Data per top Countries & yearly trends

Javier Torralba

2023-02-16

```
knitr::opts_chunk$set(warning = FALSE, message = FALSE, echo = TRUE)
```

Reading in the data & loading libraries

```
library(ggplot2)
library(dplyr)
library(readr)
library(tidyr)
library(lubridate)
library(viridis)
```

```
users_cleaned <- read_csv("gen/temp/users_cleaned.csv")

books_cleaned <- read_csv("gen/temp/books_cleaned.csv")
```

Reducing the data in books cleaned and merging it with users cleaned

```
# Rename the variable reader id to user_id
books_cleaned <- books_cleaned %>%
  filter(date_read >= as.Date("2015-01-01") & date_read <= as.Date("2022-06-01")) %>%
  rename(user_id = `reader id`)

# Merge the books_cleaned and users_cleaned datasets by user_id
merged_data <- merge(books_cleaned, users_cleaned, by = 'user_id')
```

Now that we have used books_cleaned and users_cleaned, I will remove both data sets to make R run smoother

```
rm(books_cleaned, users_cleaned)
```

Renaming the columns for clarity

```
merged_data <- merged_data %>%
  rename(avg_rating_given_by_user = Avg.Rating,
         avg_rating_of_book = avg_rating,
         book_url = `book url`,
         user_rating = `user rating`,
         nr_ratings_by_user = `Nr.Ratings`,
         nr_reviews_by_user = `Nr.Reviews`,
         nr_books_read_by_user = `Nr.Books.Read`)
```

Finding top 10 countries for graphs

Top 10 countries based on activity

```
# Use the table function to count the number of occurrences of each unique country
country_counts <- table(merged_data$Country)

# Sort the country counts in descending order
sorted_counts <- sort(country_counts, decreasing = TRUE)

# Print the top 10 countries with the most observations
head(sorted_counts, n = 10)
```

```
##
##          Italy      Australia United Kingdom          India      Indonesia
##         937137         492260         441023         420599         380932
##        Finland       Portugal    Philippines          Egypt         Mexico
##         328552         295376         295360         288455         286477
```

Top 10 countries based on users

```
# Use the aggregate function to count the number of unique users in each country
country_user_counts <- aggregate(merged_data$user_id, by = list(merged_data$Country), FUN = function(x)

# Rename the columns of the output
names(country_user_counts) <- c("Country", "User_Count")

# Sort the country user counts in descending order
sorted_counts <- country_user_counts[order(-country_user_counts$User_Count),]

# Print the top 10 countries based on the number of users
head(sorted_counts, n = 10)
```

```
##           Country User_Count
## 11          Italy      10171
## 9           India       8413
## 10      Indonesia       7156
## 6           Egypt       5903
## 21    Philippines       3707
## 15         Mexico       3701
## 23       Portugal       3343
## 13      Lithuania       3269
## 30 United Kingdom       3189
## 2       Australia       3104
```

This data set will be used from now on to make the graphs per country

```
# Define a vector of the countries you want to keep
selected_countries <- c("Italy", "India", "Indonesia", "Egypt", "Phillipines", "Mexico", "Portugal", "Li

# Use the filter function in dplyr to filter the dataset to only keep the selected countries
top10countries <- merged_data %>%
  filter(Country %in% selected_countries)
```

Graphs for top 10 countries

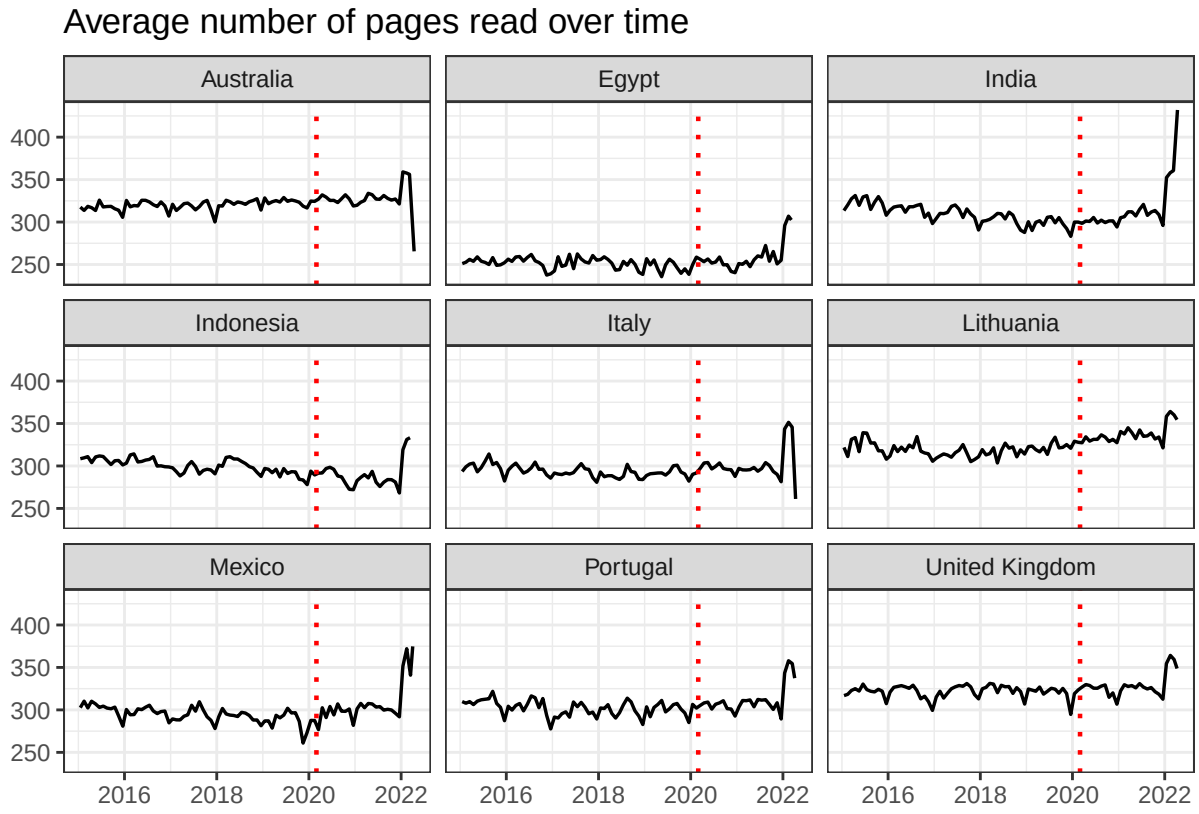
Average number of pages read per country

