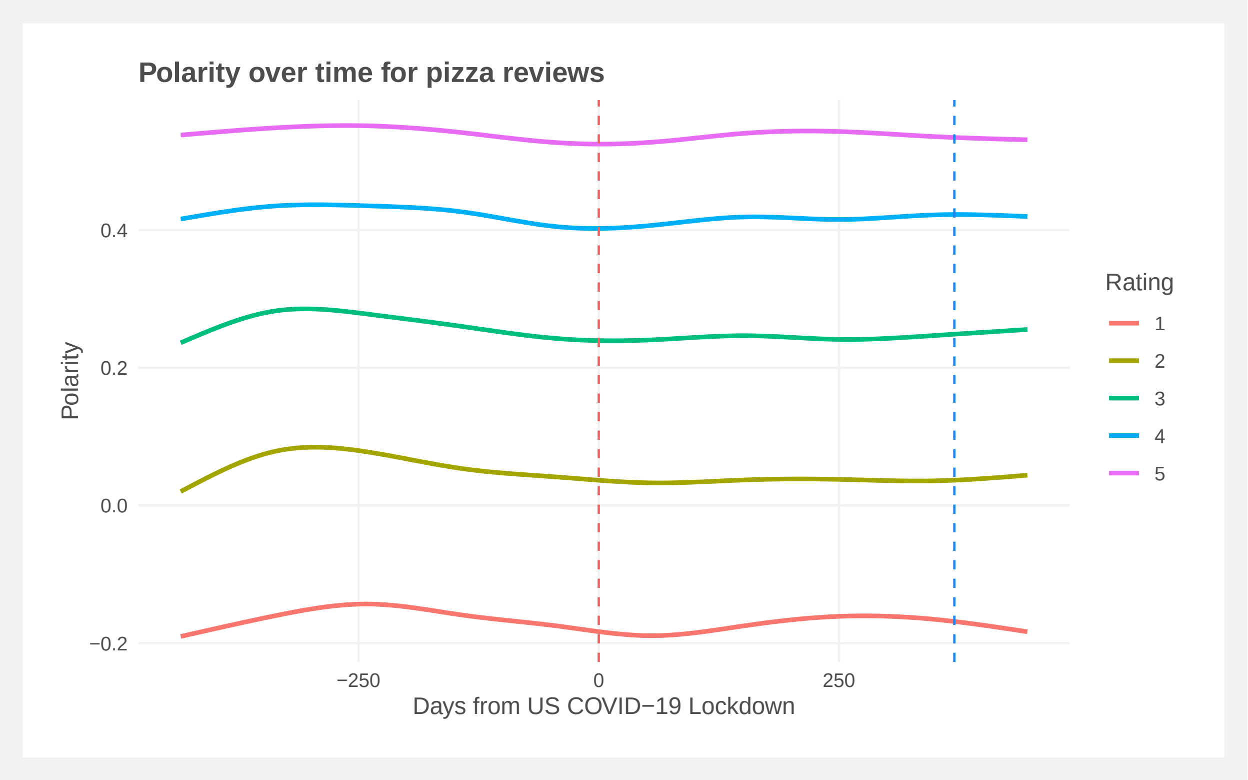
```
one_star_df <- data.frame(covid_date = -435:446, rating = 1)

asian_polarity <- predict(gam_asian, newdata = one_star_df)
mexican_polarity <- predict(gam_mexican, newdata = one_star_df)
pizza_polarity <- predict(gam_pizza, newdata = one_star_df)

one_star_preds <- bind_rows(
  bind_cols(one_star_df, polarity = asian_polarity, type = "asian"),
  bind_cols(one_star_df, polarity = mexican_polarity, type = "mexican"),
  bind_cols(one_star_df, polarity = pizza_polarity, type = "pizza")
)

ggplot(data = one_star_preds, aes(x = covid_date, y = polarity, color = type)) +
  geom_line(lwd = 0.1) +
  geom_point() +
  scale_color_manual(values = c("lightpink", "steelblue", "darkseagreen")) +
  geom_vline(xintercept = 0, color = "indianred2", lty = "dashed") +
  geom_vline(xintercept = 370, color = "dodgerblue2", lty = "dashed") +
  labs(title = "Polarity over time for 1 star reviews", color = "Cuisine", x = "Days from US COVID-19 L
  theme(plot.title = element_text(hjust = 0.5))
```



