DatingApp Review

Xiaoqian Xiang

2025-07-23

Contents

Step 1: Read and view the basic information

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## 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 errors
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```r
# read into the data
tinder <- read_csv("tinder_google_play_reviews.csv")
```

```
## Rows: 654532 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): reviewId, userName, userImage, content, reviewCreatedVersion, repl...
## dbl  (2): score, thumbsUpCount
## dttm (2): at, repliedAt
##
## 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.
```

```r
bumble <- read_csv("bumble_google_play_reviews.csv")
```

```
## Rows: 169358 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): reviewId, userName, userImage, content, reviewCreatedVersion, repl...
## dbl  (2): score, thumbsUpCount
## dttm (2): at, repliedAt
##
## 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.
```

```r
hinge <- read_csv("hinge_google_play_reviews.csv")
```

```
## Rows: 79671 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): reviewId, userName, userImage, content, reviewCreatedVersion, repl...
## dbl  (2): score, thumbsUpCount
## dttm (2): at, repliedAt
##
## 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.
```

```r
okcupid <- read_csv("okcupid_google_play_reviews.csv")
```

```
## Rows: 143467 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): reviewId, userName, userImage, content, reviewCreatedVersion, repl...
## dbl  (2): score, thumbsUpCount
## dttm (2): at, repliedAt
##
## 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.
```

```
reviews_all <- read_csv("DatingAppReviewsDataset.csv")
```

```
## New names:
## Rows: 681994 Columns: 7
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (4): Name, Review, Date&Time, App dbl (3): ...1, Rating, #ThumbsUp
## 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`
```

Step 2: Clean, organize, and merge the data

```
tinder_clean <- tinder %>%
  transmute(
    review = content,
    rating = score,
    thumbs_up = thumbsUpCount,
    date = at,
    reply = replyContent,
    app_name = "Tinder"
  )
bumble_clean <- bumble %>%
  transmute(
    review = content,
    rating = score,
    thumbs_up = thumbsUpCount,
    date = at,
    reply = replyContent,
    app_name = "Bumble"
  )

hinge_clean <- hinge %>%
  transmute(
    review = content,
    rating = score,
    thumbs_up = thumbsUpCount,
    date = at,
    reply = replyContent,
    app_name = "Hinge"
  )

okcupid_clean <- okcupid %>%
  transmute(
    review = content,
    rating = score,
    thumbs_up = thumbsUpCount,
    date = at,
    reply = replyContent,
    app_name = "OkCupid"
  )
```

```r
library(tidyverse)
library(lubridate)

reviews_all_clean <- reviews_all %>%
  transmute(
    review = Review,
    rating = Rating,
    thumbs_up = `#ThumbsUp`,
    date = dmy_hm(`Date&Time`),
    reply = NA_character_,
    app_name = App
  )

all_reviews <- bind_rows( tinder_clean, bumble_clean, hinge_clean,
                          okcupid_clean, reviews_all_clean )
head(all_reviews, 10)
```

```
## # A tibble: 10 x 6
##    review                    rating thumbs_up date                reply app_name
##    <chr>                      <dbl>     <dbl> <dttm>              <chr> <chr>
##  1 "messages won't load sup~      2         0 2025-07-18 00:16:45 <NA>  Tinder
##  2 "Terrible! Searching nea~      1         0 2025-07-18 00:11:19 <NA>  Tinder
##  3 "Good"                         4         0 2025-07-18 00:04:00 <NA>  Tinder
##  4 "fake profiles"                1         0 2025-07-17 23:45:47 <NA>  Tinder
##  5 "year of the Monkey, may~      5         0 2025-07-17 23:40:52 <NA>  Tinder
##  6 "I've matched with so ma~      1         0 2025-07-17 23:22:49 <NA>  Tinder
##  7 "the update and taking m~      5         0 2025-07-17 22:37:11 <NA>  Tinder
##  8 "Like all other apps, th~      2         0 2025-07-17 22:25:36 <NA>  Tinder
##  9 "not bad , is really int~      2         0 2025-07-17 21:37:39 <NA>  Tinder
## 10 "very good app"                5         0 2025-07-17 21:17:58 <NA>  Tinder
```