Preparing data

```
# Use select function to only select some columns from the top 10 countries data
top10countries_num_pages_average <- top10countries %>%
  select(date_read, num_pages, Country)

top10countries_num_pages_average <- na.omit(top10countries_num_pages_average)

# summarizing the data per month
top10countries_num_pages_average <- top10countries_num_pages_average %>%
  mutate(month_start = floor_date(date_read, unit = "month")) %>%
  group_by(month_start, Country) %>%
  summarize_all(mean)
```

Making the graph

```
# Making the graph
ggplot(data = top10countries_num_pages_average, aes(x = date_read, y = num_pages)) +
  geom_line(size = 0.6) +
  facet_wrap(~Country) +
  geom_vline(xintercept = as.Date("2020-03-01"), color = "red", linetype = "dotted", size = 0.8) +
  theme_bw() +
  labs(title = "Average number of pages read over time", y = "", x = "")
```



Removing the data that will no longer be used

```
rm(country_user_counts, sorted_counts, top10countries_num_pages_average)
```

User rating given over time per country

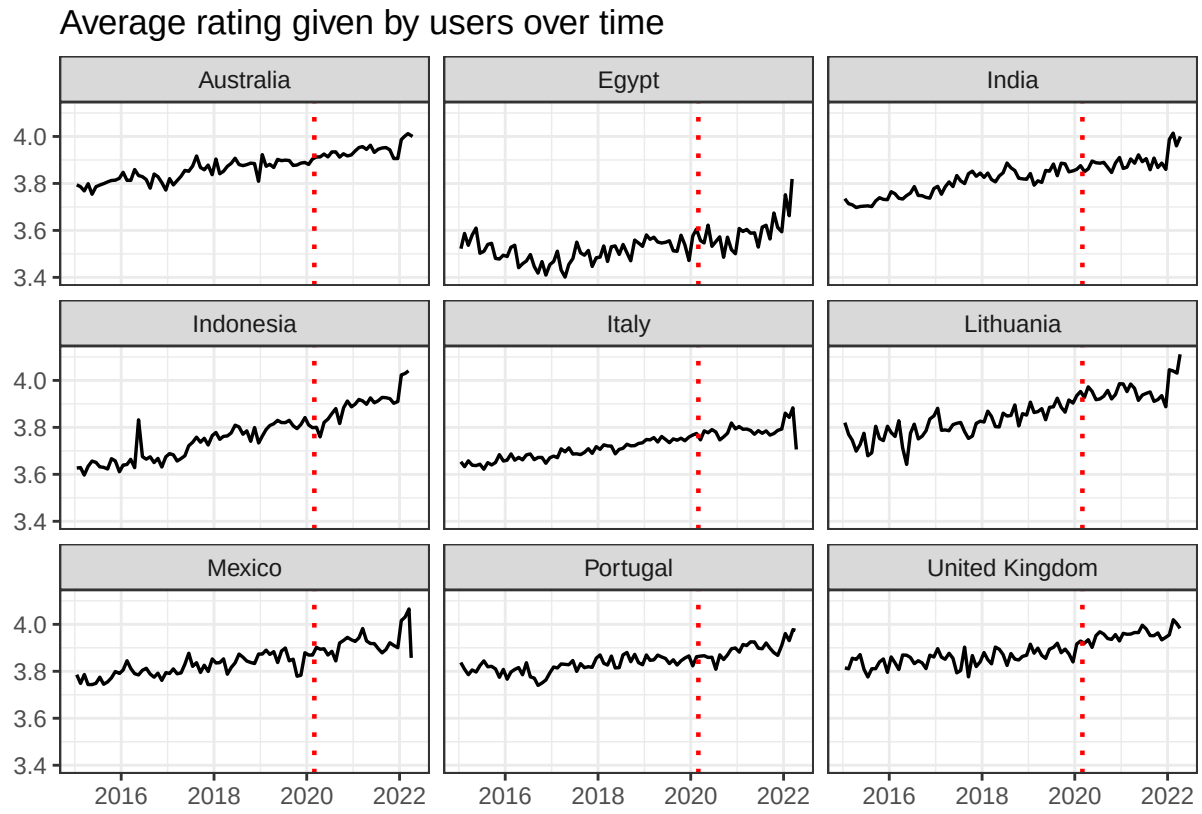
Preparing data

```
# Use select function to only select some columns from the top 10 countries data
top10countries_user_rating <- top10countries %>%
  select(date_read, user_rating, Country)

top10countries_user_rating <- na.omit(top10countries_user_rating)

# summarizing the data to have the average rating given by user at the start of each month
 top10countries_user_rating <- top10countries_user_rating %>%
  mutate(month_start = floor_date(date_read, unit = "month")) %>%
  group_by(month_start, Country) %>%
  summarize_all(mean)
```

Making the graph

```
# Making the graph
ggplot(data = top10countries_user_rating, aes(x = date_read, y = user_rating)) +
  geom_line(size = 0.6) +
```

```
  facet_wrap(~Country) +
  geom_vline(xintercept = as.Date("2020-03-01"), color = "red", linetype = "dotted", size = 0.8) +
  theme_bw() +
  labs(title = "Average rating given by users over time", y = "", x = "")
```