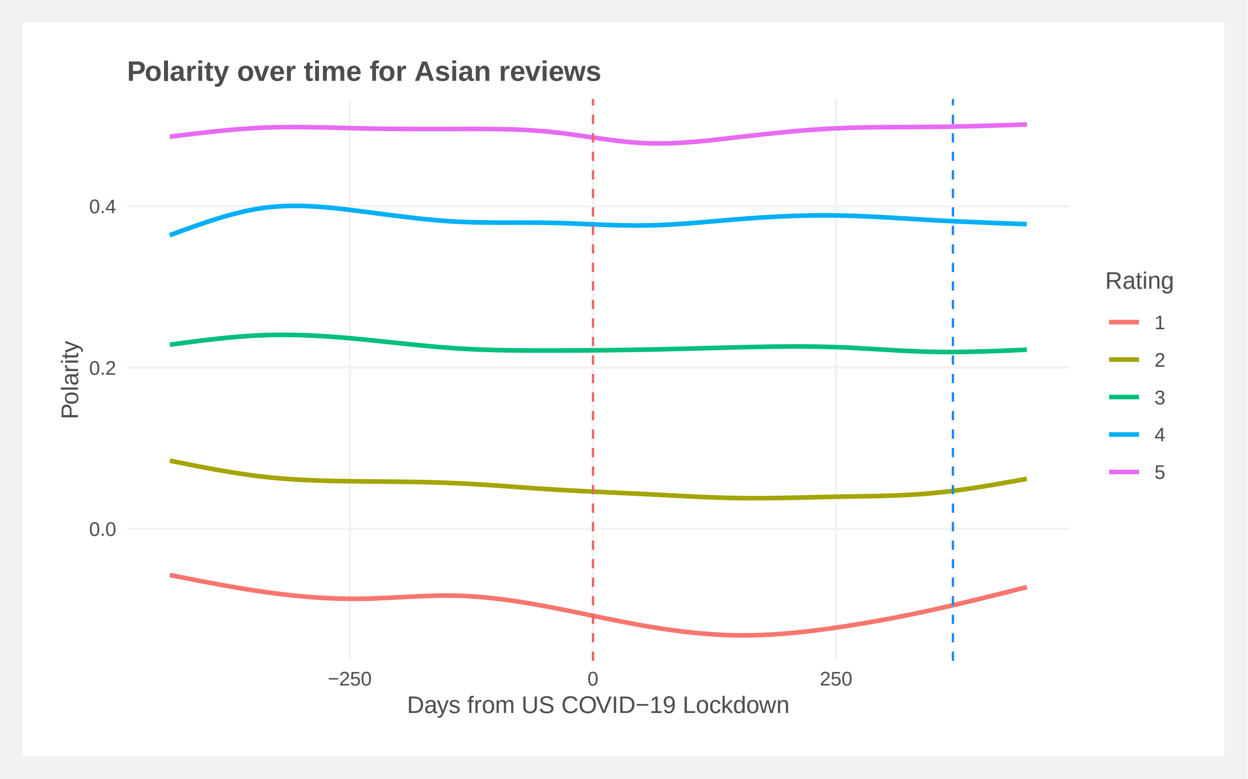
```r
# ggsave(filename = "../figures/polarity_for_1_star.png", width = 8, height = 5)
```