```r
str(all_reviews)
```

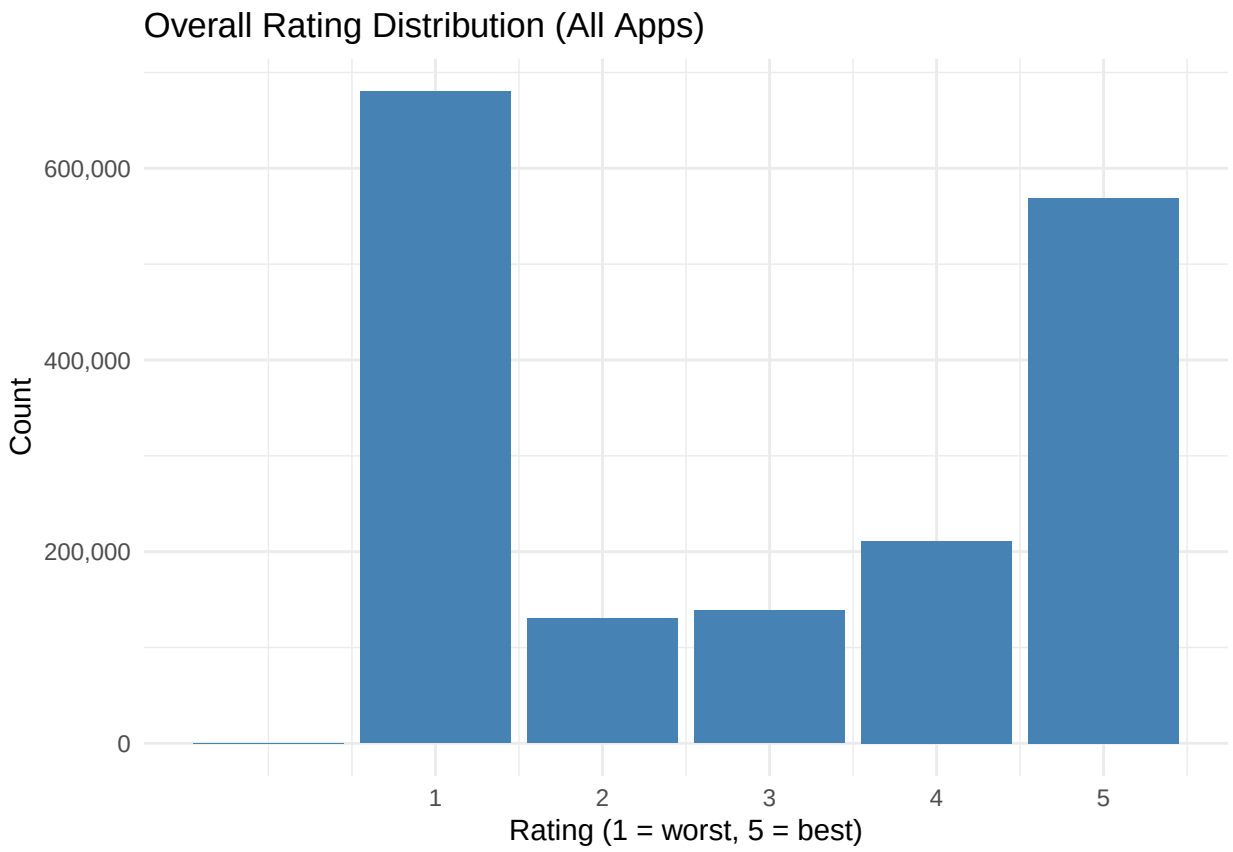
```
## tibble [1,729,022 x 6] (S3: tbl_df/tbl/data.frame)
##  $ review   : chr [1:1729022] "messages won't load support doesn't assist with bugs or respond so mar
##  $ rating   : num [1:1729022] 2 1 4 1 5 1 5 2 2 5 ...
##  $ thumbs_up: num [1:1729022] 0 0 0 0 0 0 0 0 0 0 ...
##  $ date     : POSIXct[1:1729022], format: "2025-07-18 00:16:45" "2025-07-18 00:11:19" ...
##  $ reply    : chr [1:1729022] NA NA NA NA ...
##  $ app_name : chr [1:1729022] "Tinder" "Tinder" "Tinder" "Tinder" ...
```