Removing data

```
rm(top10countries_user_rating)
```

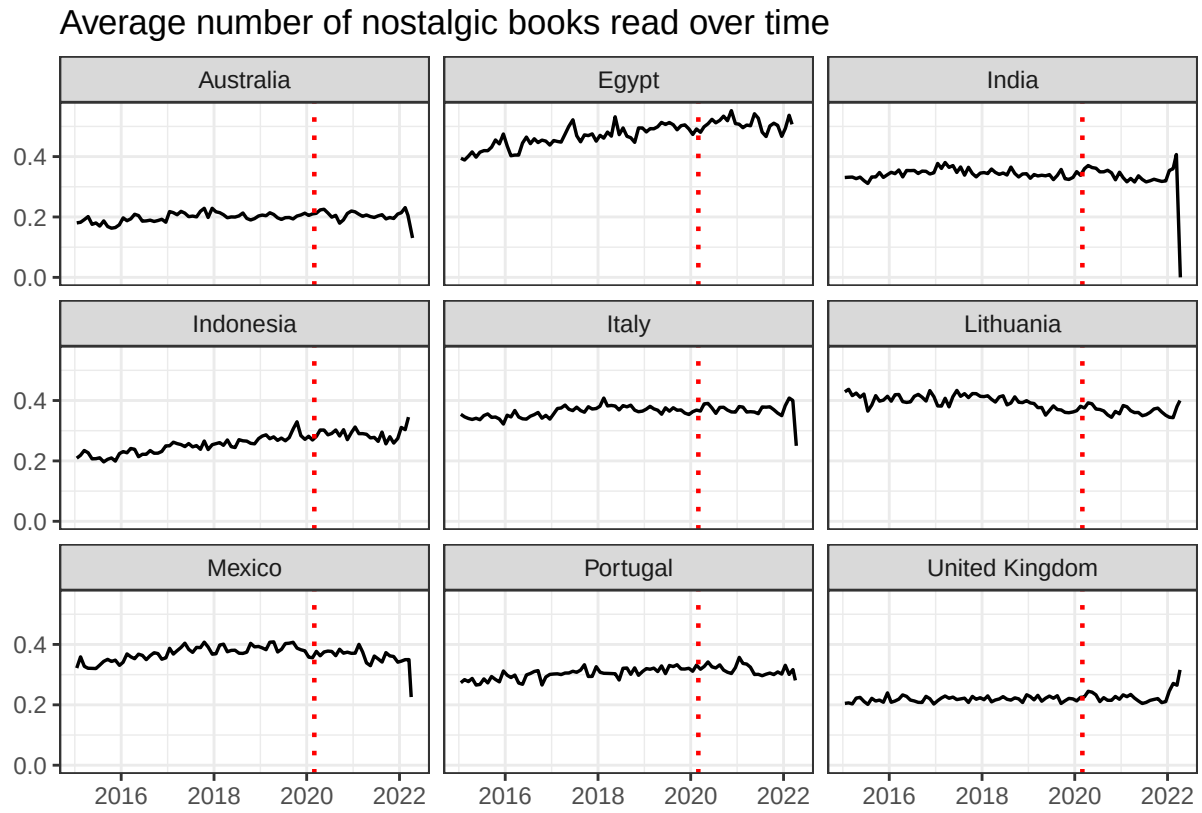
Nostalgic books read over time

Preparing data for graph

```
# Use select function to only select some columns from the top 10 countries data
top10countries_nostalgic <- top10countries %>%
  select(date_read, nostalgic, Country)

top10countries_nostalgic <- na.omit(top10countries_nostalgic)

# summarizing the data per month
 top10countries_nostalgic <- top10countries_nostalgic %>%
  mutate(month_start = floor_date(date_read, unit = "month")) %>%
  group_by(month_start, Country) %>%
  summarize_all(mean)
```

Making the graph

```
# Making the graph
ggplot(data = top10countries_nostalgic, aes(x = date_read, y = nostalgic)) +
  geom_line(size = 0.6) +
  facet_wrap(~Country) +
  geom_vline(xintercept = as.Date("2020-03-01"), color = "red", linetype = "dotted", size = 0.8) +
  theme_bw() +
  labs(title = "Average number of nostalgic books read over time", y = "", x = "")
```



Removing the data

```
rm(top10countries_nostalgic)
```

Books per day over time

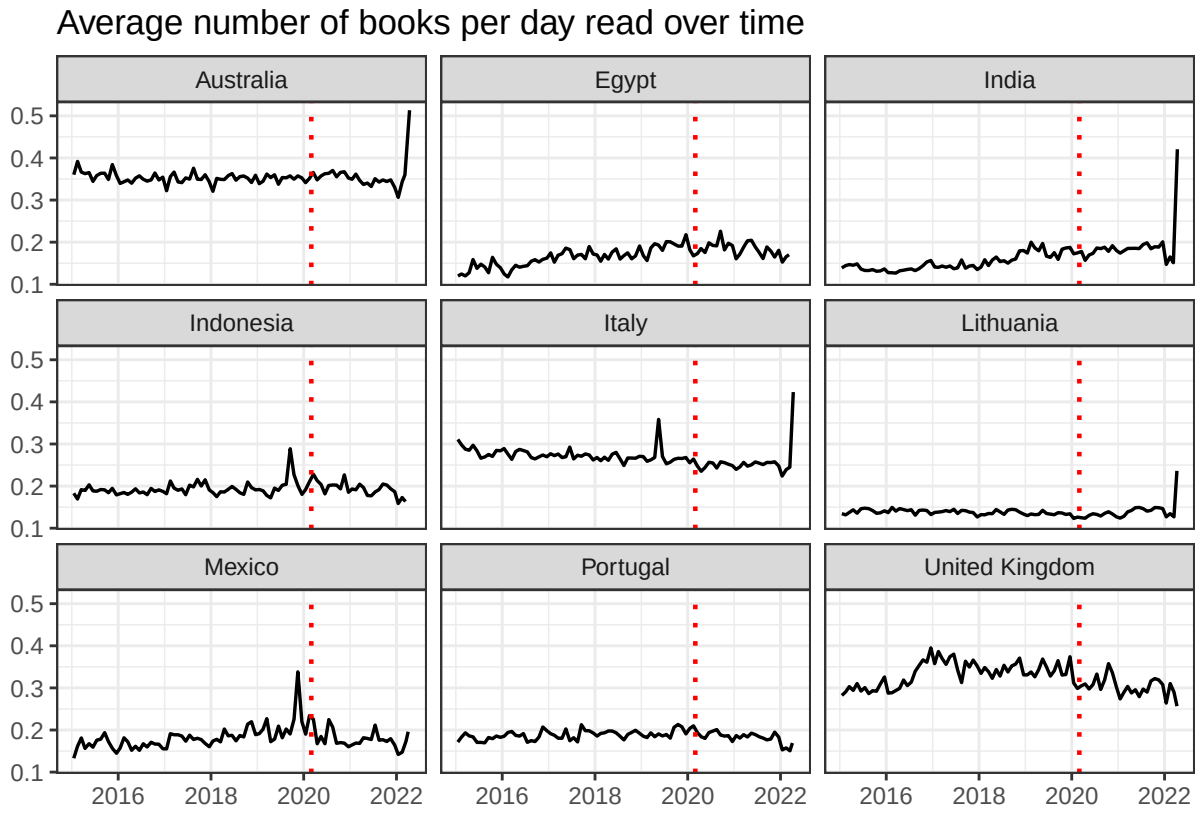
Preparing data for graph

```
# Use select function to only select some columns from the top 10 countries data
top10countries_books_per_day <- top10countries %>%
  select(date_read, books_per_day, Country)

top10countries_books_per_day <- na.omit(top10countries_books_per_day)

# summarizing the data per month
```

```
top10countries_books_per_day <- top10countries_books_per_day %>%
 mutate(month_start = floor_date(date_read, unit = "month")) %>%
 group_by(month_start, Country) %>%
 summarize_all(mean)
```