Comparing 5 star reviews

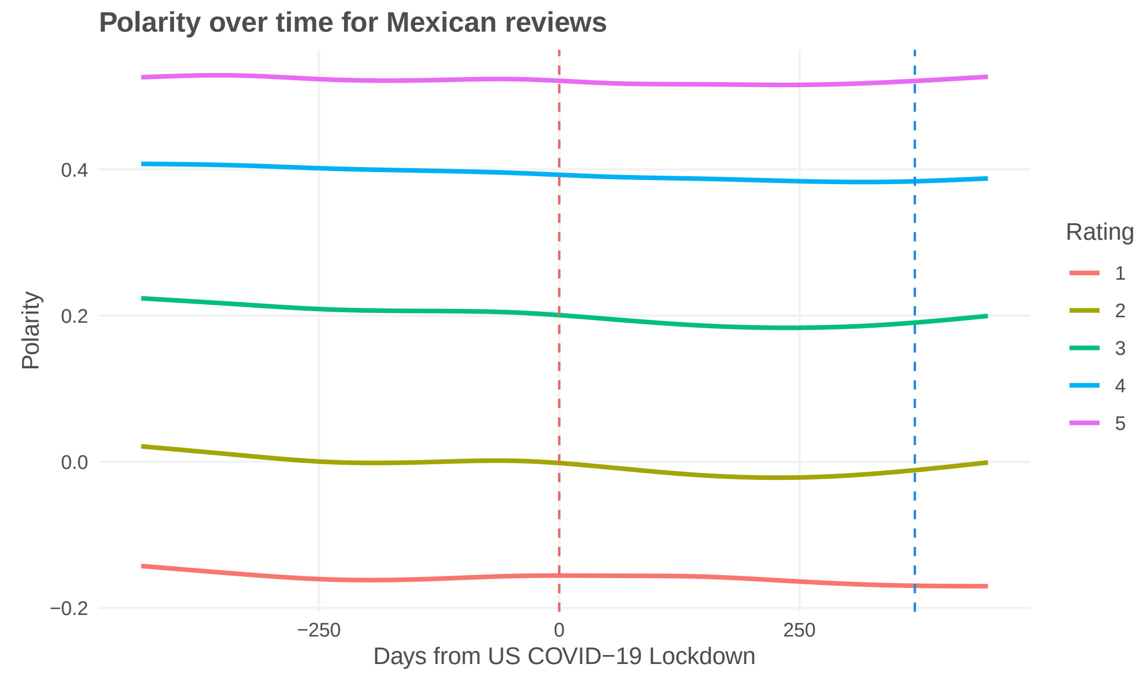
```r
all_stars_df <- data.frame(covid_date = rep(-435:446, each = 5), rating = 1:5)

asian_polarity <- predict(gam_asian, newdata = all_stars_df)
mexican_polarity <- predict(gam_mexican, newdata = all_stars_df)
pizza_polarity <- predict(gam_pizza, newdata = all_stars_df)

all_stars_preds <- bind_rows(
  bind_cols(all_stars_df, polarity = asian_polarity, type = "asian"),
  bind_cols(all_stars_df, polarity = mexican_polarity, type = "mexican"),
  bind_cols(all_stars_df, polarity = pizza_polarity, type = "pizza")
)

all_stars_preds |>
  filter(type == "pizza") |>
  ggplot(aes(x = covid_date, y = polarity, color = as.character(rating))) +
  geom_line(lwd = 1) +
  geom_vline(xintercept = 0, color = "indianred2", lty = "dashed") +
  geom_vline(xintercept = 370, color = "dodgerblue2", lty = "dashed") +
  labs(title = "Polarity over time for pizza reviews", color = "Rating", x = "Days from US COVID-19 Loc
```



```r
all_stars_preds |>
  filter(type == "asian") |>
  ggplot(aes(x = covid_date, y = polarity, color = as.character(rating))) +
  geom_line(lwd = 1) +
  geom_vline(xintercept = 0, color = "indianred2", lty = "dashed") +
  geom_vline(xintercept = 370, color = "dodgerblue2", lty = "dashed") +
  labs(title = "Polarity over time for Asian reviews", color = "Rating", x = "Days from US COVID-19 Lock
```