```r
write.csv(all_reviews, file = "DatingAppReview-cleanData.csv")
```

Step 3: Rating distribution of apps

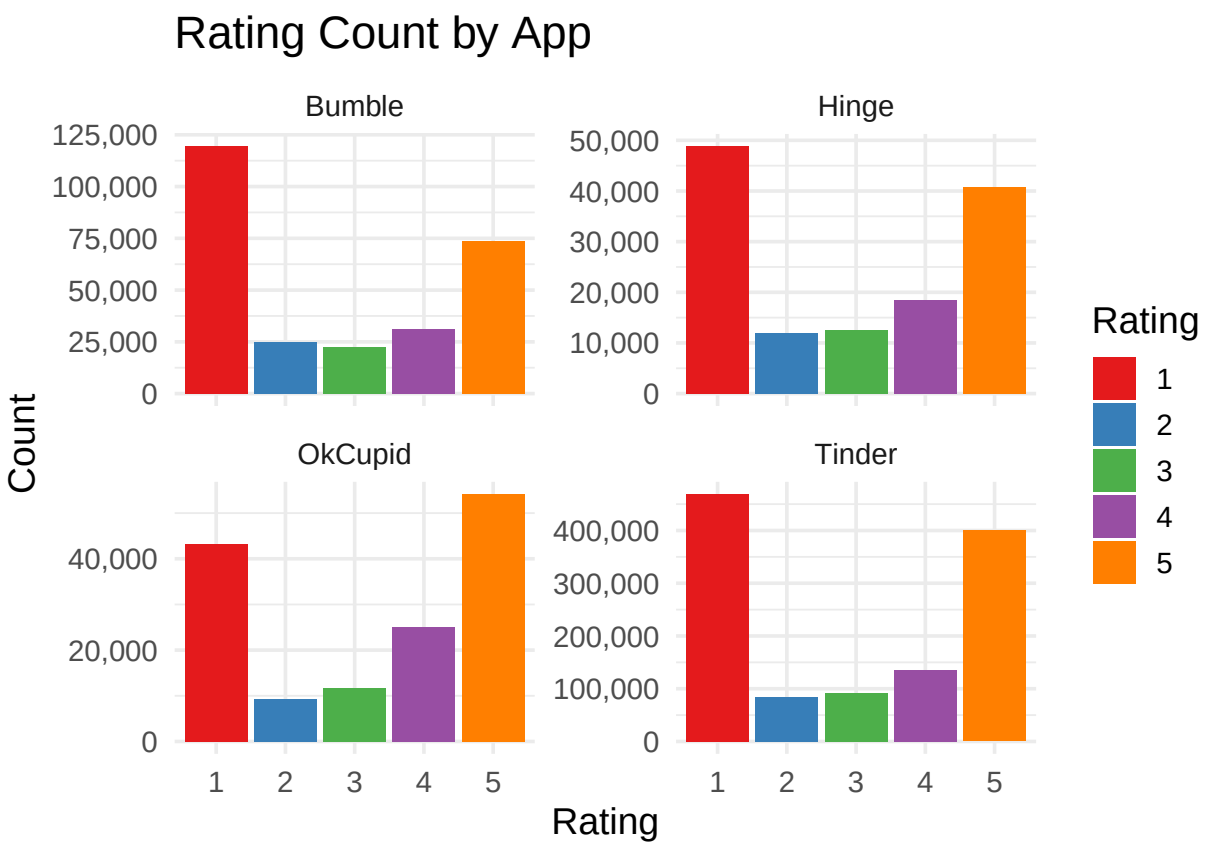
```r
# Overall Rating Score Distribution (All Apps)
ggplot(all_reviews, aes(x = rating)) +
  geom_bar(fill = "steelblue") +
  scale_x_continuous(breaks = 1:5) +
```