Making the graph

```
# Making the graph
ggplot(data = top10countries_books_per_day, aes(x = date_read, y = books_per_day)) +
  geom_line(size = 0.6) +
  facet_wrap(~Country) +
  geom_vline(xintercept = as.Date("2020-03-01"), color = "red", linetype = "dotted", size = 0.8) +
  theme_bw() +
  labs(title = "Average number of books per day read over time", y = "", x = "")
```



Removing the data

```
rm(top10countries_books_per_day)
```

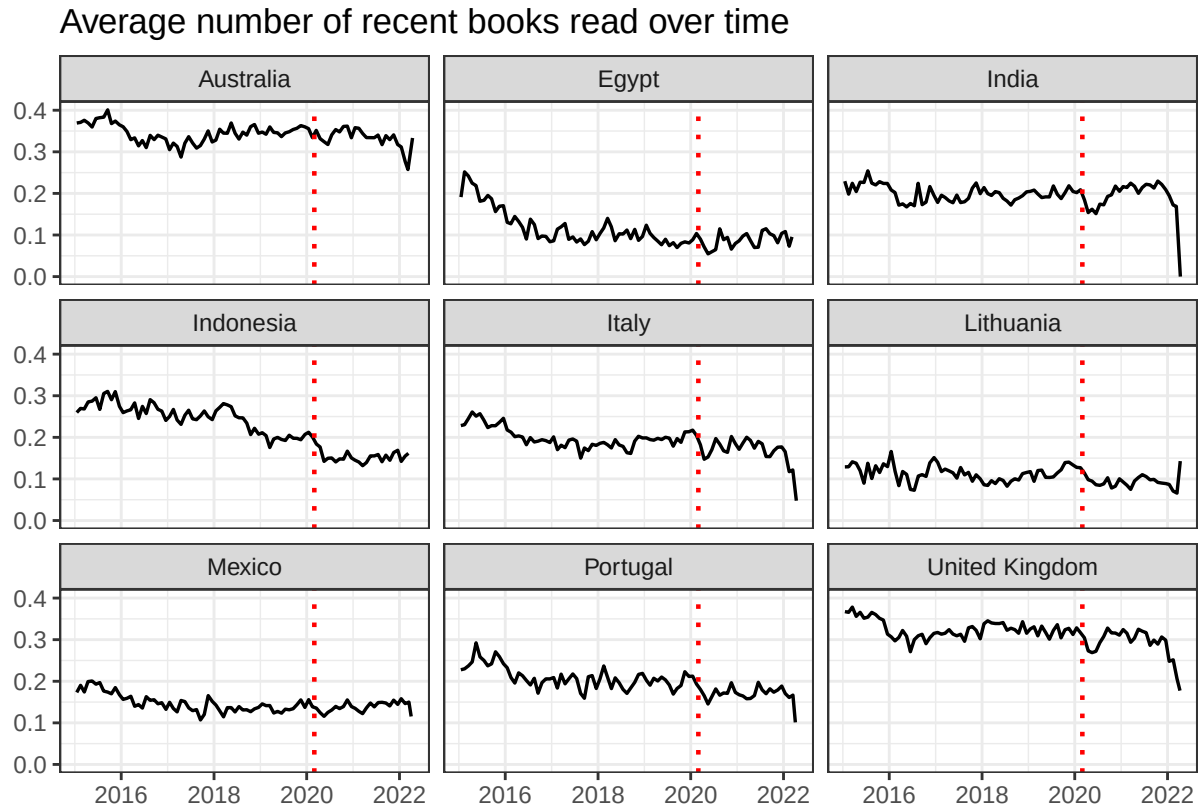
Recent books over time

Preparing data for graph

```r
# Use select function to only select some columns from the top 10 countries data
top10countries_recent <- top10countries %>%
  select(date_read, recent, Country)

top10countries_recent <- na.omit(top10countries_recent)

# summarizing the data per month
 top10countries_recent <- top10countries_recent %>%
  mutate(month_start = floor_date(date_read, unit = "month")) %>%
  group_by(month_start, Country) %>%
  summarize_all(mean)
```

Making the graph

```r
# Making the graph
ggplot(data = top10countries_recent, aes(x = date_read, y = recent)) +
  geom_line(size = 0.6) +
  facet_wrap(~Country) +
  geom_vline(xintercept = as.Date("2020-03-01"), color = "red", linetype = "dotted", size = 0.8) +
  theme_bw() +
  labs(title = "Average number of recent books read over time", y = "", x = "")
```



Removing the data

```
rm(top10countries_recent, top10countries)
```

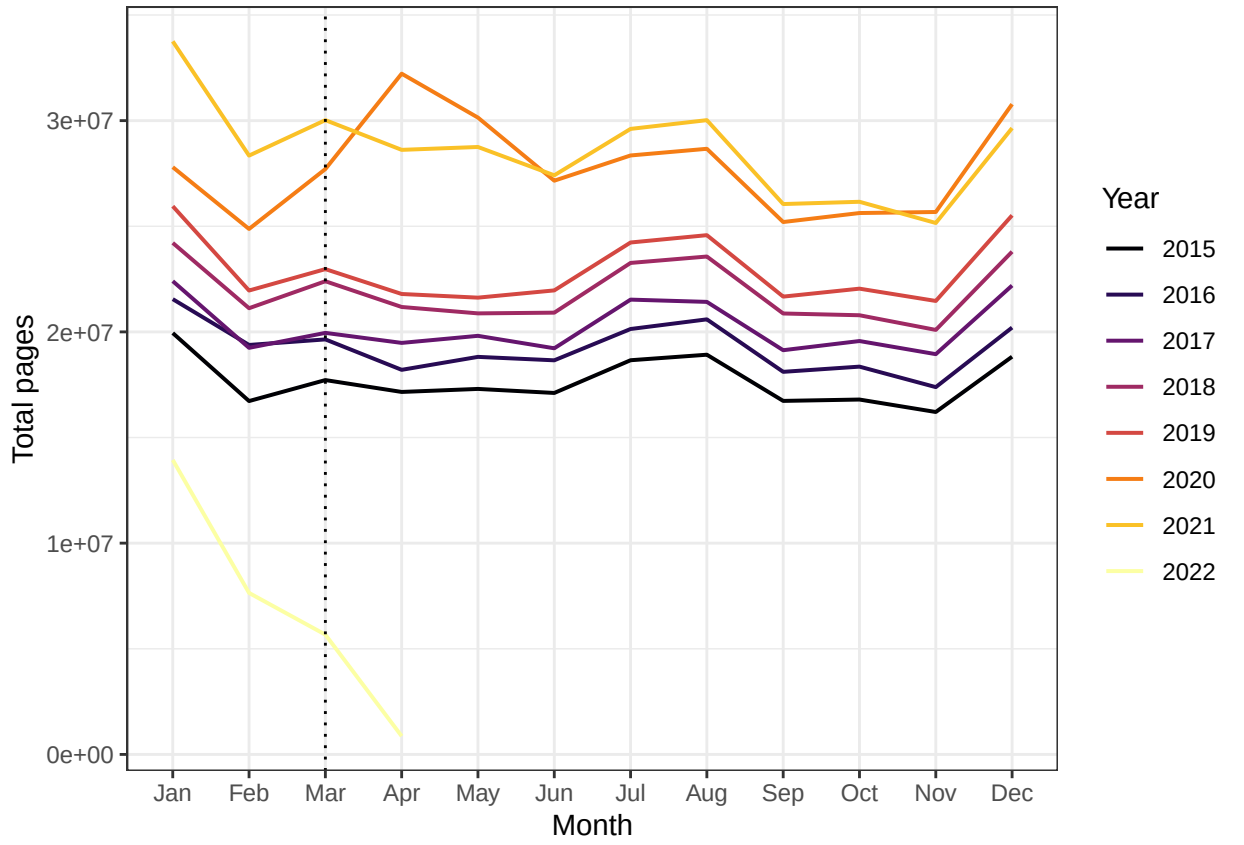
Number of pages by all users in a given month