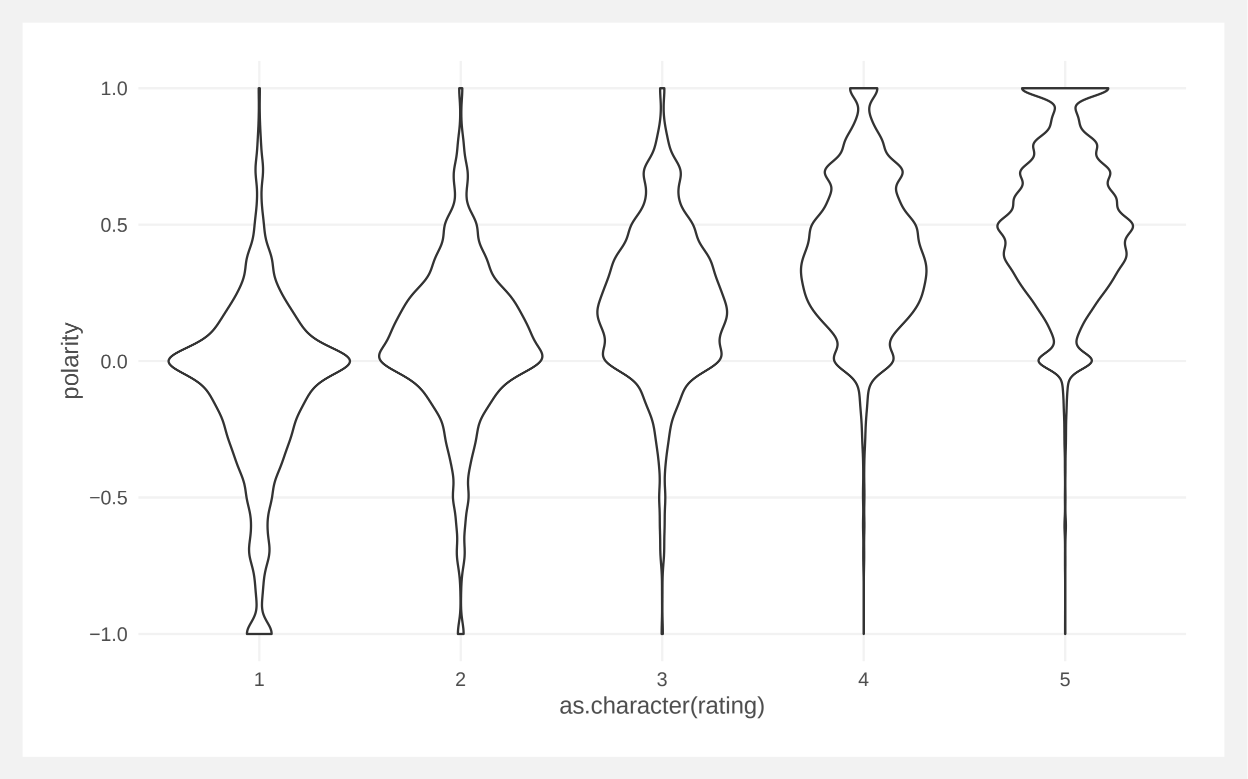


```
# ggsave("../figures/polarity_asian_1_to_5_stars.png", width = 8, height = 5)

all_stars_preds |>
  filter(type == "mexican") |>
  ggplot(aes(x = covid_date, y = polarity, color = as.character(rating))) +
  geom_line(lwd = 1) +
  geom_vline(xintercept = 0, color = "indianred2", lty = "dashed") +
  geom_vline(xintercept = 370, color = "dodgerblue2", lty = "dashed") +
  labs(title = "Polarity over time for Mexican reviews", color = "Rating", x = "Days from US COVID-19 L
```