```
  scale_y_continuous(labels = label_comma()) +
  labs(
    title = "Overall Rating Distribution (All Apps)",
    x = "Rating (1 = worst, 5 = best)",
    y = "Count"
  ) +
  theme_minimal()
```




```
# Rating Score Count by App
ggplot(all_reviews, aes(x = factor(rating), fill = factor(rating))) +
  geom_bar(show.legend = TRUE) +
  facet_wrap(~ app_name, scales = "free_y") +
  scale_x_discrete(limits = as.character(1:5)) +
  scale_y_continuous(labels = label_comma()) +
  scale_fill_brewer(palette = "Set1", name = "Rating") +
  labs(
    title = "Rating Count by App",
    x = "Rating",
    y = "Count"
  ) +
  theme_minimal(base_size = 14)
```

```
## Warning: Removed 9 rows containing non-finite outside the scale range
## (`stat_count()`).
```



Interpretation:

Users' rating is clearly polarized: The number of 1-star and 5-star reviews far exceeds the median. It indicates that users' experience of using dating apps is extremely differentiated, with love and hate being polarized.

Step 4: Word Cloud of Reviews by apps

```
library(tidytext)
library(dplyr)
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(RColorBrewer)

# Customize the common words to be excluded (platform name, login, dating, etc.)
custom_stop <- c("app", "apps", "tinder", "bumble", "hinge", "okcupid",
                 "login", "account", "dating", "subscription", "log")

# Tinder
# Word segmentation & removes stop words
top_words <- all_reviews %>%
  filter(app_name == "Tinder") %>%
  unnest_tokens(word, review) %>%
  filter(!word %in% custom_stop) %>%
```

```r
  anti_join(stop_words, by = "word") %>%
  count(word, sort = TRUE) %>%
  filter(n > 30)

# plot use color schemes- Paired
set.seed(123)
wordcloud(
  words = top_words$word,
  freq = top_words$n,
  min.freq = 30,
  max.words = 120,
  random.order = FALSE,
  rot.per = 0.35,
  colors = brewer.pal(8, "Paired"),
  scale = c(3, 1)
)
```

```
## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : verification could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : matched could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : application could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : uninstalled could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : version could not be fit on page. It will not be plotted.
```


```
## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : stupid could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : meeting could not be fit on page. It will not be plotted.
```

```
# Bumble
top_words <- all_reviews %>%
  filter(app_name == "Bumble") %>%
  unnest_tokens(word, review) %>%
  filter(!word %in% custom_stop) %>%
  anti_join(stop_words, by = "word") %>%
  count(word, sort = TRUE) %>%
  filter(n > 30)


set.seed(123)
wordcloud(
  words = top_words$word,
  freq = top_words$n,
  min.freq = 30,
  max.words = 120,
  random.order = FALSE,
  rot.per = 0.35,
  colors = brewer.pal(8, "Paired"),
  scale = c(3, 1)
)
```

```
## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : awesome could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : guy could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : photos could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : stars could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : accounts could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : super could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : change could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : payment could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : photo could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : review could not be fit on page. It will not be plotted.

## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : sign could not be fit on page. It will not be plotted.
```


```
## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : email could not be fit on page. It will not be plotted.
```

```r
# Hinge
top_words <- all_reviews %>%
  filter(app_name == "Hinge") %>%
  unnest_tokens(word, review) %>%
  filter(!word %in% custom_stop) %>%
  anti_join(stop_words, by = "word") %>%
  count(word, sort = TRUE) %>%
  filter(n > 30)


set.seed(123)
wordcloud(
  words = top_words$word,
  freq = top_words$n,
  min.freq = 30,
  max.words = 120,
  random.order = FALSE,
  rot.per = 0.35,
  colors = brewer.pal(8, "Paired"),
  scale = c(3, 1)
)
```

```
## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : absolutely could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : scammers could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : swiping could not be fit on page. It will not be plotted.
```


```
## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : verification could not be fit on page. It will not be plotted.
```

