Lab 3

Conie O'Malley

2025-02-09

Table of contents

```
library(reticulate)
```

```
Warning: package 'reticulate' was built under R version 4.3.3
```

```
library(dplyr)
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
packages <- c("rvest", "tm.plugin.mail", "Rcrawler", "RSelenium", "RedditExtractoR", "bskyr"
for (i in packages) {
    renv::install(i)
}
```

```
The following package(s) will be installed:
- rvest [1.0.4]
These packages will be installed into "~/Library/Caches/org.R-project.R/R/renv/library/AdvTx

# Installing packages --------------------------------------------------------
- Installing rvest ...                          OK [linked from cache]
Successfully installed 1 package in 6 milliseconds.
```

```
The following package(s) will be installed:
- tm.plugin.mail [0.3-1]
These packages will be installed into "~/Library/Caches/org.R-project.R/R/renv/library/AdvTx

# Installing packages --------------------------------------------------------
- Installing tm.plugin.mail ...                 OK [linked from cache]
Successfully installed 1 package in 5 milliseconds.
The following package(s) will be installed:
- Rcrawler [0.1.9-1]
These packages will be installed into "~/Library/Caches/org.R-project.R/R/renv/library/AdvTx

# Installing packages --------------------------------------------------------
- Installing Rcrawler ...                       OK [linked from cache]
Successfully installed 1 package in 5.2 milliseconds.
The following package(s) will be installed:
- RSelenium [1.7.9]
These packages will be installed into "~/Library/Caches/org.R-project.R/R/renv/library/AdvTx

# Installing packages --------------------------------------------------------
- Installing RSelenium ...                      OK [linked from cache]
Successfully installed 1 package in 5.3 milliseconds.
The following package(s) will be installed:
- RedditExtractoR [3.0.9]
These packages will be installed into "~/Library/Caches/org.R-project.R/R/renv/library/AdvTx

# Installing packages --------------------------------------------------------
- Installing RedditExtractoR ...                OK [linked from cache]
Successfully installed 1 package in 5 milliseconds.
The following package(s) will be installed:
- bskyr [0.2.0]
These packages will be installed into "~/Library/Caches/org.R-project.R/R/renv/library/AdvTx

# Installing packages --------------------------------------------------------
- Installing bskyr ...                          OK [linked from cache]
Successfully installed 1 package in 5.1 milliseconds.
The following package(s) will be installed:
- rtoot [0.3.5]
These packages will be installed into "~/Library/Caches/org.R-project.R/R/renv/library/AdvTx

# Installing packages --------------------------------------------------------
- Installing rtoot ...                          OK [linked from cache]
Successfully installed 1 package in 5.3 milliseconds.
```

1 Part 1: Importing Data Using the tm Package

1.1 Deliverable 1: Review File Types

```
tm::getReaders() # obtain available readers from tm package
```

```
 [1] "readDataframe"           "readDOC"
 [3] "readPDF"                 "readPlain"
 [5] "readRCV1"                "readRCV1asPlain"
 [7] "readReut21578XML"        "readReut21578XMLasPlain"
 [9] "readTagged"              "readXML"
```

Word: "readDOC" PDF: "readPDF" Plaintext: "readPlain" HTML: "readHTML" Email: "readMail"

1.2 Deliverable 2: Getting and Setting a Working Directory

```
getwd() # get working directory
```

```
[1] "/Users/coniecakes/Library/CloudStorage/OneDrive-Personal/001. Documents - Main/023. Prog
```