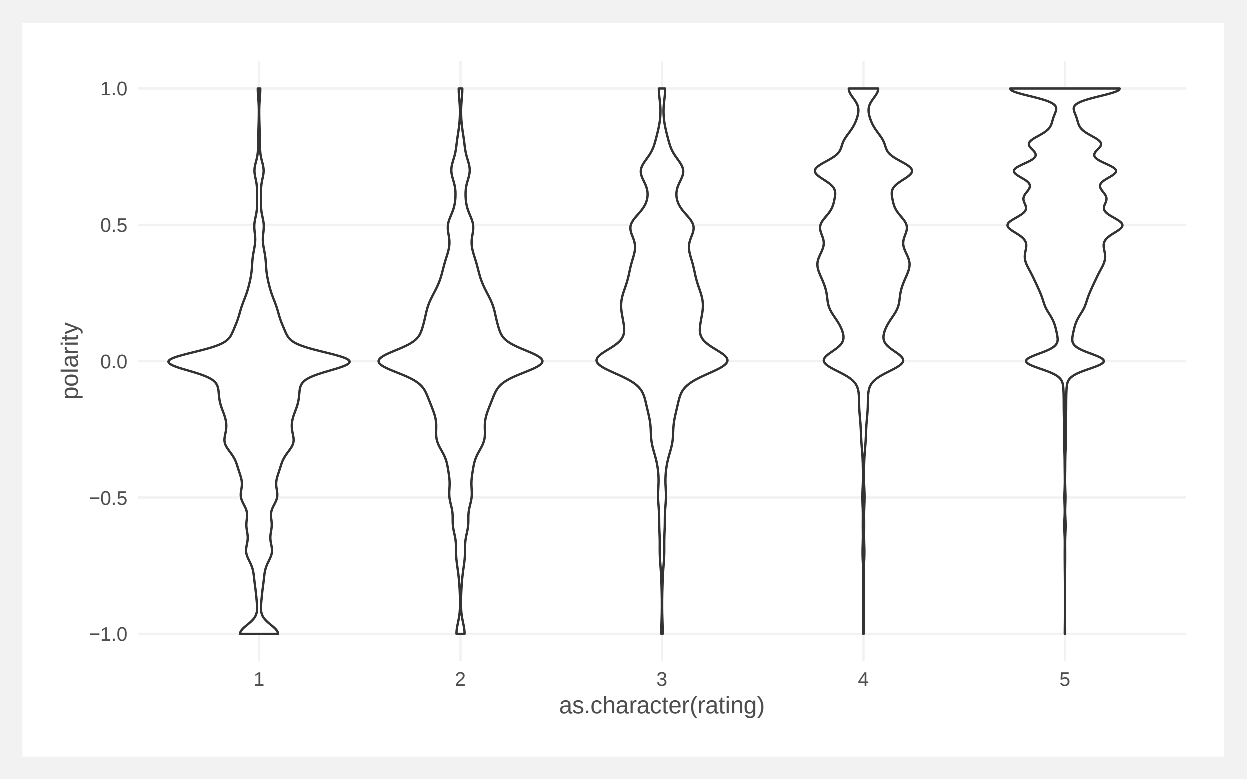


Violin plots

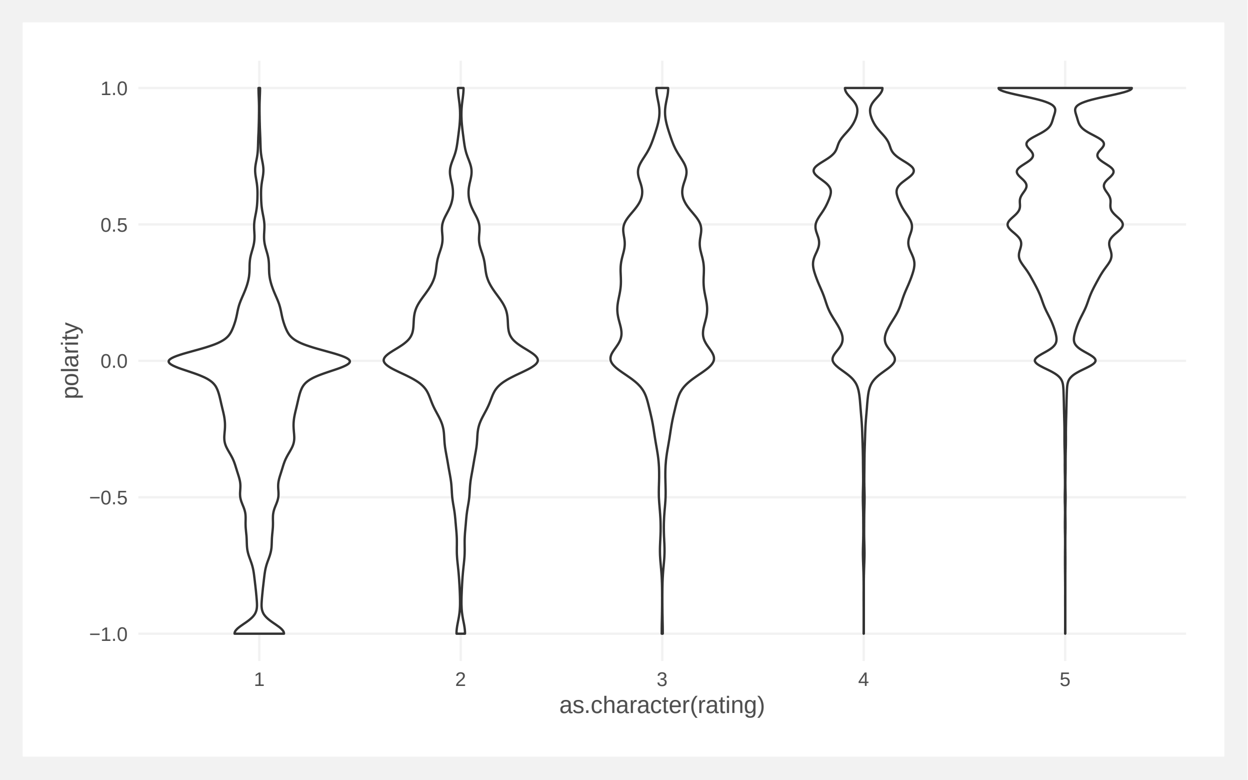
```
# polarity vs rating
ggplot(data = asian_sentiment, aes(x = as.character(rating), y = polarity)) +
  geom_violin()
```



```r
ggplot(data = mexican_sentiment, aes(x = as.character(rating), y = polarity)) +
  geom_violin()
```