```r
# OkCupid
top_words <- all_reviews %>%
  filter(app_name == "OkCupid") %>%
  unnest_tokens(word, review) %>%
  filter(!word %in% custom_stop) %>%
  anti_join(stop_words, by = "word") %>%
  count(word, sort = TRUE) %>%
  filter(n > 30)


set.seed(123)
wordcloud(
  words = top_words$word,
  freq = top_words$n,
  min.freq = 30,
  max.words = 120,
  random.order = FALSE,
  rot.per = 0.35,
  colors = brewer.pal(8, "Paired"),
  scale = c(3, 1)
)
```

```
## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : excellent could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : matching could not be fit on page. It will not be plotted.
```


```
## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : annoying could not be fit on page. It will not be plotted.
```

```
## Warning in wordcloud(words = top_words$word, freq = top_words$n, min.freq = 30,
## : download could not be fit on page. It will not be plotted.
```

Interpretation:

Keywords in the Tinder Word Cloud: fake, match, ban, scam, useless, support

- Users are mainly complaining about fake accounts, malfunctioning features and account bans.

Bumble Word Cloud Keywords: premium, cancel, features, blocked, reply

- It reflects users' concerns about payment, functional limitations and customer service issues.

Hinge & OkCupid:

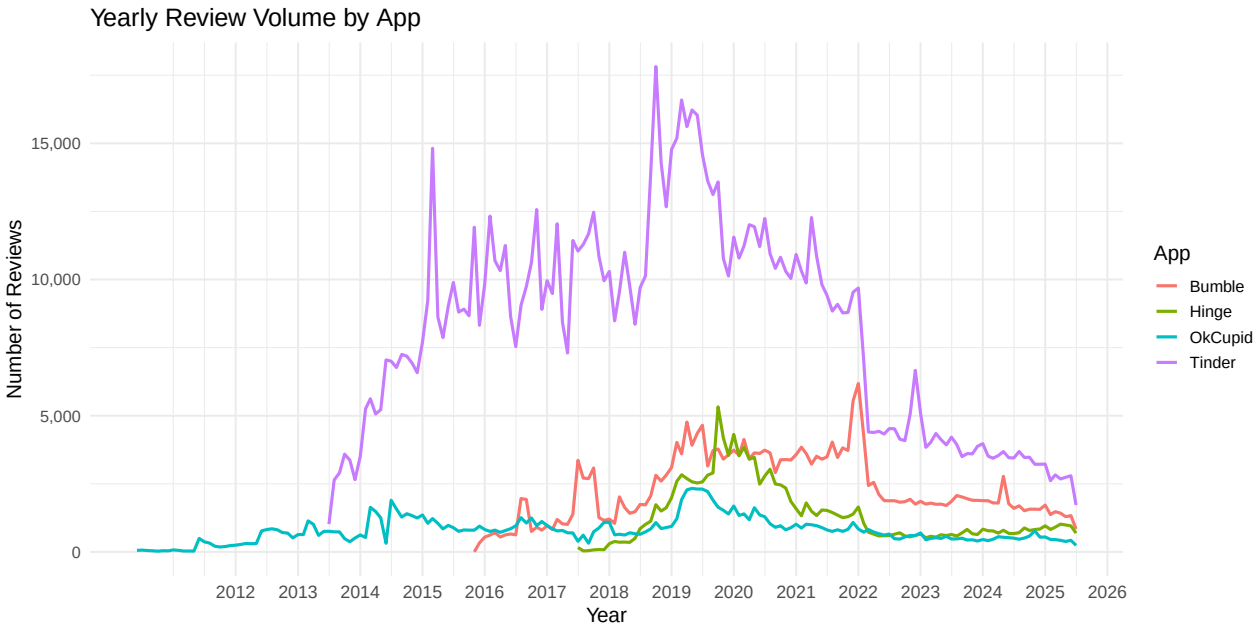
- Key words include fake, pay, chat, profile, ban, conversation
- Repeated pain points such as "fake accounts", "difficult chatting", and "poor payment experience" are mentioned.

Conclusion: User comments focus on themes such as "false information", "bans", "customer service", and "paid experience", which are consistent with the trends in ratings and emotions.

Step 5: Time Series For Numbers of Reviews

