Swift Analysis

Brandwatch data

Data collection began by instructing Brandwatch to search social media content posted from Feb. 1 through June 1, 2024, using the following query:

```
"Taylor Swift" AND ("Tortured Poets" OR "#TSTTPD" OR "#TTPD" OR
"#TTPDTheAnthology") NOT "RT @"
```

The query, which automatically switched to a 44.92% sampling rate due to the sheer volume of the matching content, yielded about 41.7 million posts from accounts identified by Brandwatch as “individual” accounts and another approximately 8.6 million posts from accounts identified by Brandwatch as “organizational” accounts.

Plotted by day, in units of 1,000 posts, both individual and organizational volume showed a distinct peak during the latter half of April, a period corresponding to the “Tortured Poets” album release.

```r
# Required packages
if (!require("tidyverse"))
  install.packages("tidyverse")
if (!require("gmodels"))
  install.packages("gmodels")
if (!require("gtsummary"))
  install.packages("gtsummary")

library(tidyverse)
library(gmodels)
library(ggplot2)
library(gtsummary)
library(readxl)

# Get Brandwatch daily volume summary data
#
https://github.com/drkblake/Data/raw/main/Brandwatch%20TTPD%20mention%20volum
e%20in%20thousands%20Feb%20to%20Jun.xlsx

download.file(

"https://github.com/drkblake/Data/raw/main/Brandwatch%20TTPD%20mention%20volu
me%20in%20thousands%20Feb%20to%20Jun.xlsx",
  "FebToJune.xlsx",
  quiet = TRUE,
  mode = "wb"
)

FebToJune <- read_xlsx("FebToJune.xlsx", sheet = "Long")
```

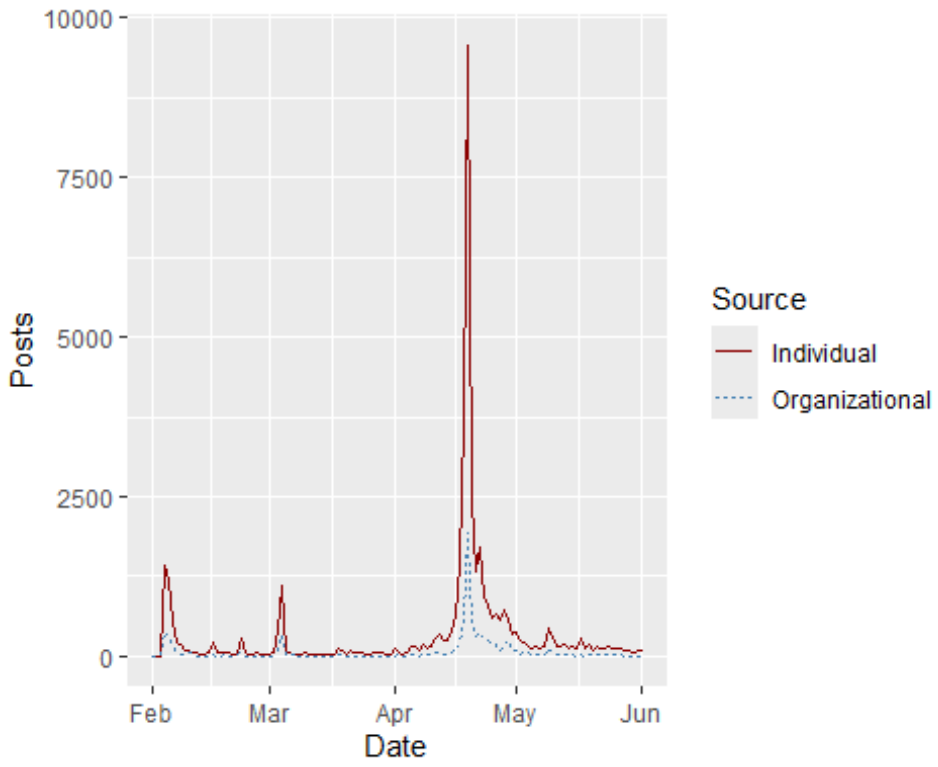
```r
FebToJune$days <- as.POSIXct(FebToJune$days, tz = "America/Chicago")
sum(FebToJune$individual)*1000

## [1] 41686000

sum(FebToJune$organisational)*1000

## [1] 8625000

# Reformatting
FebToJune2 <- FebToJune %>%
  rename(Date = days,
         Individual = individual,
         Organizational = organisational) %>%
  select(Date, Individual, Organizational) %>%
  gather(key = "Source", value = "Posts", -Date)
# Visualization
ggplot(FebToJune2, aes(x = Date, y = Posts)) +
  geom_line(aes(color = Source, linetype = Source)) +
  scale_color_manual(values = c("darkred", "steelblue"))
```



Considering the obvious April peak and the limitations of the full period's sampling rate of well under half of the relevant posts, the analysis focused on posts shared between April 14 and May 1, 2024. The album released on April 19, 2024.