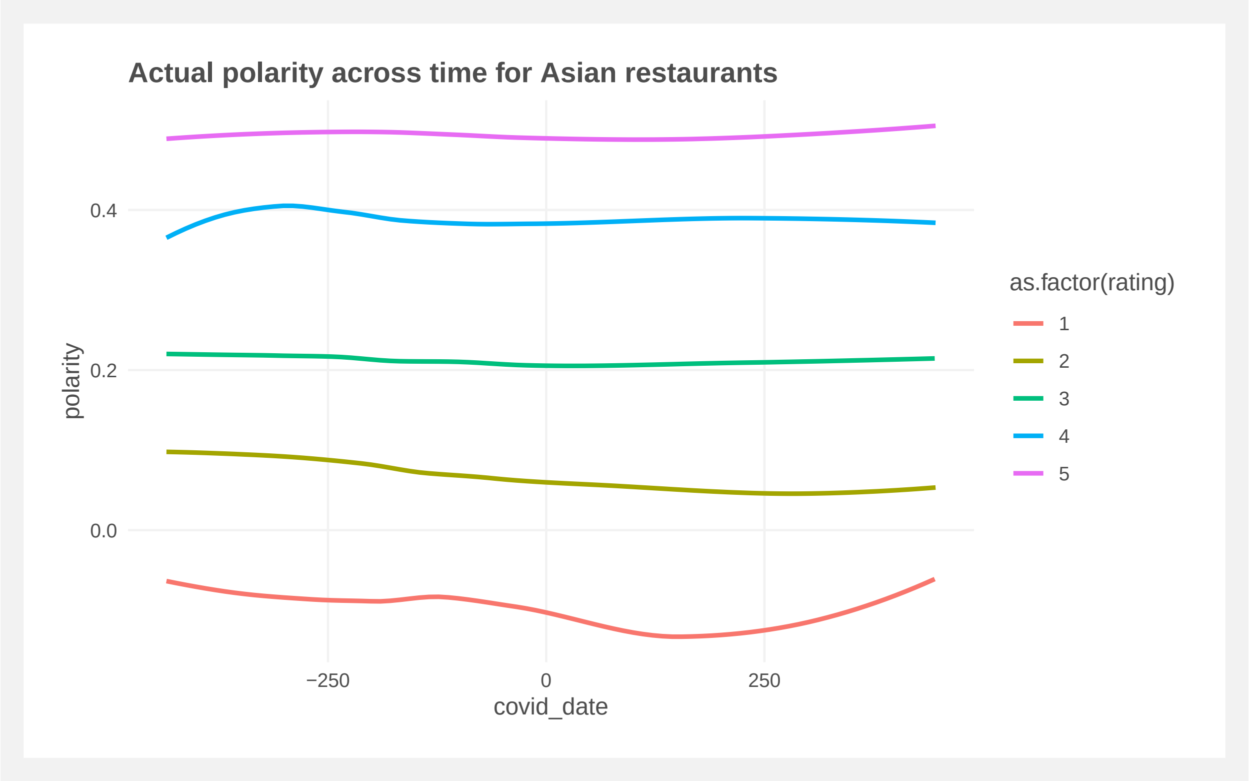
```
ggplot(data = pizza_sentiment, aes(x = as.character(rating), y = polarity)) +
  geom_violin()
```