```
# 1. convert column 'date' to Date type and create new col-month
review_by_month <- all_reviews %>%
  mutate(date = as.Date(date),
         month = floor_date(date, unit = "month")) %>%
  group_by(app_name, month) %>%
  summarise(count = n(), .groups = "drop")

# 2. graph the changes in the number of comments over time
ggplot(review_by_month, aes(x = month, y = count, color = app_name)) +
  geom_line(size = 1) +
  scale_x_date(
    date_labels = "%Y",
    breaks = seq(as.Date("2012-01-01"), as.Date("2026-01-01"), by = "1 year")
  ) +
  scale_y_continuous(labels = scales::label_comma()) +
  labs(
    title = "Yearly Review Volume by App",
    x = "Year",
    y = "Number of Reviews",
    color = "App"
  ) +
  theme_minimal(base_size = 14) +
  theme(axis.text.x = element_text(angle = 0, size = 12))
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Interpretation:

Trend of annual comment volume changes

- Tinder had the highest peak in its comment count, being extremely active from 2018 to 2021;
- Hinge and Bumble also saw an increase during the pandemic, but their comment counts have decreased and stabilized in recent years;
- OkCupid had the fewest comments, with its growth stagnating.

Conclusion: Tinder has been active for a long time, but the increase in user numbers does not necessarily indicate an improvement in satisfaction. Further confirmation is needed in combination with emotional analysis.