Preparing the data

```
# Select columns and drop NAs
number_of_pages_by_user <- merged_data %>%
  select(user_id, date_read, num_pages) %>%
  drop_na(num_pages)

# group by month and user, and sum the num_pages column
number_of_pages_by_user_sum <- number_of_pages_by_user %>%
  mutate(month = format(date_read, "%Y-%m")) %>%
  group_by(month) %>%
  summarize(total_pages = sum(num_pages))

# convert month to a date format
number_of_pages_by_user_sum$month <- as.Date(paste0(number_of_pages_by_user_sum$month, "-01"))

# create a year column
number_of_pages_by_user_sum$year <- format(number_of_pages_by_user_sum$month, "%Y")
```

Making the graph

```
# create the line graph
ggplot(number_of_pages_by_user_sum, aes(x = format(month, "%b"), y = total_pages, group = year, color =
  geom_line(size = 0.7) +
  scale_x_discrete(name = "Month", limits = month.abb) +
  scale_color_viridis(discrete = TRUE, option = "inferno") +
  labs(y = "Total pages", color = "Year") +
  theme_bw() +
  geom_vline(xintercept = which(month.abb == "Mar"), linetype = "dotted", color = "black")
```



Remove data

```
rm(number_of_pages_by_user, number_of_pages_by_user_sum)
```

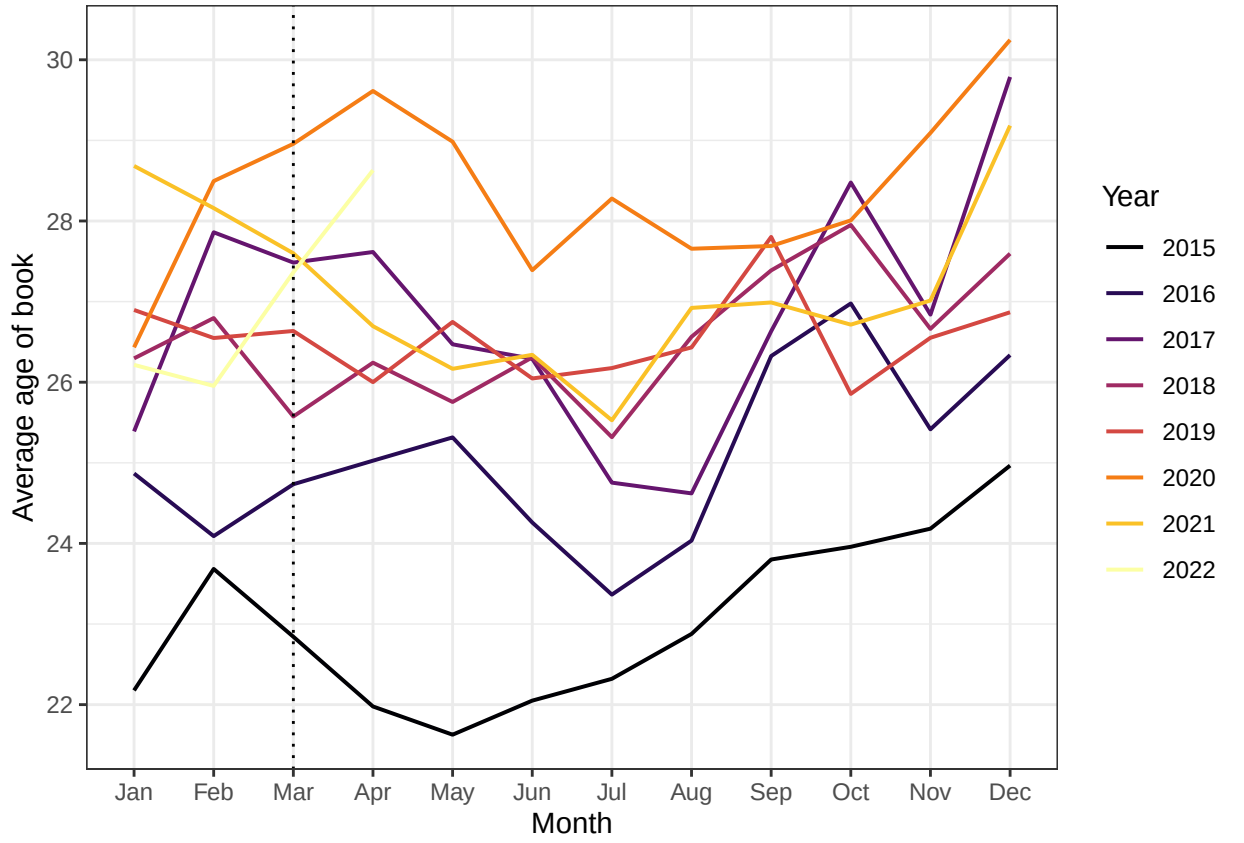
Average age of book per user each month

Preparing the data

```
age_of_book2 <- merged_data %>% select(user_id, date_read, age_of_book) %>%
  na.omit()

# group by month and user, and do average of age of book
age_of_book_average2 <- age_of_book2 %>%
  mutate(month = format(date_read, "%Y-%m"))

# convert month to a date format
age_of_book_average2$month <- as.Date(paste0(age_of_book_average2$month, "-01"))

# create a year column
age_of_book_average2$year <- format(age_of_book_average2$month, "%Y")

# Sum total number of pages per year
result2 <- age_of_book_average2 %>%
  group_by(year, month) %>%
  summarise(average_age_of_book = mean(age_of_book))
```