Actual polarity vs date, coloured by rating

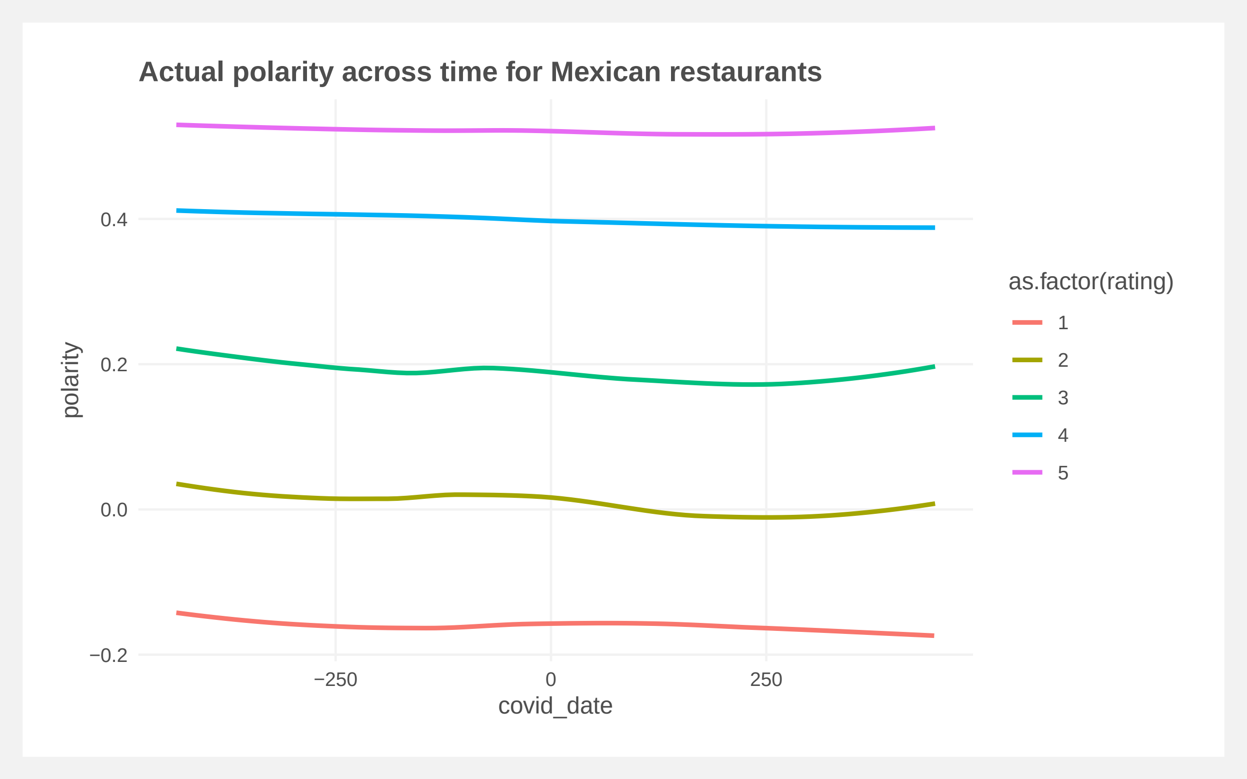
```
# polarity vs date, color by rating

ggplot(data = asian_sentiment, aes(x = covid_date, y = polarity)) +
  geom_smooth(aes(color = as.factor(rating)), method = "loess", se = FALSE) +
  labs(title = "Actual polarity across time for Asian restaurants")
```