Repeating the above Brandwatch query for posts shared between April 14 and May 1, 2024, yielded 168,828 posts. Because of the smaller volume, Brandwatch retrieved all available

posts instead of only a sample. The standard Brandwatch export includes 135 variables, many relevant to only one of the monitored platforms. The analysis retained only seven.

```
# Read from local file

SwiftData <- read.csv("SwiftData.csv")

# Getting all column names

ColumnNames <- as.data.frame(colnames(SwiftData))

# Keeping columns of interest after making a copy
# of the full dataset for reference

FullSwiftData <- SwiftData
SwiftData <- select(SwiftData,
                    Date,
                    Url,
                    Domain,
                    Page.Type,
                    Account.Type,
                    Author,
                    Full.Text)

# Removing any duplicate rows

SwiftData <- SwiftData %>%
  distinct(Url, .keep_all = TRUE)

#Formatting "Date" as POSIXct object

SwiftData$Date <- as.POSIXct(SwiftData$Date, tz = "America/Chicago")

#Sorting by Date

SwiftData <- arrange(SwiftData,Date)

glimpse(SwiftData)

## Rows: 168,828
## Columns: 7
## $ Date         <dttm> 2024-04-14 05:00:00, 2024-04-14 05:00:00, 2024-04-14
05:…
## $ Url          <chr> "https://hiindia.com/taylor-swift-travis-kelce-kiss-
and-d…
## $ Domain       <chr> "hiindia.com", "arabherald.com",
"calgarymonitor.com", "p…
## $ Page.Type    <chr> "news", "news", "news", "news", "news", "news",
```

```
"news", "…
## $ Account.Type <chr> "", "", "", "", "", "", "", "", "", "", "", "", "",
"", "…
## $ Author       <chr> "Indi A", "", "Voice of America14 Apr 2024, 21:35
GMT+10"…
## $ Full.Text    <chr> "Los Angeles, April 14 (IANS) Singer-songwriter
Taylor Sw…
```

Categorizing posts by type of source

The Brandwatch query retrieved posts from nine different platforms, including “news” platforms, a category consisting of web sites for online news operations. Analysis found 60,749 posts from such “news” platforms. A recode added to the “news” category 5,680 X/Twitter posts from sources marked “organisational” by X/Twitter. All other posts, including all other posts from X/Twitter, were classified as posts from “individual” sources.

```
#Categorizing by source type

PageTable <- SwiftData %>%
  select(Page.Type) %>%
  tbl_summary()
PageTable
```

Characteristic	N = 168,828[1]
Page.Type	
blog	795 (0.5%)
facebook_public	6,499 (3.8%)
forum	1,474 (0.9%)
news	60,749 (36%)
reddit	33,973 (20%)
review	13 (<0.1%)
tumblr	28,385 (17%)
twitter	36,450 (22%)
youtube	490 (0.3%)

[1]n (%)

```
SwiftData <- SwiftData %>%
  mutate(SourceType =
           case_when(
             Page.Type == "news" ~ "News",
             Page.Type == "twitter" &
               Account.Type == "organisational" ~ "News",
             TRUE ~ "Individual"))

SourceTable <- SwiftData %>%
  select(SourceType) %>%
  tbl_summary()
SourceTable
```

Characteristic	N = 168,828[1]
SourceType	
Individual	102,399 (61%)
News	66,429 (39%)

[1]n (%)

Chi-squared test of source type proportions

Obviously, the proportion of individual posts greatly exceeded the proportion of news posts. But was the difference statistically significant? A Chi-squared test can give an answer:

```
SourceType_test <- chisq.test(x = table(SwiftData$SourceType))
SourceType_test

##
##  Chi-squared test for given probabilities
##
## data:  table(SwiftData$SourceType)
## X-squared = 7663.7, df = 1, p-value < 2.2e-16
```

Yes. The proportion of individual posts significantly exceeded the proportion of news posts, $X^2(1, N = 168{,}828) = 7{,}663.7$, $p < .0001$.

Daily counts, by source type

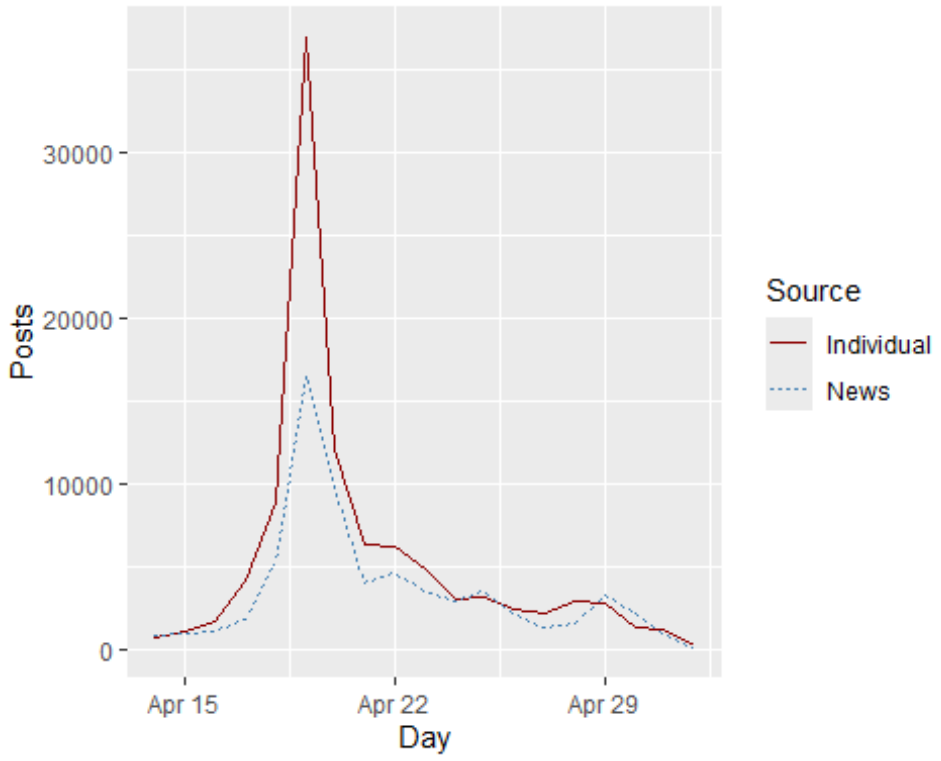
With a few exceptions, daily counts of individual posts outstripped daily counts of news posts:

```
# Re-expressing "Date" as "Day," the day from "Date."

SwiftData <- SwiftData %>%
  mutate(Day = floor_date(Date,
                          unit = "day"))
# Graphing posts per day

PostsByDay <- SwiftData %>%
  group_by(Day, SourceType) %>%
  summarize(
    Posts = n()) %>%
  rename(Source = SourceType)

ggplot(PostsByDay, aes(x = Day, y = Posts)) +
  geom_line(aes(color = Source, linetype = Source)) +
  scale_color_manual(values = c("darkred", "steelblue"))
```




```
rm(PostsByDay)
```