1.3 Deliverable 3: Importing Tab Delimited Files: Prepare Trump Impeachment Object

```
trump1 <- read.delim("/Users/coniecakes/Library/CloudStorage/OneDrive-Personal/001. Documents
str(trump1) # structure of the data frame
```

```
'data.frame':   10987 obs. of  5 variables:
 $ HEARING     : chr  "2019-11-20" "2019-11-20" "2019-11-20" "2019-11-20" ...
 $ SPEAKER     : chr  "Adam Schiff" "Adam Schiff" "Adam Schiff" "Adam Schiff" ...
 $ MAIN.SPEAKER: chr  "D-Schiff" "D-Schiff" "D-Schiff" "D-Schiff" ...
 $ ROLE        : chr  "Democrat" "Democrat" "Democrat" "Democrat" ...
 $ TEXT        : chr  "your interest in being here. In turn, we ask for your respect as we pr
```

```r
trump2 <- read.delim("/Users/coniecakes/Library/CloudStorage/OneDrive-Personal/001. Documents
str(trump2)
```

```
'data.frame':   10987 obs. of  5 variables:
 $ HEARING     : chr  "2019-11-20" "2019-11-20" "2019-11-20" "2019-11-20" ...
 $ SPEAKER     : chr  "Adam Schiff" "Adam Schiff" "Adam Schiff" "Adam Schiff" ...
 $ MAIN.SPEAKER: chr  "D-Schiff" "D-Schiff" "D-Schiff" "D-Schiff" ...
 $ ROLE        : chr  "Democrat" "Democrat" "Democrat" "Democrat" ...
 $ TEXT        : chr  "your interest in being here. In turn, we ask for your respect as we pr
```

```r
trump3 <- readr::read_tsv("/Users/coniecakes/Library/CloudStorage/OneDrive-Personal/001. Docu
```

```
Rows: 10987 Columns: 5
-- Column specification --------------------------------------------------------
Delimiter: "\t"
chr  (4): SPEAKER, MAIN SPEAKER, ROLE, TEXT
date (1): HEARING

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
str(trump3)
```

```
spc_tbl_ [10,987 x 5] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ HEARING     : Date[1:10987], format: "2019-11-20" "2019-11-20" ...
 $ SPEAKER     : chr [1:10987] "Adam Schiff" "Adam Schiff" "Adam Schiff" "Adam Schiff" ...
 $ MAIN SPEAKER: chr [1:10987] "D-Schiff" "D-Schiff" "D-Schiff" "D-Schiff" ...
 $ ROLE        : chr [1:10987] "Democrat" "Democrat" "Democrat" "Democrat" ...
 $ TEXT        : chr [1:10987] "your interest in being here. In turn, we ask for your respect
 - attr(*, "spec")=
  .. cols(
  ..   HEARING = col_date(format = ""),
  ..   SPEAKER = col_character(),
  ..   `MAIN SPEAKER` = col_character(),
  ..   ROLE = col_character(),
  ..   TEXT = col_character()
  .. )
 - attr(*, "problems")=<externalptr>
```

```
class(trump1)
```

```
[1] "data.frame"
```

```
class(trump2)
```

```
[1] "data.frame"
```

```
class(trump3)
```

```
[1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

1.4 Deliverable 4: Import and Inspect a Folder of .txt files Using TM: IGF Bali Transcripts

```
igfbali <- tm::Corpus(tm::DirSource("/Users/coniecakes/Library/CloudStorage/OneDrive-Personal
                      readerControl=list(reader=tm::readPlain))
tm::inspect(igfbali[1]) # inspect igfbali
```

```
<<SimpleCorpus>>
Metadata:  corpus specific: 1, document level (indexed): 0
Content:  documents: 1


6th Meeting of the Dynamic Coalition on Internet and Climate Change \n\n\n\n\nEIGHTH INTERNE
ENHANCING MULTI-STAKEHOLDER COOPERATION FOR GROWTH AND SUSTAINABLE DEVELOPMENT\nOCTOBER 23, 2
```

```
class(igfbali) -> igfbali_class # identify the class of igfbali
```

We use readPlain in order to read the plain text data files from the `txt_data` into a corpus.

`igfbali` is a SimpleCorpus, Corpus object.

2 Part 2: Regular Expressions (REGEX)

2.1 Deliverable 5: Create Objects Containing a Vector of Characters and Explore using Regex

```
animals <- c("jaguar", "jay", "bat") # create a list vector

stringr::str_detect(animals, "j") # check if any element in the vector contains the characte
```

```
[1]  TRUE  TRUE FALSE
```

The function identified any element that contained the letter "j" and returned a boolean.

```
stringr::str_extract(animals, "j") # extract all instances of the character "j" from each el
```

```
[1] "j" "j" NA
```

```
stringr::str_locate(animals, "j") # locate position of first instance of "j" in each element
```

```
     start end
[1,]     1   1
[2,]     1   1
[3,]    NA  NA
```

The function tells us the location of the first instance of "j" in each element.

```
stringr::str_detect(animals, "jag") # check if any element contains the string "jag"
```

```
[1]  TRUE FALSE FALSE
```

```
wows <- c("wow", "WoW", "WOW") # assign list vector

stringr::str_detect(wows, "WOW") # check if any elementcontains the string "WOW"
```

```
[1] FALSE FALSE  TRUE
```

2.2 Deliverable 6: Understanding and Using the Regex Meta Characters

```r
math <- c("1=2", "14+5", "3-5") # assign list vector
#stringr::str_detect(math, "+") # produces an error
stringr::str_detect(math, "\\+") # check if any element contains the "+" character
```

```
[1] FALSE  TRUE FALSE
```

```r
strings <- c("cat", "cut", "cue") # assign list vector
stringr::str_extract(strings, "c.") # detect letter c and next character
```

