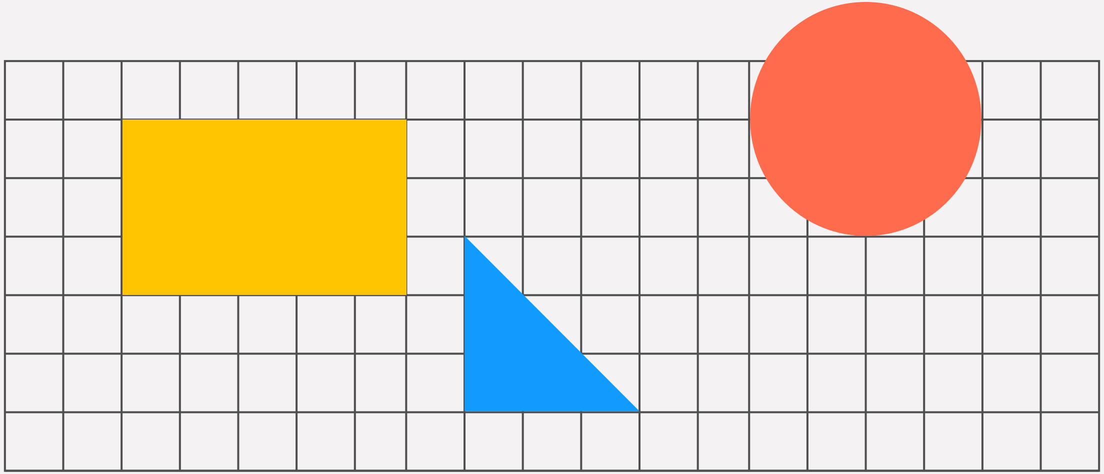
Analysing User Sentiment Shifts

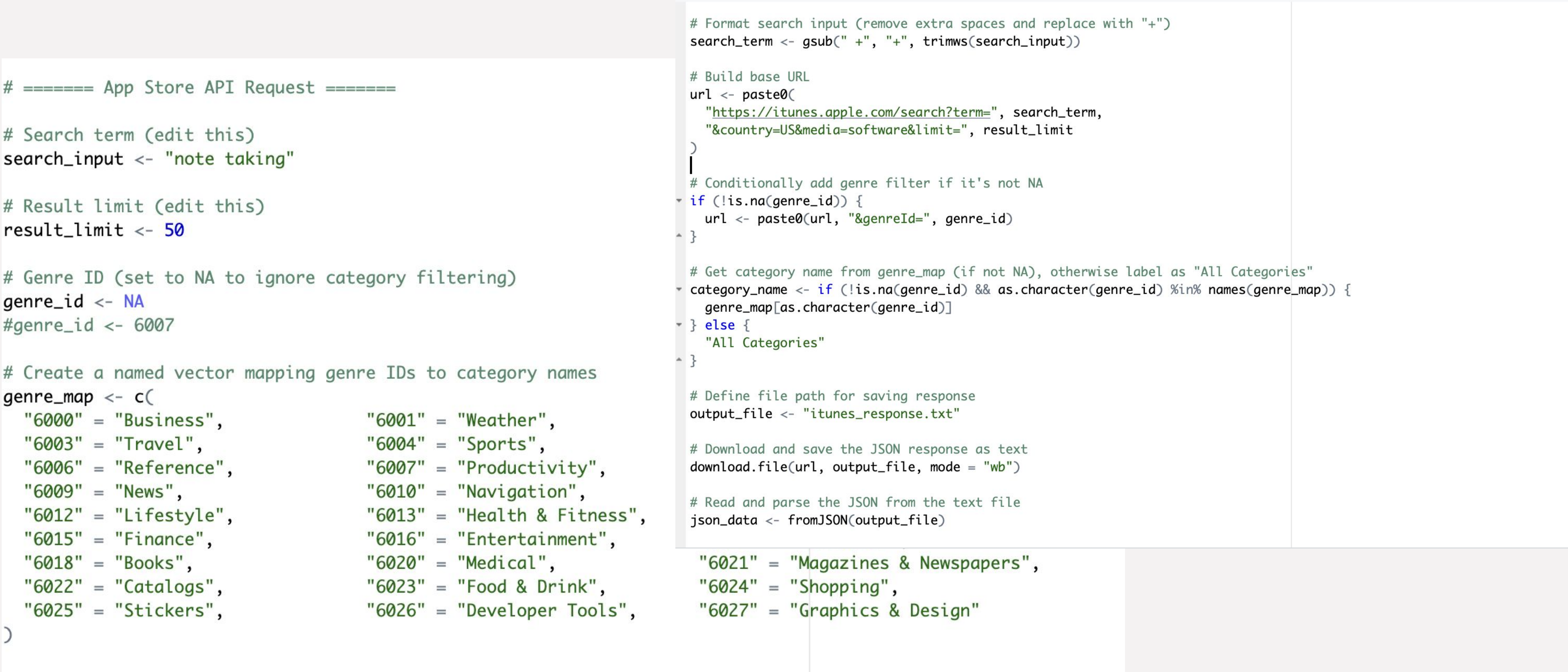
Progress Report

EPPS 6323 Knowledge Mining
March 25th, 2025
By Oliver Myers



Gathering App Names and their ID’s

For this first pass of the project 10 app were collected for

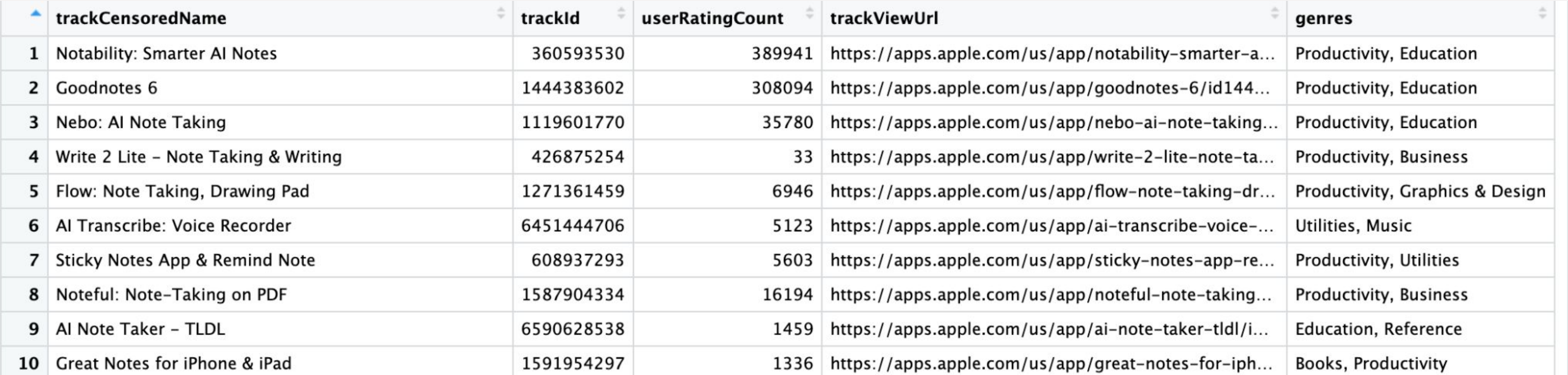


```
# ======= App Store API Request =======

# Search term (edit this)
search_input <- "note taking"

# Result limit (edit this)
result_limit <- 50

# Genre ID (set to NA to ignore category filtering)
genre_id <- NA
#genre_id <- 6007

# Create a named vector mapping genre IDs to category names
genre_map <- c(
  "6000" = "Business",          "6001" = "Weather",
  "6003" = "Travel",            "6004" = "Sports",
  "6006" = "Reference",         "6007" = "Productivity",
  "6009" = "News",              "6010" = "Navigation",
  "6012" = "Lifestyle",         "6013" = "Health & Fitness",
  "6015" = "Finance",           "6016" = "Entertainment",
  "6018" = "Books",             "6020" = "Medical",            "6021" = "Magazines & Newspapers",
  "6022" = "Catalogs",          "6023" = "Food & Drink",       "6024" = "Shopping",
  "6025" = "Stickers",          "6026" = "Developer Tools",    "6027" = "Graphics & Design"
)
```

```
# Format search input (remove extra spaces and replace with "+")
search_term <- gsub(" +", "+", trimws(search_input))

# Build base URL
url <- paste0(
  "https://itunes.apple.com/search?term=", search_term,
  "&country=US&media=software&limit=", result_limit
)

# Conditionally add genre filter if it's not NA
if (!is.na(genre_id)) {
  url <- paste0(url, "&genreId=", genre_id)
}

# Get category name from genre_map (if not NA), otherwise label as "All Categories"
category_name <- if (!is.na(genre_id) && as.character(genre_id) %in% names(genre_map)) {
  genre_map[as.character(genre_id)]
} else {
  "All Categories"
}

# Define file path for saving response
output_file <- "itunes_response.txt"

# Download and save the JSON response as text
download.file(url, output_file, mode = "wb")

# Read and parse the JSON from the text file
json_data <- fromJSON(output_file)
```