Hunting Easter eggs

The analysis attempted to identify Easter-egg-themed posts by searching for key words and phrases associated with each of nine frequently-alluded-to Easter eggs. Here are the search terms for each, followed by counts and percentages for the number of posts treated as associated with each egg.

```
# Easter egg variables

# Check for "thanK you aIMee" egg

SwiftData <- SwiftData %>%
  mutate(
    aIMee = case_when(
      grepl("thanK you aIMee", Full.Text) ~ "aIMee egg",
      grepl("Kardashian", Full.Text) ~ "aIMee egg",
      grepl("diss track", Full.Text) ~ "aIMee egg",
      grepl("bully", Full.Text) ~ "aIMee egg",
      grepl("lessons learned", Full.Text) ~ "aIMee egg",
      grepl(" Kim ", Full.Text) ~ "aIMee egg",
      TRUE ~ "Other topic"
    )
  )
```

```r
# Check for "So High School" egg

SwiftData <- SwiftData %>%
  mutate(
    HighSchool = case_when(
      grepl("So High School", Full.Text) ~ "High School egg",
      grepl("Kelce", Full.Text) ~ "High School egg",
      grepl("Marry Kiss or Kill me", Full.Text) ~ "High School egg",
      grepl("how to ball", Full.Text) ~ "High School egg",
      grepl("I know Aristotle", Full.Text) ~ "High School egg",
      grepl("love track", Full.Text) ~ "High School egg",
      TRUE ~ "Other topic"
    )
  )


# Check for "So Long, London" egg

SwiftData <- SwiftData %>%
  mutate(
    London = case_when(
      grepl("So Long, London", Full.Text) ~ "London egg",
      grepl("diss track", Full.Text) ~ "London egg",
      grepl("heartbreak song", Full.Text) ~ "London egg",
      grepl("Joe Alwyn", Full.Text) ~ "London egg",
      grepl("London Boy", Full.Text) ~ "London egg",
      grepl("Had a good run", Full.Text) ~ "London egg",
      grepl("demise of relationship", Full.Text) ~ "London egg",
      grepl("Every day of a love affair", Full.Text) ~ "London egg",
      TRUE ~ "Other topic"
    )
  )

# Check for "The Black Dog" egg

SwiftData <- SwiftData %>%
  mutate(
    BlackDog = case_when(
      grepl("The Black Dog", Full.Text) ~ "Black Dog egg",
      grepl(" pub ", Full.Text) ~ "Black Dog egg",
      grepl(" ex ", Full.Text) ~ "Black Dog egg",
      grepl(" bar ", Full.Text) ~ "Black Dog egg",
      grepl("forgetting his location", Full.Text) ~ "Black Dog egg",
      TRUE ~ "Other topic"
    )
  )


# Check for "The Alchemy" egg
```

```r
SwiftData <- SwiftData %>%
  mutate(
    Alchemy = case_when(
      grepl("The Alchemy", Full.Text) ~ "Alchemy egg",
      grepl("Kelce", Full.Text) ~ "Alchemy egg",
      grepl("Travis", Full.Text) ~ "Alchemy egg",
      grepl(" Chiefs ", Full.Text) ~ "Alchemy egg",
      grepl("love song", Full.Text) ~ "Alchemy egg",
      grepl("Where's the trophy", Full.Text) ~ "Alchemy egg",
      grepl("he comes running over to me", Full.Text) ~ "Alchemy egg",
      grepl(
        "So when I touch down, call the amateurs and cut em from the team",
        Full.Text
      ) ~ "Alchemy egg",
      TRUE ~ "Other topic"
    )
  )

# Check for "Cassandra" egg

SwiftData <- SwiftData %>%
  mutate(
    Cassandra = case_when(
      grepl("Cassandra", Full.Text) ~ "Cassandra egg",
      grepl("KimYe", Full.Text) ~ "Cassandra egg",
      grepl("Greek Mythology", Full.Text) ~ "Cassandra egg",
      grepl("snake imagery", Full.Text) ~ "Cassandra egg",
      grepl("doomed to see the future", Full.Text) ~ "Cassandra egg",
      grepl("no one believes her", Full.Text) ~ "Cassandra egg",
      grepl(" greed ", Full.Text) ~ "Cassandra egg",
      grepl("family", Full.Text) ~ "Cassandra egg",
      grepl("Kardashian-West diss track", Full.Text) ~ "Cassandra egg",
      TRUE ~ "Other topic"
    )
  )

# Check for "I Hate it Here" egg

SwiftData <- SwiftData %>%
  mutate(
    HateitHere = case_when(
      grepl("I Hate it Here", Full.Text) ~ "Hate it Here egg",
      grepl("Alwyn", Full.Text) ~ "Hate it Here egg",
      grepl(" ex ", Full.Text) ~ "Hate it Here egg",
      grepl("relationship was painful", Full.Text) ~ "Hate it Here egg",
      grepl("hiding in her relationship", Full.Text) ~ "Hate it Here egg",
      TRUE ~ "Other topic"
    )
```

```
  )


# Check for "2am Release of second album" egg

SwiftData <- SwiftData %>%
  mutate(
    SecondAlbum = case_when(
      grepl("2am Release of second album", Full.Text) ~ "Second Album Time
Release egg",
      grepl("Clock in Midnights room was 2am", Full.Text) ~ "Second Album
Time Release egg",
      grepl("2 fingers during album annoucement", Full.Text) ~ "Second Album
Time Release egg",
      grepl("past mentions of 2am in past songs", Full.Text) ~ "Second Album
Time Release egg",
      grepl("pocket watch in Bejewled video", Full.Text) ~ "Second Album Time
Release egg",
      grepl("double album", Full.Text) ~ "Second Album Time Release egg",
      grepl("31 songs", Full.Text) ~ "Second Album Time Release egg",
      TRUE ~ "Other topic"
    )
  )


# Check for "Album Release Date" egg

SwiftData <- SwiftData %>%
  mutate(
    ReleaseDate = case_when(
      grepl("Album Release Date", Full.Text) ~ "Release Date egg",
      grepl("famous dinner", Full.Text) ~ "Release Date egg",
      grepl("Blake Lively", Full.Text) ~ "Release Date egg",
      grepl("Ryan Reynolds", Full.Text) ~ "Release Date egg",
      grepl("announced breakup", Full.Text) ~ "Release Date egg",
      grepl("April 19th", Full.Text) ~ "Release Date egg",
      grepl("all unfollowed Alwyn", Full.Text) ~ "Release Date egg",
      TRUE ~ "Other topic"
    )
  )

# Check for generic "Easter egg" mention

SwiftData <- SwiftData %>%
  mutate(
    GenericEgg = case_when(
      grepl("Easter egg", Full.Text) ~ "Generic egg",
      grepl("Easter eggs", Full.Text) ~ "Generic egg",
      TRUE ~ "Other topic"
```

```
    )
  )

# How many of each egg?

EggTable <- SwiftData %>%
  select(
    aIMee,
    HighSchool,
    London,
    BlackDog,
    Alchemy,
    Cassandra,
    HateitHere,
    SecondAlbum,
    ReleaseDate,
    GenericEgg,

  ) %>%
  tbl_summary()
EggTable
```