```
[1] "ca" "cu" "cu"
```

```r
stringr::str_extract(strings, "c.t") # detect letter c, any number of characters, then t
```

```
[1] "cat" "cut" NA
```

```r
strings2 <- c("a", "b", "c") # assign list vector
stringr::str_detect(strings2, "[ac]") # identify anything with a or c
```

```
[1]  TRUE FALSE  TRUE
```

```r
numbers <- c("1","2","3","4","5","6","7","8","9") # assign list vector
stringr::str_detect(numbers, "[2-7]") # identify numbers between 2 and 7
```

```
[1] FALSE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE FALSE
```

```r
sentence <- "This is a long sentence with 2 numbers with 1 digits." # assign string vector
stringr::str_locate_all(sentence, "[1-2a-b]") # locate any numbers from 1-2 and letter from a
```

```
[[1]]
     start end
[1,]     9   9
[2,]    30  30
[3,]    35  35
[4,]    45  45
```

This output means that there is a character from our search criteria `1-2a-b` at the positions listed in the sentence. `Start` and `End` signify the starting position of the located character and the ending position respectively. So `Start - 9` and `End - 9` means there is a character from our search criteria at the 9th index of the sentence.

```
col <- c("colour", "color", "farver") # assign list vector
stringr::str_detect(col, "colou?r") # check if any element contains the string "colour" or "
```

```
[1]  TRUE  TRUE FALSE
```

```
sentences <- c("The year was 1776.", "Alexander Hamilton died at 47.") # assign string vecto
stringr::str_extract(sentences, "\\d{4}") # extract all instances of a four-digit number
```

```
[1] "1776" NA
```

2.3 Deliverable 7: Understanding and Using the Regex Anchors

```
seasons <- c("The summer is hot this year","The spring is a lovely time", # assign string ve
"Winter is my favorite time of the year","Fall is a time of peace")
stringr::str_detect(seasons, "^The") # check if any element starts with "The"
```

```
[1]  TRUE  TRUE FALSE FALSE
```

```
stringr::str_extract(seasons, "^The") # extract all instances that start with "The"
```

```
[1] "The" "The" NA    NA
```

```
stringr::str_detect(seasons, "year$") # check if any element ends with "year"
```

```
[1]  TRUE FALSE  TRUE FALSE
```

```
folder_names <- c("analysis","data-raw","data","R") # assing string vector
stringr::str_detect(folder_names, "^data$") # check if any element is exactly "data"
```

```
[1] FALSE FALSE  TRUE FALSE
```

3 Part 3: Web Scraping

3.1 Deliverable 8: Reading html files into R and Manipulating with readr

```
weatherlink <- "https://forecast.weather.gov/MapClick.php?lat=38.95604000000003&lon=-77.1782
weatherlink <- rvest::read_html(weatherlink) # read html file into R

forecasthtml <- rvest::html_nodes(weatherlink,
                "detailed-forecast-body b, .forecast-text") # select nodes with class "foreca
forecasttext <- rvest::html_text(forecasthtml)
forecasttext
```

```
 [1] "Mostly cloudy, with a low around 28. Northwest wind 5 to 8 mph becoming calm  in the ev
 [2] "Mostly sunny, with a high near 44. Northwest wind 3 to 7 mph. "
 [3] "Mostly cloudy, with a low around 29. Calm wind. "
 [4] "Snow before 4pm, then snow, possibly mixed with rain.  High near 35. Calm wind becoming
 [5] "Snow, possibly mixed with rain, becoming all snow after 7pm.  Low around 31. Light nort
 [6] "Snow likely before 1pm, then rain and snow.  High near 37. Chance of precipitation is 8
 [7] "Rain.  Low around 34. Chance of precipitation is 100%."
 [8] "A chance of rain before 1pm.  Mostly cloudy, with a high near 47. Chance of precipitati
 [9] "Partly cloudy, with a low around 26."
[10] "Mostly sunny, with a high near 37."
[11] "Mostly cloudy, with a low around 26."
[12] "Rain and snow likely.  Mostly cloudy, with a high near 43. Chance of precipitation is 6
[13] "Rain.  Low around 37. Chance of precipitation is 90%."
[14] "Rain.  High near 49. Chance of precipitation is 90%."
```

```
paste(forecasttext, collapse = " ") # combine all text into one string
```

```
[1] "Mostly cloudy, with a low around 28. Northwest wind 5 to 8 mph becoming calm  in the eve
```

3.2 Deliverable 9: Webscraping with Given CSS Fields