Gathering App Names and their ID's

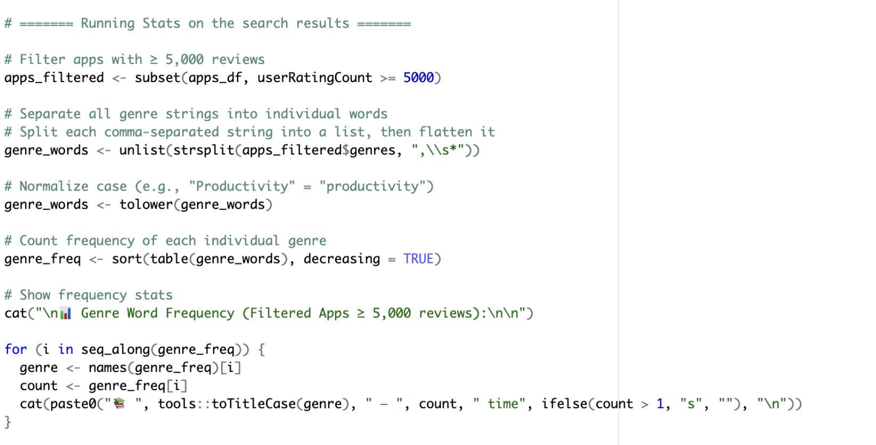
For this first pass of the project 10 app were collected for



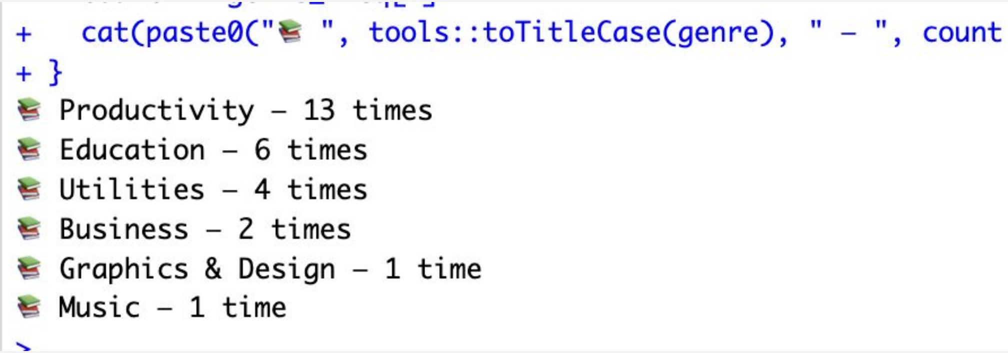
	trackCensoredName	trackId	userRatingCount	trackViewUrl	genres
1	Notability: Smarter AI Notes	360593530	389941	https://apps.apple.com/us/app/notability-smarter-a...	Productivity, Education
2	Goodnotes 6	1444383602	308094	https://apps.apple.com/us/app/goodnotes-6/id144...	Productivity, Education
3	Nebo: AI Note Taking	1119601770	35780	https://apps.apple.com/us/app/nebo-ai-note-taking...	Productivity, Education
4	Write 2 Lite - Note Taking & Writing	426875254	33	https://apps.apple.com/us/app/write-2-lite-note-ta...	Productivity, Business
5	Flow: Note Taking, Drawing Pad	1271361459	6946	https://apps.apple.com/us/app/flow-note-taking-dr...	Productivity, Graphics & Design
6	AI Transcribe: Voice Recorder	6451444706	5123	https://apps.apple.com/us/app/ai-transcribe-voice-...	Utilities, Music
7	Sticky Notes App & Remind Note	608937293	5603	https://apps.apple.com/us/app/sticky-notes-app-re...	Productivity, Utilities
8	Noteful: Note-Taking on PDF	1587904334	16194	https://apps.apple.com/us/app/noteful-note-taking...	Productivity, Business
9	AI Note Taker - TLDL	6590628538	1459	https://apps.apple.com/us/app/ai-note-taker-tldl/i...	Education, Reference
10	Great Notes for iPhone & iPad	1591954297	1336	https://apps.apple.com/us/app/great-notes-for-iph...	Books, Productivity