Characteristic	N = 168,828[1]
aIMee	
aIMee egg	4,221 (2.5%)
Other topic	164,607 (97%)
HighSchool	
High School egg	5,919 (3.5%)
Other topic	162,909 (96%)
London	
London egg	5,617 (3.3%)
Other topic	163,211 (97%)
BlackDog	
Black Dog egg	3,017 (1.8%)
Other topic	165,811 (98%)
Alchemy	
Alchemy egg	6,887 (4.1%)
Other topic	161,941 (96%)
Cassandra	
Cassandra egg	1,548 (0.9%)
Other topic	167,280 (99%)
HateitHere	
Hate it Here egg	4,149 (2.5%)
Other topic	164,679 (98%)
SecondAlbum	
Other topic	162,461 (96%)
Second Album Time Release egg	6,367 (3.8%)
ReleaseDate	
Other topic	168,287 (100%)
Release Date egg	541 (0.3%)
GenericEgg	
Generic egg	778 (0.5%)
Other topic	168,050 (100%)

[1]n (%)

Easter egg counts by source type

The analysis explored whether the number of posts qualifying as Easter egg posts varied significantly by source type. First, some code to categorize each post as alluding to at least one of the specified eggs or as alluding to none of the specified eggs:

```
# Quant transformation of Easter egg variables

SwiftData <- SwiftData %>%
```

```r
  mutate(aIMee = case_when(
    grepl("thanK you aIMee", Full.Text) ~ 1,
    grepl("Kardashian", Full.Text) ~ 1,
    grepl("diss track", Full.Text) ~ 1,
    grepl("bully", Full.Text) ~ 1,
    grepl("lessons learned", Full.Text) ~ 1,
    grepl(" Kim ", Full.Text) ~ 1,
    TRUE ~ 0
  ))

# Check for "So High School" egg

SwiftData <- SwiftData %>%
  mutate(HighSchool = case_when(
    grepl("So High School", Full.Text) ~ 1,
    grepl("Kelce", Full.Text) ~ 1,
    grepl("Marry Kiss or Kill me", Full.Text) ~ 1,
    grepl("how to ball", Full.Text) ~ 1,
    grepl("I know Aristotle", Full.Text) ~ 1,
    grepl("love track", Full.Text) ~ 1,
    TRUE ~ 0
  ))


# Check for "So Long, London" egg

SwiftData <- SwiftData %>%
  mutate(
    London = case_when(
      grepl("So Long, London", Full.Text) ~ 1,
      grepl("diss track", Full.Text) ~ 1,
      grepl("heartbreak song", Full.Text) ~ 1,
      grepl("Joe Alwyn", Full.Text) ~ 1,
      grepl("London Boy", Full.Text) ~ 1,
      grepl("Had a good run", Full.Text) ~ 1,
      grepl("demise of relationship", Full.Text) ~ 1,
      grepl("Every day of a love affair", Full.Text) ~ 1,
      TRUE ~ 0
    )
  )

# Check for "The Black Dog" egg

SwiftData <- SwiftData %>%
  mutate(BlackDog = case_when(
    grepl("The Black Dog", Full.Text) ~ 1,
    grepl(" pub ", Full.Text) ~ 1,
    grepl(" ex ", Full.Text) ~ 1,
    grepl(" bar ", Full.Text) ~ 1,
```

```
    grepl("forgetting his location", Full.Text) ~ 1,
    TRUE ~ 0
  ))


# Check for "The Alchemy" egg

SwiftData <- SwiftData %>%
  mutate(
    Alchemy = case_when(
      grepl("The Alchemy", Full.Text) ~ 1,
      grepl("Kelce", Full.Text) ~ 1,
      grepl("Travis", Full.Text) ~ 1,
      grepl(" Chiefs ", Full.Text) ~ 1,
      grepl("love song", Full.Text) ~ 1,
      grepl("Where's the trophy", Full.Text) ~ 1,
      grepl("he comes running over to me", Full.Text) ~ 1,
      grepl(
        "So when I touch down, call the amateurs and cut em from the team",
        Full.Text
      ) ~ 1,
      TRUE ~ 0
    )
  )

# Check for "Cassandra" egg

SwiftData <- SwiftData %>%
  mutate(
    Cassandra = case_when(
      grepl("Cassandra", Full.Text) ~ 1,
      grepl("KimYe", Full.Text) ~ 1,
      grepl("Greek Mythology", Full.Text) ~ 1,
      grepl("snake imagery", Full.Text) ~ 1,
      grepl("doomed to see the future", Full.Text) ~ 1,
      grepl("no one believes her", Full.Text) ~ 1,
      grepl(" greed ", Full.Text) ~ 1,
      grepl("family", Full.Text) ~ 1,
      grepl("Kardashian-West diss track", Full.Text) ~ 1,
      TRUE ~ 0
    )
  )

# Check for "I Hate it Here" egg

SwiftData <- SwiftData %>%
  mutate(HateitHere = case_when(
    grepl("I Hate it Here", Full.Text) ~ 1,
    grepl("Alwyn", Full.Text) ~ 1,
```