Making the graph

```
# create the line graph
ggplot(result2, aes(x = format(month, "%b"), y = average_age_of_book, group = year, color = year)) +
  geom_line(size = 0.7) +
  scale_x_discrete(name = "Month", limits = month.abb) +
  scale_color_viridis(discrete = TRUE, option = "inferno") +
  labs(y = "Average age of book", color = "Year") +
  theme_bw() +
  geom_vline(xintercept = which(month.abb == "Mar"), linetype = "dotted", color = "black")
```



Removing data

```
rm(age_of_book, age_of_book_average, age_of_book_average2, age_of_book2, result, result2)
```

Average nostalgia per month

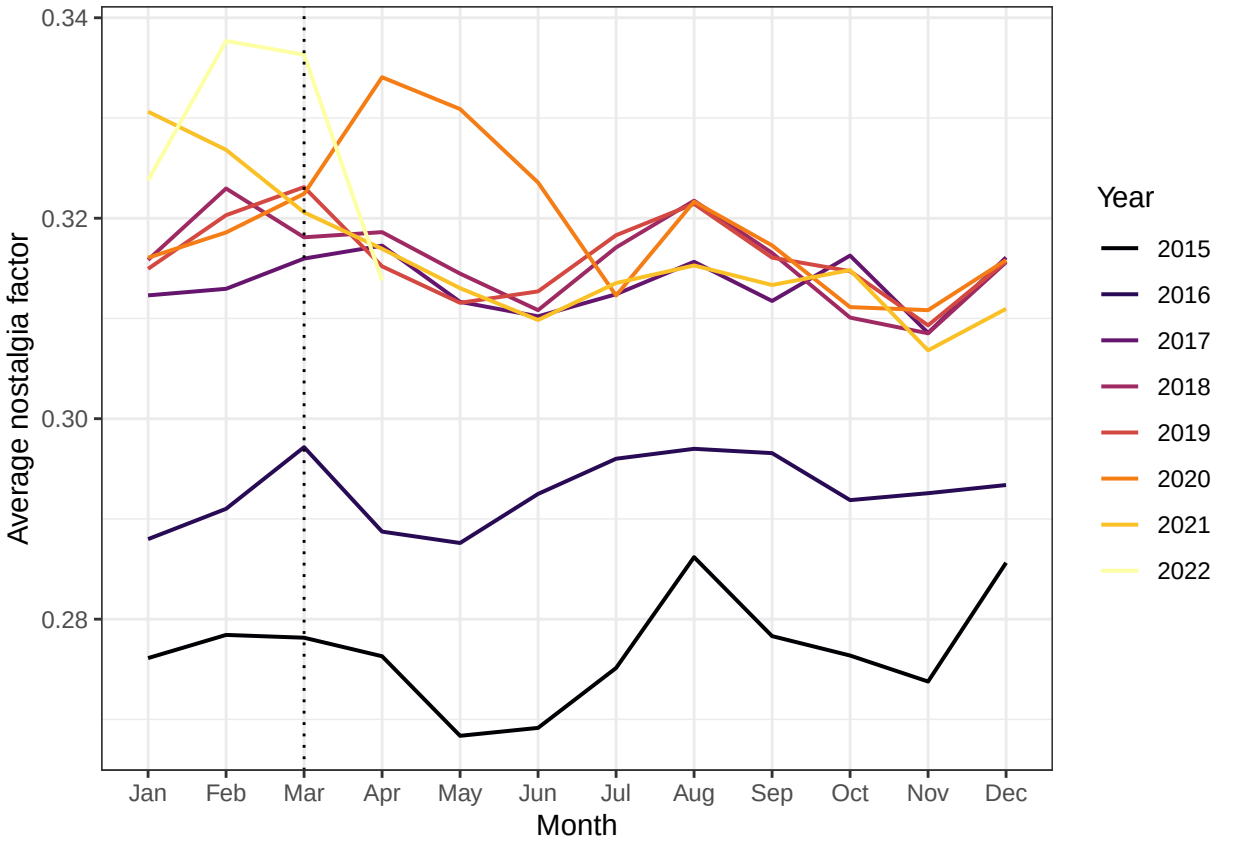
Preparing the data

```
nostalgia_per_month <- merged_data %>%
  select(user_id, date_read, nostalgic) %>%
  na.omit()

# group by month and user, and do average of age of book
nostalgia_per_month <- nostalgia_per_month %>%
```

```
  mutate(month = format(date_read, "%Y-%m"))

# convert month to a date format
nostalgia_per_month$month <- as.Date(paste0(nostalgia_per_month$month, "-01"))

# create a year column
nostalgia_per_month$year <- format(nostalgia_per_month$month, "%Y")

# Calculating average nostalgia factor per month
result <- nostalgia_per_month %>%
  group_by(year, month) %>%
  summarise(average_nostalgia_per_month = mean(nostalgic))
```

Making the graph

```
# create the line graph
ggplot(result, aes(x = format(month, "%b"), y = average_nostalgia_per_month, group = year, color = year)
  geom_line(size = 0.7) +
  scale_x_discrete(name = "Month", limits = month.abb) +
  scale_color_viridis(discrete = TRUE, option = "inferno") +
  labs(y = "Average nostalgia factor", color = "Year") +
  theme_bw() +
  geom_vline(xintercept = which(month.abb == "Mar"), linetype = "dotted", color = "black")
```