Getting stats on this list

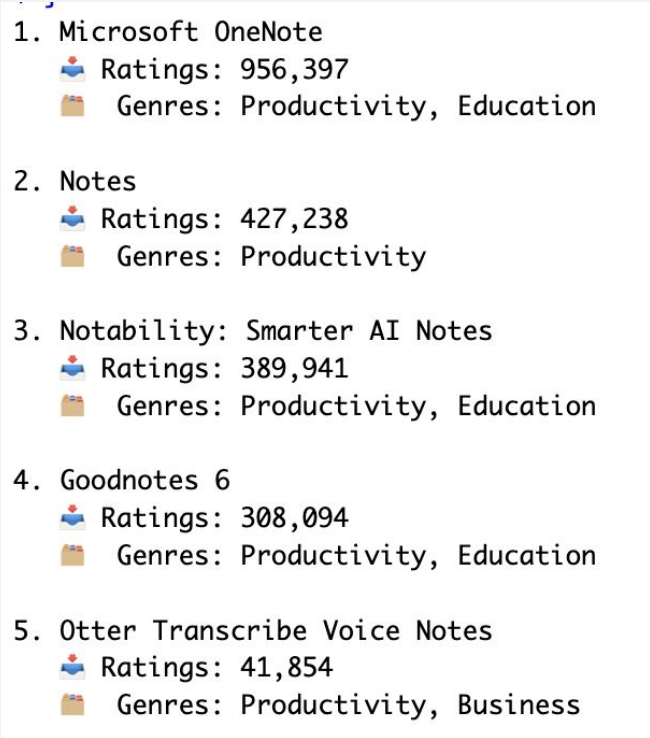
I wanted to know **genre frequency** and the **top 5 apps** to be our test set (for the stats I used 50+ apps filtered for those that had at least 5k reviews



```r
# ======= Running Stats on the search results =======

# Filter apps with ≥ 5,000 reviews
apps_filtered <- subset(apps_df, userRatingCount >= 5000)

# Separate all genre strings into individual words
# Split each comma-separated string into a list, then flatten it
genre_words <- unlist(strsplit(apps_filtered$genres, ",\\s*"))

# Normalize case (e.g., "Productivity" = "productivity")
genre_words <- tolower(genre_words)

# Count frequency of each individual genre
genre_freq <- sort(table(genre_words), decreasing = TRUE)

# Show frequency stats
cat("\n📊 Genre Word Frequency (Filtered Apps ≥ 5,000 reviews):\n\n")

for (i in seq_along(genre_freq)) {
  genre <- names(genre_freq)[i]
  count <- genre_freq[i]
  cat(paste0("📚 ", tools::toTitleCase(genre), " - ", count, " time", ifelse(count > 1, "s", ""), "\n"))
}
```



```
+   cat(paste0("📚 ", tools::toTitleCase(genre), " - ", count
+ }
📚 Productivity - 13 times
📚 Education - 6 times
📚 Utilities - 4 times
📚 Business - 2 times
📚 Graphics & Design - 1 time
📚 Music - 1 time
>
```



```
1. Microsoft OneNote
   📥 Ratings: 956,397
   🗃️  Genres: Productivity, Education

2. Notes
   📥 Ratings: 427,238
   🗃️  Genres: Productivity

3. Notability: Smarter AI Notes
   📥 Ratings: 389,941
   🗃️  Genres: Productivity, Education

4. Goodnotes 6
   📥 Ratings: 308,094
   🗃️  Genres: Productivity, Education

5. Otter Transcribe Voice Notes
   📥 Ratings: 41,854
   🗃️  Genres: Productivity, Business
```

Gathering User comment Data

Now I am working out how to go through and automate the process of gathering user comments for a given app, then I will clean that data and move onto the sentiment analysis part of this project.