Step 6: Time Series for Sentiment

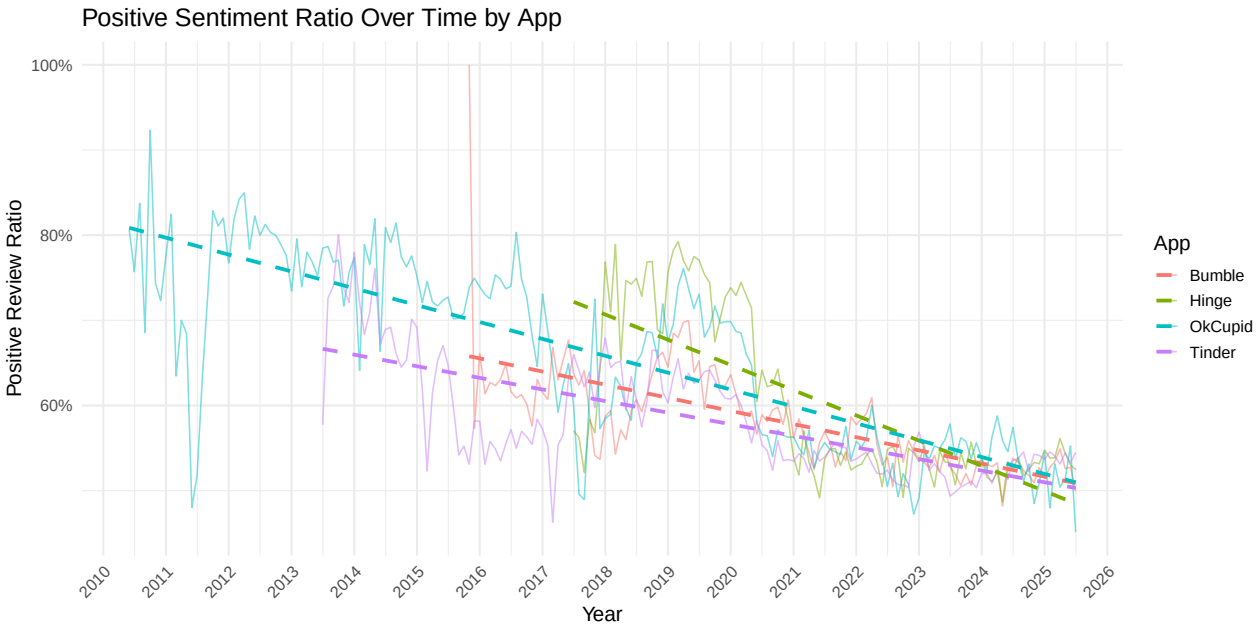
```
# Load into 'bing emotion dictionary' positive / negative
bing_lex <- get_sentiments("bing")

# Extract keyword + emotion tags
sentiment_time <- all_reviews %>%
  filter(!is.na(review), review != "") %>%
  mutate(date = as.Date(date),
         month = floor_date(date, "month")) %>%
  unnest_tokens(word, review) %>%
  inner_join(bing_lex, by = "word") %>%
  count(app_name, month, sentiment) %>%
  tidyr::pivot_wider(names_from = sentiment, values_from = n, values_fill = 0) %>%
  mutate(total = positive + negative,
         pos_ratio = positive / total)
```

```
## Warning in inner_join(., bing_lex, by = "word"): Detected an unexpected many-to-many relationship be
```

```
## i Row 10434919 of `x` matches multiple rows in `y`.
## i Row 2248 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
ggplot(sentiment_time, aes(x = month, y = pos_ratio, color = app_name)) +
  geom_line(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE, linetype = "dashed", size = 1.2) +  # regression
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  scale_x_date(date_labels = "%Y", date_breaks = "1 year") +
  labs(
    title = "Positive Sentiment Ratio Over Time by App",
    x = "Year",
    y = "Positive Review Ratio",
    color = "App"
  ) +
  theme_minimal(base_size = 14) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

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



Interpretation:

Tinder

- The proportion of positive initial comments was approximately 65%, and it remained within the range of 55% to 65% thereafter.
- Although the number of comments was the largest, the overall sentiment was relatively low.
- The trend line indicates that the sentiment has been steadily declining, but the decline is not very sharp.

Bumble

- The starting point is approximately 70%, with relatively small fluctuations, remaining around 60%.
- The slope of the trend line is gentle, indicating that the user sentiment is relatively stable.
- There are slight upward movements at several key points, performing slightly better than Tinder.

Hinge

- During the period from 2019 to 2020, the emotional peak was notably high.
- Although the trend line showed a downward trend, it remained at a relatively high level overall over the long term.

OkCupid

- The proportion of positive emotions initially reached as high as 80%, which was the highest among the four.
- However, it declined rapidly and the slope of the regression line was steep.
- By recent years, it has dropped below 55%, indicating a significant loss in user satisfaction.

Overall, the long-term emotional states and overall experiences of users for these four apps have shown an obvious downward trend (especially during and after the pandemic).

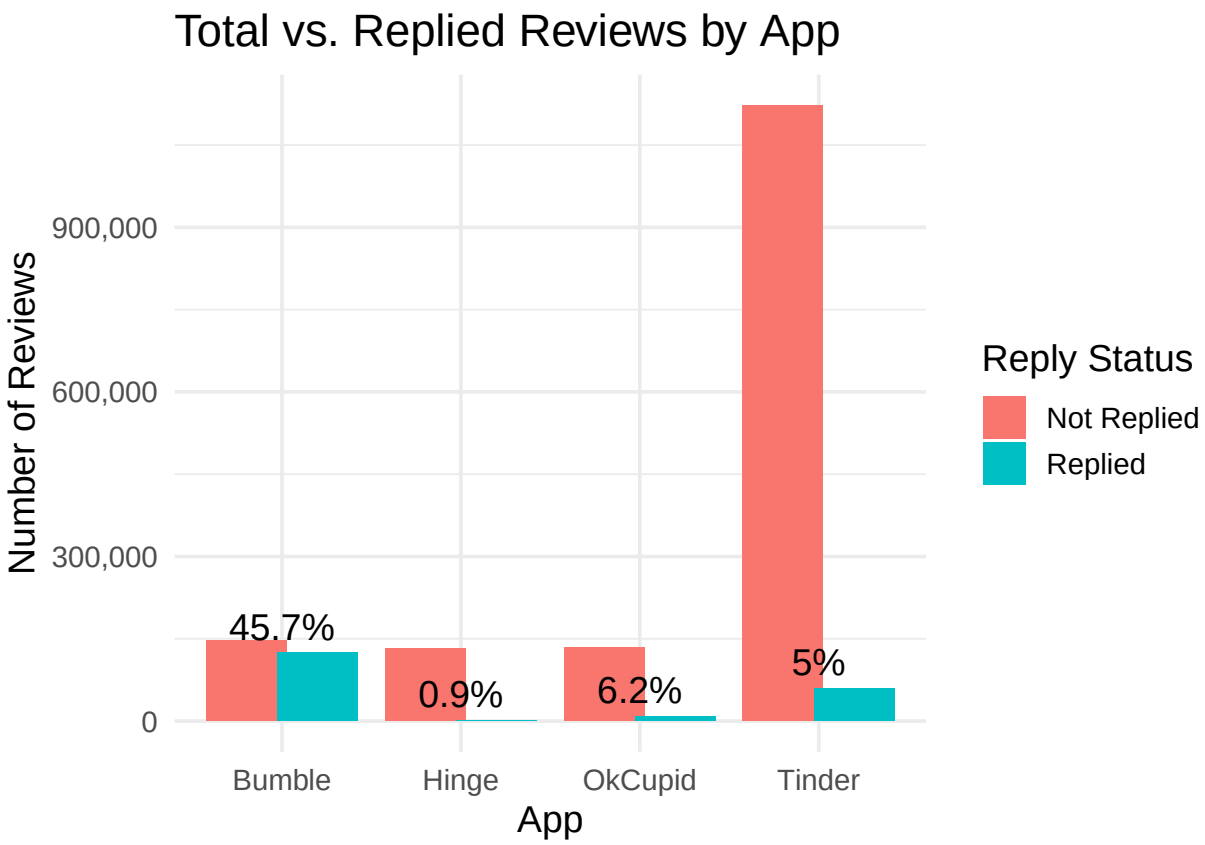
Step 7: Reply Rate