```
    grepl(" ex ", Full.Text) ~ 1,
    grepl("relationship was painful", Full.Text) ~ 1,
    grepl("hiding in her relationship", Full.Text) ~ 1,
    TRUE ~ 0
  ))


# Check for "2am Release of second album" egg

SwiftData <- SwiftData %>%
  mutate(
    SecondAlbum = case_when(
      grepl("2am Release of second album", Full.Text) ~ 1,
      grepl("Clock in Midnights room was 2am", Full.Text) ~ 1,
      grepl("2 fingers during album annoucement", Full.Text) ~ 1,
      grepl("past mentions of 2am in past songs", Full.Text) ~ 1,
      grepl("pocket watch in Bejewled video", Full.Text) ~ 1,
      grepl("double album", Full.Text) ~ 1,
      grepl("31 songs", Full.Text) ~ 1,
      TRUE ~ 0))

# Check for "Album Release Date" egg

SwiftData <- SwiftData %>%
  mutate(
    ReleaseDate = case_when(
      grepl("Album Release Date", Full.Text) ~ 1,
      grepl("famous dinner", Full.Text) ~ 1,
      grepl("Blake Lively", Full.Text) ~ 1,
      grepl("Ryan Reynolds", Full.Text) ~ 1,
      grepl("annouced breakup", Full.Text) ~ 1,
      grepl("April 19th", Full.Text) ~ 1,
      grepl("all unfollowed Alwyn", Full.Text) ~ 1,
      TRUE ~ 0
    )
  )

# Check for generic "Easter egg" mention

SwiftData <- SwiftData %>%
  mutate(
    GenericEgg = case_when(
      grepl("Easter egg", Full.Text) ~ 1,
      grepl("Easter eggs", Full.Text) ~ 1,
      TRUE ~ 0
    )
  )
```

```r
# Exploring media vs. individuals

SwiftData <- SwiftData %>%
  mutate(
    EasterEggs = aIMee +
      HighSchool +
      London +
      BlackDog +
      Alchemy +
      Cassandra +
      HateitHere +
      SecondAlbum +
      ReleaseDate +
      GenericEgg
  ) %>%
  mutate(AnyEgg = case_when(EasterEggs > 0 ~ "Egg", TRUE ~ "No egg"))
```

Next, a crosstabulation table, which shows that 24% of the posts from news sources alluded to an Easter egg, compared to only 8.2% of the posts from individuals:

```r
Crosstab <- SwiftData %>%
  tbl_cross(row = AnyEgg,
            col = SourceType,
            percent = "column")
Crosstab
```

	SourceType		
	Individual	News	Total
AnyEgg			
Egg	8,662 (8.5%)	16,163 (24%)	24,825 (15%)
No egg	93,737 (92%)	50,266 (76%)	144,003 (85%)
Total	102,399 (100%)	66,429 (100%)	168,828 (100%)

A chi-squared test found the pattern statistically significant:

```r
# Specify the DV and IV
SwiftData$DV <- SwiftData$AnyEgg #Edit YOURDVNAME
SwiftData$IV <- SwiftData$SourceType #Edit YOURIVNAME

# Run the chi-squared test
options(scipen = 999)
chitestresults <- chisq.test(SwiftData$DV, SwiftData$IV)
chitestresults

##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  SwiftData$DV and SwiftData$IV
## X-squared = 8091.7, df = 1, p-value < 0.00000000000000022
```

Easter egg counts by type of fan

It's challenging to identify, through quantitative content analysis, Taylor Swift fans who are likely to have developed parasocial relationship with Swift. The approach used here began by using Brandwatch to download all replies to the three posts Swift shared on Twitter during the late-April 2024 time period being examined. Swift's three posts were:

- https://x.com/taylorswift13/status/1779486593587315020, in which Swift promoted an exclusive vinyl version of the album from Target.
- https://x.com/taylorswift13/status/1782803343858860248, in which Swift shared a link to a Billboard.com review of the album.
- https://x.com/taylorswift13/status/1783220925480997314, in which Swift shared a link of an NPR.org review of the album.

The approach assumed that individuals in a parasocial relationship with Swift would have replied to at least one of these three posts by Swift. It did not assume, however, that all replies to the three posts came from such fans. Even a cursory look through the posts reveals many that are critical of Swift, aimed at promoting unrelated ideas, etc.