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



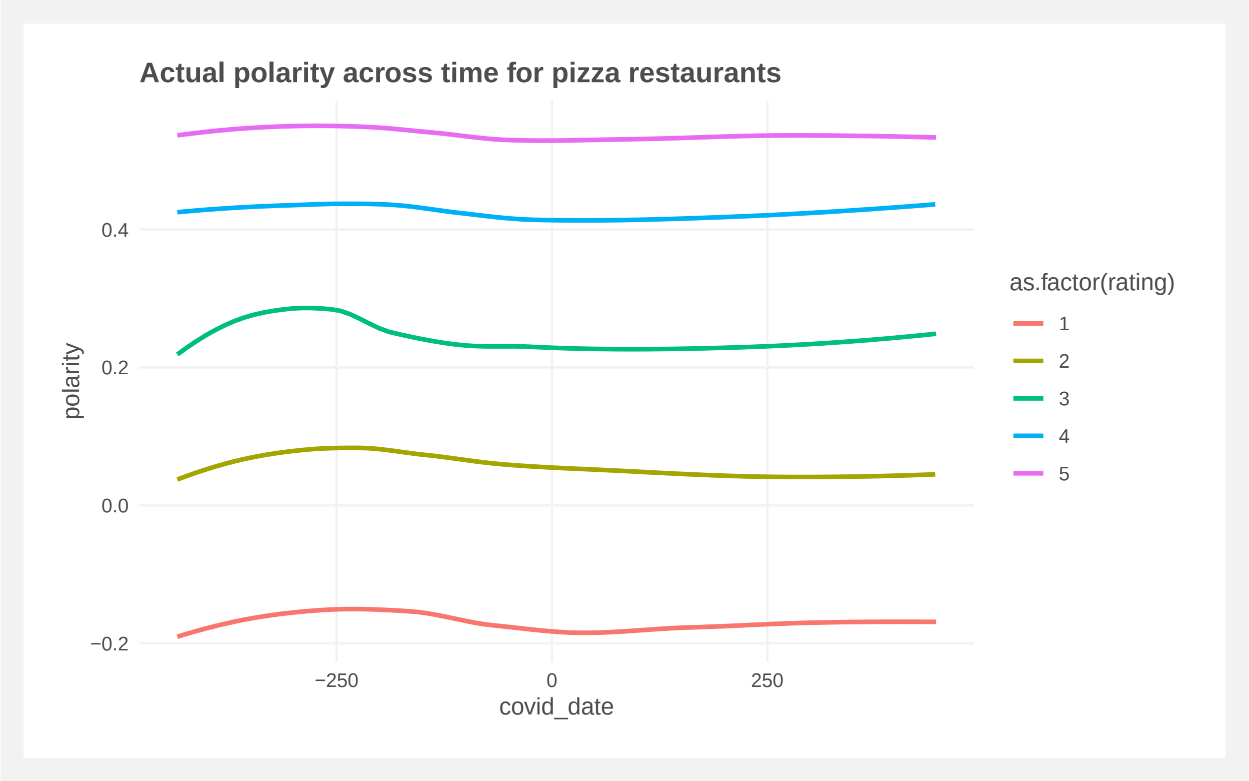
```
ggplot(data = mexican_sentiment, aes(x = covid_date, y = polarity)) +
  geom_smooth(aes(color = as.factor(rating)), method = "loess", se = FALSE) +
  labs(title = "Actual polarity across time for Mexican restaurants")
```

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



```
ggplot(data = pizza_sentiment, aes(x = covid_date, y = polarity)) +
  geom_smooth(aes(color = as.factor(rating)), method = "loess", se = FALSE) +
  labs(title = "Actual polarity across time for pizza restaurants")
```

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



Projected range along one variable (date), but fixed ratings (between 1 to 5) for all observations. We plotted the response of the model for fixed values of the date.

For a random forest: we have a median value. Can do this for up to two variables: interaction patterns, plotting by heatmap: color boxes by model output (vary two variables on x and y axis, are there hotspots for large or small model outputs)