```
all_reviews %>%
  mutate(replied = !is.na(reply)) %>%
  group_by(app_name) %>%
  summarise(
    total_reviews = n(),
    replied_count = sum(replied),
    reply_rate = mean(replied)
  )
```

```
## # A tibble: 4 x 4
##   app_name total_reviews replied_count reply_rate
##   <chr>            <int>         <int>      <dbl>
## 1 Bumble          271742        124248    0.457
## 2 Hinge           132665          1231    0.00928
## 3 OkCupid         143467          8871    0.0618
## 4 Tinder         1181148         59008    0.0500
```

```
# Side-by-side barplot Replied vs Not Replied Reviews)

# summarize the data
reply_summary <- all_reviews %>%
  mutate(reply_flag = ifelse(!is.na(reply), "Replied", "Not Replied")) %>%
  group_by(app_name, reply_flag) %>%
  summarise(count = n(), .groups = "drop")
```

```r
# count the total + reply amount
reply_rate <- reply_summary %>%
  group_by(app_name) %>%
  summarise(
    total = sum(count),
    replied = sum(count[reply_flag == "Replied"]),
    reply_rate = replied / total
  )

# Merge the labeled data
reply_summary_labeled <- reply_summary %>%
  left_join(reply_rate, by = "app_name")

# plot
ggplot(reply_summary_labeled, aes(x = app_name, y = count, fill = reply_flag)) +
  geom_col(position = position_dodge(width = 0.8)) +
  geom_text(
    data = reply_summary_labeled %>% filter(reply_flag == "Replied"),
    aes(label = paste0(round(reply_rate * 100, 1), "%")),
    vjust = -0.5,
    position = position_dodge(width = 0.8),
    size = 5,
    color = "black"
  ) +
  labs(
    title = "Total vs. Replied Reviews by App",
    x = "App",
    y = "Number of Reviews",
    fill = "Reply Status"
  ) +
  scale_y_continuous(labels = scales::label_comma()) +
  theme_minimal(base_size = 14)
```



Interpretation:

- The response rate of Bumble is much higher than that of other platforms. Nearly half of the comments receive responses from the developers.
- While Hinge and Tinder have a large number of users, their responses are extremely few and the customer service responses are poor.

Conclusion: Whether developers respond to comments is closely related to the app's reputation. Bumble's high response rate may help it maintain a good user experience.