This code retrieves a file of Brandwatch data containing the text of each reply, the reply author's user name, and more. It also writes a copy of the file to the user's hard drive.

```
# Get data on replies to three Swift posts

Fans <- read.csv("https://github.com/drkblake/Data/raw/main/Fans.csv")

write.csv(Fans,"Fans.csv", row.names = FALSE)
```

The "NPR Review" post drew the most replies, followed by the "Billboard review" post, and then the "Target promo" post:

```
# Reply counts by post

ReplyTable <- Fans %>%
  select(Post) %>%
  tbl_summary()

ReplyTable
```

Characteristic	N = 3,045[1]
Post	
Billboard review	1,333 (44%)
NPR review	1,479 (49%)
Target promo	233 (7.7%)

[1]n (%)

The approach assumed that Twitter users who had replied to at least one of Swift's posts and also had shared a post captured by the Brandwatch search for posts about the album would be at least somewhat likely to have formed a parasocial relationship with Swift.

In keeping with this assumption, a list of each user who had posted at least one reply to at least one of Swift's posts was extracted, then merged with the full dataset, so that each post in the full dataset could be identified has having come from a potential "PSRFan," who had replied to at least one of Swift's posts, or some "Other" type of user.

This code accomplishes the merge, then produces a crosstabulation showing the number of Easter egg posts among "PSR Fans" and "Others." It also saves a copy of the merged file, SwiftData2.csv, to the user's computer.

```
# Extract list of Twitter IDs from Fans data

FanList <- Fans %>%
  group_by(Author) %>%
  summarize(PSRPosts = n())

# Merge list of Fan IDs with larger dataset
# Flag each post as from a "PSRFan" or "Other"

SwiftData2 <- merge(SwiftData,FanList,
                    by="Author",
                    all.x = TRUE)
SwiftData2 <- SwiftData2 %>%
  mutate(PSRPosts = coalesce(PSRPosts,0)) %>%
  mutate(PSRFan = case_when(PSRPosts > 0 ~ "PSRFan",
                            TRUE ~ "Other"))
write.csv(SwiftData2, "SwiftData2.csv",
          row.names = FALSE)

# Get count of PSR fans

if (!require("gtExtras"))
  install.packages("gtExtras")

## Loading required package: gtExtras

## Loading required package: gt
```

```r
library(gtExtras)

PSRFanPosters <- SwiftData2 %>%
  filter(PSRFan == "PSRFan") %>%
  group_by(Author) %>%
  summarize(Posts = n()) %>%
  arrange(desc(Posts))

nrow(PSRFanPosters)

## [1] 559

gt(PSRFanPosters) %>%
  cols_align(align = "center") %>%
  gt_theme_538
```

SwiftNYC	37
kenziesversion	24
happinessforts	22
imhharry	21
ZanaHajiSaid	20
ToniMarieKeys	18
reputanush13	18
swiftarmy1989	18
thisisabbytryin	18
dhart0412	17
myhappinesstayy	16
wakinginthedark	16
RaghavsRep	15
getawaycargirlz	15
learn_with_san	15
alltooriah	14
emsswiftieadv	13
folksyswift	13
kaleighsversion	13
kloetheswiftie	13
INA_YH95	12
corneliastagain	12
shookswiftie	12
stilllgotscars	12
PatriotPointman	11
fairyytaleswift	11
icouldstaylor	11
kubaqzn	11
Fityeth	10
alltoohailee	10
wkswift13	10
ExileftNini	9
OmgItsMeAllison	9
soscarletsar	9
swiftie1ncanada	9
swiftsvrse	9
holygroundsound	8
tswizzle89girl	8
AllTooSammi13	7
bigelam	7
bisScottTea	7
cowboylikediane	7

inafolklore	7
kush07_	7
mirrorswifft	7
swifferwins	7
CorneliaDynasty	6
CourtneySong_	6
aurorabora518	6
becky_maroon	6
bejeweltay	6
ciwywisteria	6
ddenver13	6
deathbytaylor4	6
exiledarren	6
hiss_swift_hiss	6
just__meraki	6
karmaisamack	6
lucia_swiftiee	6
remisversion	6
EmilyG234	5
KarmaIsAFad	5
MichaelMou61314	5
SwiftCriminal13	5
crumblingcait13	5
difficultswift	5
doepikapadukone	5
fatihahere	5
folkIegends	5
iamhaunted13	5
intheweeeds	5
lysstaysversion	5
metsswift	5
musikistlife	5
nothlngsnew	5
repftv	5
romaricharlz13	5
solongldn	5
sweeetener	5
the13manuscript	5
theopinionline	5
tizthedamnszn_	5
vallovestswift	5
zaralovestaylor	5

AlexStarlight13	4
Dakotasversion	4
Mr_Swift1594	4
SwiftieLR	4
__theantihero__	4
annasflicker	4
jasonlipshutz	4
jessica_parks	4
karmaisagad	4
mirrorballaly	4
mirrorballydia	4
sleeepyeyes	4
sabsevermore	4
safeandsoundhan	4
skysmirrorball	4
sobrittgoes	4
swiftiealyza	4
thisiscetrying	4
tis_sjoberg	4
1mogenn	3
4nd123w_	3
CL_Hayate	3
NiamhlovesTay13	3
Rebecca_CW13	3
SwiftSocial1989	3
TS19_89	3
That_July_9	3
TisTheDamnPhD	3
al3xaplaytswift	3
anarchistswift	3
benstafford100	3
brarda_emilia	3
chloeg_13	3
cjaemd	3
cmaikerujei	3
eggscrams	3
everslay13	3
f0lkl0r313	3
fearlesslygrace	3
folkhwore	3
gleam_twinkle	3
graynovember13	3

imyoelvy	3
inezbythelakes	3
ivygrows1213	3
japrilss	3
lexofmyheart	3
liekeisootw	3
littletookind	3
lovestephy	3
mirrorballkate	3
ninaswiftie13	3
ninetiestrend	3
oncornelias	3
polkadotrryy	3
porcocteles	3
savyouaseat	3
serge_golubev	3
simplyyswiftie	3
sippingaugust	3
sweetnothingnew	3
teacherswiftie	3
thegreatwar13	3
tooth4taylor	3
vinyuh	3
weallgotcrowns7	3
1lactiflora9	2
9tay8tay9_	2
Abbylylee	2
Alyssa_TV13	2
BTaylorVersion	2
Bestoptionisme	2
COLLAGECHARTS	2
Cikyugindayo	2
EurovisionSwift	2
GauchoTeddy	2
GetawayCaroline	2
HannahSapp505	2
HollyIvyDaisy13	2
JStandridge_	2
JomirBrands	2
JordynLWYMMD	2
KeyFactoryNYC	2
MariaLoyoC	2