Removing data

```
rm(nostalgia_per_month, result)
```

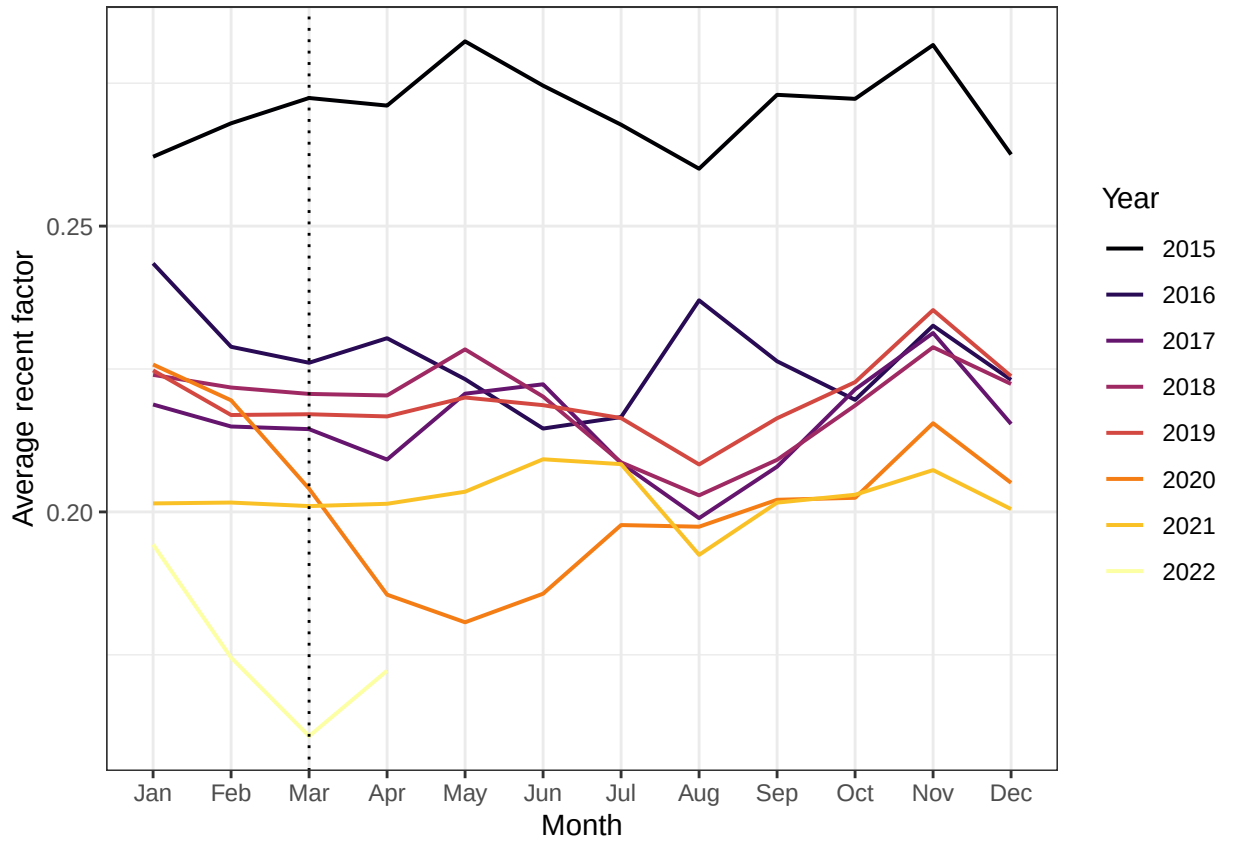
Average recent factor per month

Preparing the data

```
recent_per_month <- merged_data %>% select(user_id, date_read, recent) %>%
  na.omit()

# group by month and user, and do average of age of book
recent_per_month <- recent_per_month %>%
  mutate(month = format(date_read, "%Y-%m"))

# convert month to a date format
recent_per_month$month <- as.Date(paste0(recent_per_month$month, "-01"))

# create a year column
recent_per_month$year <- format(recent_per_month$month, "%Y")

# Sum total number of pages per year
result <- recent_per_month %>%
  group_by(year, month) %>%
  summarise(average_recent_per_month = mean(recent))
```

Making the graph

```
# create the line graph
ggplot(result, aes(x = format(month, "%b"), y = average_recent_per_month, group = year, color = year)) +
  geom_line(size = 0.7) +
  scale_x_discrete(name = "Month", limits = month.abb) +
  scale_color_viridis(discrete = TRUE, option = "inferno") +
  labs(y = "Average recent factor", color = "Year") +
  theme_bw() +
  geom_vline(xintercept = which(month.abb == "Mar"), linetype = "dotted", color = "black")
```