MikaSwift13	2
MothestMo	2
OrdinaryfoolNJ	2
RepGirlTV1989	2
Rickymidnights	2
RicoMorcielo	2
RockinBlondieTS	2
SelsGraceful	2
Swiftie092	2
Swiftlyness1989	2
Swifts__Version	2
Swiftsloverx	2
Tcheckx	2
UrbanVicki9324	2
VaniaRobles24	2
VinSwift13	2
WhiTaysVersion	2
afroallura	2
alexsscardigan	2
ashleyegreer	2
autisticswiftie	2
autumnhoaxes	2
beasreputation	2
bejeweledselena	2
beyonceBGKC	2
bloom_bloom___	2
bwswiftcats	2
castlscrumblng7	2
champgns	2
chellabey	2
dani_readyforit	2
diancuh	2
elainereadz	2
ellebellelvr	2
everaishamore	2
exiledpoetsdept	2
faithless_hoax	2
flkloreee	2
floridaamii	2
footnqtes	2
gagasyuyi	2
grcelivia	2

happilylauren	2
hearthandsswift	2
henryjekylls	2
hotasswiftin	2
i_poetrylove	2
iammrswift	2
illicitayfairs	2
ionafyfe	2
ioveyourparty	2
isaactheswiftie	2
jjsversion	2
jordynsversion	2
kaiamal13	2
kiwik94	2
lauralynn1955	2
lavenderhanz	2
lavenderrain31	2
lindsheartstayx	2
lizzyyy1989	2
londonbcy	2
longlivereptour	2
longlivethtlook	2
maamstfu	2
mattydedios	2
meawromantics	2
missbraziliana	2
ninfetinhacis	2
notcoolbanana	2
oldcardlgan	2
obsesedlily	2
pinkpurp11	2
polaroidofus_tv	2
rachutation	2
repuhtayytion	2
roses_are_tay13	2
santaclaraswift	2
saydontgos	2
scrofanoswift	2
shinythingswift	2
sirentheswiftie	2
smollerpauly	2
sparklysari	2

spell1ngisfunn	2
stillruinmylife	2
swift_af_23	2
swiftharry2	2
swiftie_livie13	2
swiftiedayas	2
swifttvs	2
taylorsivygrows	2
theeladybird13	2
thisisluutrying	2
tinaxxwash	2
tswiftsttpd	2
waoyfswift	2
writsbyme	2
yailpersonified	2
1989swiftafboi	1
AllinderLi81282	1
B1azeYouStart3d	1
BernThisWey	1
CCConstantina	1
Carlos05443850	1
CodyCaston13	1
Crazyfangirl1Jo	1
D88673933D	1
DanaBrigoli	1
Dont_Blame_Bri	1
EVERG0D	1
EmilyyyHale	1
Enchantedivy2	1
EpiphanyStan725	1
EternalJolie_	1
F3MININOMENON	1
FORNlGHT	1
GabrielleL11535	1
GoodGuyGingy	1
HolyJosee	1
IceWolf_17	1
Ilariaisreading	1
ImChaos_Revelry	1
Iondonboylouis	1
Jen4tro	1
KarmaIsAQuinn	1

KevinThebar	1
LONGLIVE_	1
LauraEden13	1
LilyMeade	1
LongLiveBee13	1
MBSanAndreas13	1
Mads_IsTrying13	1
MarkyMark_Jr81	1
MidnightsMads_	1
MindofRollins	1
MotherSwiftly	1
OrangesAndJess	1
PurrfectlyTS	1
RhodeToLove	1
Rithikisabitch	1
RrChaaaa	1
SWIFTSGETAWAAY	1
ScottishLovatic	1
Sertralineque3n	1
Sir6ALotB	1
Stanlyapp	1
StreamOver99	1
Streamqueen486	1
TTPDBolter	1
Taylorsswift_22	1
ThatFanGirl1194	1
TheGoatfox	1
VictorVfoc	1
WaiYanM76350781	1
Whit5Eve	1
_Exile_Cardigan	1
Jacob89	1
_ashleeeyc	1
_bariumsulfide	1
_finallycleann	1
ablikefolksongs	1
adisynev	1
aleftespos	1
alenasfolksongs	1
ali_too_well	1
alossstar	1
alyssitaffairs	1

amigorzito	1
aotydored	1
aswiftieaccount	1
augustxaudrey	1
averysavy	1
baldshelby	1
bbejeweledd	1
bbyiknowplaces	1
beginagqin	1
bigrepstan	1
bloodmoonlit97	1
braxtonmg	1
breebrandnew	1
btweentheliines	1
cantdolife2	1
castlescrmblng	1
cattitude_girl	1
chiaswizzle	1
chinitoprablems	1
clarabowswift	1
cmtswift	1
comebackbehere	1
comet5inthe5ky	1
cosyswift	1
cowboylikejuli	1
cowboy_like_ava	1
cowboylikegabs	1
cowboylikehale	1
cowboylikejac0b	1
cr3lintentions	1
crvstify	1
daenerfs	1
dancedintherain	1
dani_kindiskey	1
dayasbrina	1
daylight_poetry	1
daylighttwtz	1
dontwannashar3	1
dreamtimeswift	1
dudaswift13_	1
dwohtdeluxe	1
ejosefineee13	1

electriqforit	1
enchantedstan89	1
endlessfebruary	1
envythewoods	1
erastour0324	1
erastourhoe	1
evegotbitten	1
everspeak13	1
evremoree	1
f1replace_ash3s	1
fangirlburner	1
farmerswiftie	1
fearlvess	1
finelinelouiswt	1
fireflymystery	1
fireplaceeashes	1
folklaura_	1
folklore_nfr	1
folkloreemind	1
folklorestarss	1
folkmoonlore	1
folktayles	1
foreverwiiinter	1
foreverwinterbb	1
forksmycity	1
fumerossantiago	1
gabssfm08	1
gabyeskye13	1
georgiastarss	1
getawaykate13	1
gilmour34	1
gra6cee	1
harrypurplebow	1
hauntedbymyrep	1
hauntedstaylor	1
hauntedua	1
hbwannie	1
heydorothea	1
hitsdiffrent13	1
homesicklauren	1
hrts4dorothea	1
huspsa	1

ilovedylandchlo	1
imamirrorbll	1
imgonnagetubacc	1
inmydreamlanddd	1
inshadesofwrong	1
invisibleasu	1
itxsev	1
itzsdc	1
ivyromance	1
izzykitty27	1
jacketisyours	1
janexxxmw	1
janineteaques	1
jeannielangz	1
jo0rgesandoval_	1
jrglennon	1
juliewrestling	1
kaearnah	1
karmaismybtrfly	1
kendramunozz	1
krisofmyheart	1
lacyafterglow	1
larkofthemisery	1
lavenderxwonder	1
lavgays13	1
leesiebridgers	1
lexis8685	1
liberTAYrian	1
lilly30_	1
llamarswift13	1
lostinamericaa	1
loversloml	1
lyingtraitors	1
magick_bones	1
mariana_pereira	1
marisaswiftie	1
maroontv13	1
meghanosleyy_	1
meiaheartstay	1
midnightsashley	1
milasagna	1
moeralu	1

moonberryswift	1
moontosaturns	1
nansversion	1
naylortation13	1
newmaserati89	1
niallsshining	1
nintencel	1
nobootaynocrime	1
noellepolo	1
nottayloras	1
nrendipity	1
oflittleoldme13	1
ohmyreputaytion	1
onastarlitnight	1
ovrdrmaticntrue	1
painTStheEras	1
pikachooseyou	1
pinkspeaknow	1
plaasticheartss	1
plainrocktweets	1
plasticoffdsofa	1
popsisunsteady	1
pqristay	1
prentisslvr	1
raniascardigan	1
ranimisra02	1
replixie	1
repswifttt13	1
reputasentv	1
reputationdave_	1
reqnights	1
reyhannswift13	1
rightwylmm	1
rissarep_nights	1
samgirldean	1
sapphiretearsky	1
savinswift	1
scarlettcrowns	1
secretpluto13	1
shelleysilver3	1
silent_partner_	1
skinnysel	1

softpippa	1
sohighschooll	1
solongdeareader	1
sophialoves1989	1
speaknowdb	1
sportyswiftie89	1
stateofdijon13	1
stillswiffafboi	1
stormiewrites	1
stormsinmyeyez	1
sweetenermadboy	1
swiftclown	1
swiftgetawaytay	1
swiftly_gracie	1
swiftsantihero	1
swiftstrange	1
swiftwithonearm	1
taylorsel	1
tayazula	1
taydailey8	1
taydaloo	1
tayearns	1
taysversiononly	1
thatskye_	1
theboywiths	1
thegreatwrs	1
thehauntedbell	1
thepoetsintern	1
thereinlieserin	1
thislovve	1
timeless_era_	1
tio_victor_	1
tisthedamnpoets	1
tlgadfeatrena	1
tori_fetzer	1
tortured_poettt	1
tsignelin	1
tswiftmeowqueen	1
tswizzle_ontop	1
ttpddolly	1
uncommon_cns	1
velucifer	1

wlwsfilm	1
watchedithappen	1
wednesdaycxfe	1
wendytay606	1
whenwebothfell	1
wisteria1ane	1
wmtswift	1
worshipthislcve	1
yisingkao	1
yousaydontgo	1
yoyokxts	1
zbridgery	1
zionlovingswift	1

```
# Crosstab and chi-square

Crosstab2 <- SwiftData2 %>%
  tbl_cross(row = AnyEgg,
            col = PSRFan,
            percent = "column")
Crosstab2
```

	PSRFan		
	Other	PSRFan	Total
AnyEgg			
Egg	24,772 (15%)	53 (3.4%)	24,825 (15%)
No egg	142,514 (85%)	1,489 (97%)	144,003 (85%)
Total	167,286 (100%)	1,542 (100%)	168,828 (100%)

Qualitative analysis

Quantitative analysis of manifest content can miss nuances that qualitative analysis might reveal. Accordingly, 100-case random samples were extracted from key data frames in the analysis so that the full text of the sampled posts could be examined qualitatively.

```
Fans_sample <- sample_n(Fans,100)

Fan_egg_posts <- SwiftData2 %>%
  filter(PSRFan == "PSRFan" & AnyEgg == "Egg")

Fan_no_egg_posts_sample <- SwiftData2 %>%
  filter(PSRFan == "PSRFan" & AnyEgg == "No egg")
Fan_no_egg_posts_sample <- sample_n(Fan_no_egg_posts_sample,100)

write.csv(Fans_sample,
          "Fans_sample.csv",
```

```
          row.names = FALSE)

write.csv(Fan_egg_posts,
          "Fan_egg_posts.csv",
          row.names = FALSE)

write.csv(Fan_no_egg_posts_sample,
          "Fan_no_egg_posts_sample.csv",
          row.names = FALSE)
```