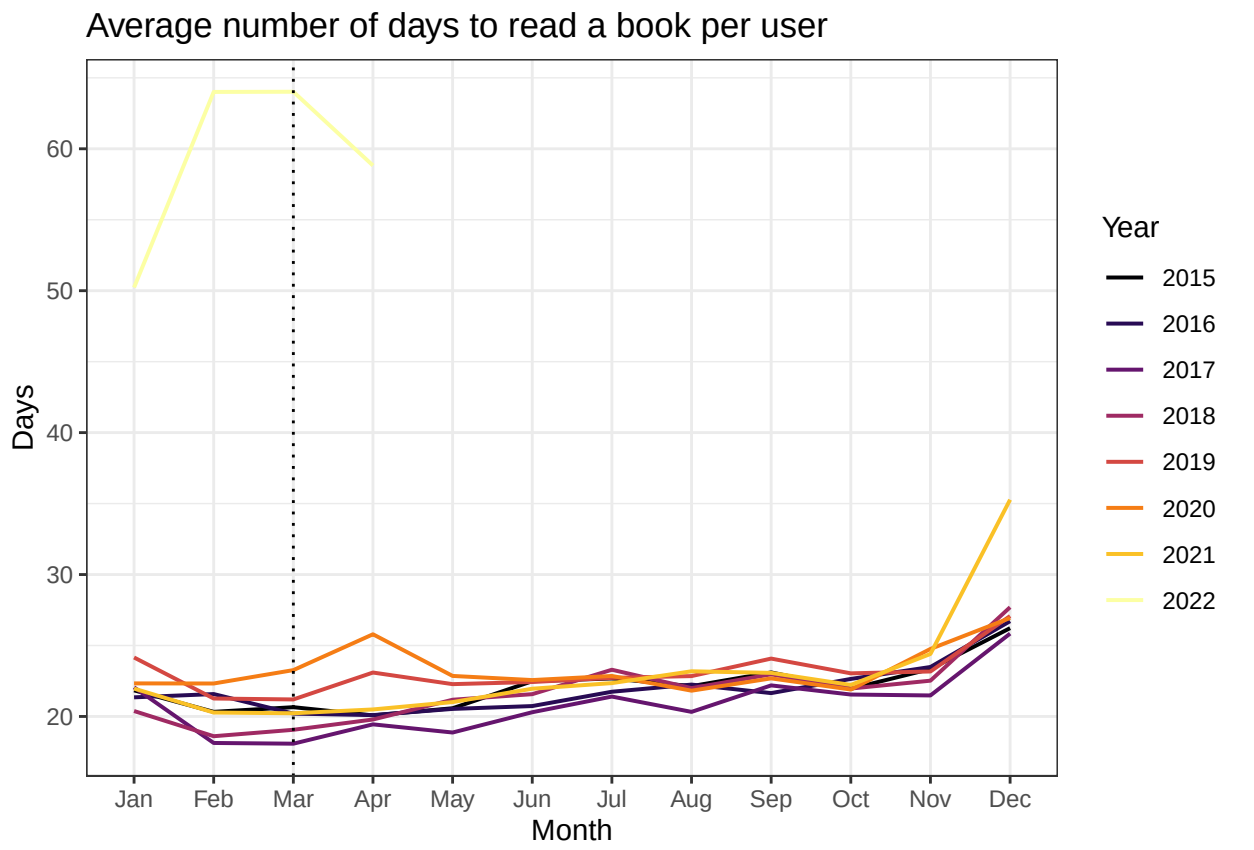

```
rm(recent_per_month, result)
```

Average reading time in days per user SECOND ATTEMPT

```
read_time_days_per_month <- merged_data %>%
  select(user_id, date_read, read_time_days) %>%
  na.omit()

# group by month and user, and do average of age of book
read_time_days_per_month <- read_time_days_per_month %>%
  mutate(month = format(date_read, "%Y-%m"))

# convert month to a date format
read_time_days_per_month$month <- as.Date(paste0(read_time_days_per_month$month, "-01"))

# create a year column
read_time_days_per_month$year <- format(read_time_days_per_month$month, "%Y")

# Computing the average time it took to read for all users in a month
jt1 <- read_time_days_per_month %>%
  group_by(month, year) %>%
  summarise(avg_read_time = mean(read_time_days))
```

Making the graph

```
# create the line graph
ggplot(jt1, aes(x = format(month, "%b"), y = avg_read_time, group = year, color = year)) +
  geom_line(size = 0.7) +
  scale_x_discrete(name = "Month", limits = month.abb) +
  scale_color_viridis(discrete = TRUE, option = "inferno") +
  labs(y = "Days", color = "Year", title = "Average number of days to read a book per user") +
  theme_bw() +
  geom_vline(xintercept = which(month.abb == "Mar"), linetype = "dotted", color = "black")
```



Removing the data

```
rm(read_time_days_per_month, result, result2, jt1)
```

Book counts per month

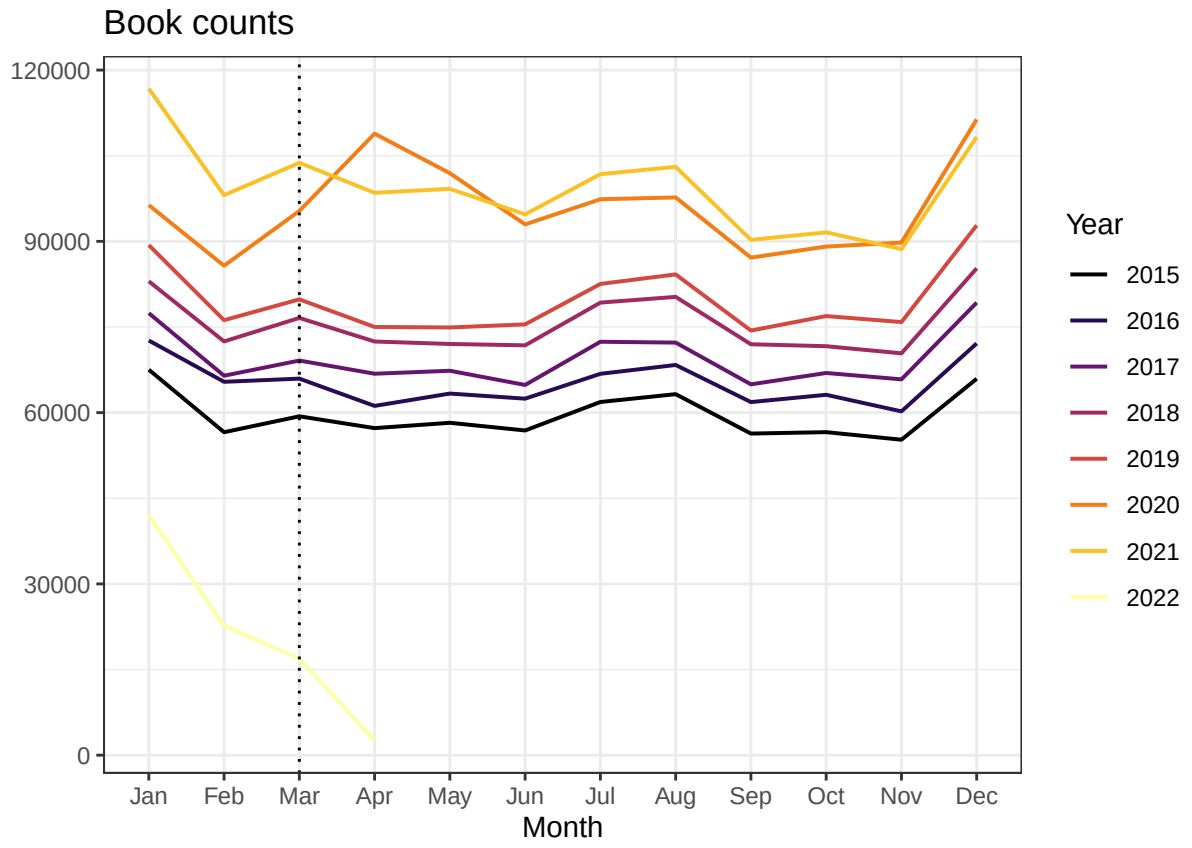
Preparing the data

```
 for_book_count <- merged_data %>% select(user_id, date_read) %>%
  na.omit()

# Convert the date_read column to a Date object and create a new column for the month
for_book_count$date_read <- as.Date(for_book_count$date_read)

for_book_count$month <- format(for_book_count$date_read, "%Y-%m")
```

```r
# Group the data by month, and count the number of entries
book_count <- for_book_count %>%
  group_by(month) %>%
  summarize(count = n())

# convert month to a date format
book_count$month <- as.Date(paste0(book_count$month, "-01"))

# create a year column
book_count$year <- format(book_count$month, "%Y")
```

Making the graph

```r
# create the line graph
ggplot(book_count, aes(x = format(month, "%b"), y = count, group = year, color = year)) +
  geom_line(size = 0.7) +
  scale_x_discrete(name = "Month", limits = month.abb) +
  scale_color_viridis(discrete = TRUE, option = "inferno") +
  labs(y = "", color = "Year", title = "Book counts") +
  theme_bw() +
  geom_vline(xintercept = which(month.abb == "Mar"), linetype = "dotted", color = "black")
```



Removing data

```
rm(book_count)
```