```
starwars <- rvest::read_html("https://rvest.tidyverse.org/articles/starwars") # read in data

films <- starwars %>%
  rvest::html_elements("section") # select all elements with class "section"
films
```

```
{xml_nodeset (7)}
[1] <section><h2 data-id="1">\nThe Phantom Menace\n</h2>\n<p>\nReleased: 1999 ...
[2] <section><h2 data-id="2">\nAttack of the Clones\n</h2>\n<p>\nReleased: 20 ...
[3] <section><h2 data-id="3">\nRevenge of the Sith\n</h2>\n<p>\nReleased: 200 ...
[4] <section><h2 data-id="4">\nA New Hope\n</h2>\n<p>\nReleased: 1977-05-25\n ...
[5] <section><h2 data-id="5">\nThe Empire Strikes Back\n</h2>\n<p>\nReleased: ...
[6] <section><h2 data-id="6">\nReturn of the Jedi\n</h2>\n<p>\nReleased: 1983 ...
[7] <section><h2 data-id="7">\nThe Force Awakens\n</h2>\n<p>\nReleased: 2015- ...
```

```
title <- films %>%
  rvest::html_element("h2") %>% # select all elements with the h2 element
  rvest::html_text2() # extract text from h2 elements
title
```

```
[1] "The Phantom Menace"      "Attack of the Clones"
[3] "Revenge of the Sith"     "A New Hope"
[5] "The Empire Strikes Back" "Return of the Jedi"
[7] "The Force Awakens"
```

```
episode <- films %>%
  rvest::html_element("h2") %>% # select all elements with the h2 element
  rvest::html_attr("data-id") %>% # select attributes from data-id
  readr::parse_integer() # parse integers
episode
```

```
[1] 1 2 3 4 5 6 7
```

3.3 Deliverable 10: Webscraping Tabular Data

```
html <- rvest::read_html("https://en.wikipedia.org/w/index.php?title=The_Lego_Movie&oldid=998

html %>%
  rvest::html_element(".tracklist") %>%
  rvest::html_table()
```

```
# A tibble: 29 x 4
   No.   Title                       `Performer(s)`                       Length
   <chr> <chr>                       <chr>                                <chr>
 1 1.    "\"Everything Is Awesome\"" "Tegan and Sara featuring The Lonel~ 2:43
```

```
 2 2.    "\"Prologue\""              ""                                  2:28
 3 3.    "\"Emmett's Morning\""      ""                                  2:00
 4 4.    "\"Emmett Falls in Love\""  ""                                  1:11
 5 5.    "\"Escape\""                ""                                  3:26
 6 6.    "\"Into the Old West\""     ""                                  1:00
 7 7.    "\"Wyldstyle Explains\""    ""                                  1:21
 8 8.    "\"Emmett's Mind\""         ""                                  2:17
 9 9.    "\"The Transformation\""    ""                                  1:46
10 10.   "\"Saloons and Wagons\""    ""                                  3:38
# i 19 more rows
```

4 Part 4: Collecting Primary Social Media Data

4.1 Deliverable 11: Prepare the RedditExtractoR Package

```
library(RedditExtractoR) # library package
```

```
Warning: package 'RedditExtractoR' was built under R version 4.3.3
```

4.2 Deliverable 12: Extract Data by Exploring Reddit and Specific Subreddits

```
pelosubred <- RedditExtractoR::find_subreddits("peloton") # extract subreddits
```

```
parsing URLs on page 1...
parsing URLs on page 2...
```

```
#RedditExtractoR::find_subreddits("python") #by keyword
#RedditExtractoR::find_thread_urls('https://www.reddit.com/subreddits/search.json?limit=100&
#RedditExtractoR::get_thread_content('https://www.reddit.com/subreddits/search.json?limit=10
#RedditExtractoR::get_user_content('coniecakes') #data related to users
```

4.2.1 Deliverable 13: Prepare the rtoot Package and Authenticate with the Public API

```
library(rtoot) # library rtoot
```

```
Warning: package 'rtoot' was built under R version 4.3.3
```

```
#rtoot::auth_setup(instance = "mastodon.social", # obtain public api key from mastadon social
#          type = NULL,
#          name = NULL,
#          path = NULL,
#          clipboard = FALSE,
#          verbose = TRUE,
#          browser = TRUE
#)
```

4.2.1.1 Collect toots using Various Endpoints

```
rtoot::get_instance_general(instance = "mastodon.social", token = "public") # create an insta
```

```
$uri
[1] "mastodon.social"

$title
[1] "Mastodon"

$short_description
[1] "The original server operated by the Mastodon gGmbH non-profit"

$description
[1] ""

$email
[1] "staff@mastodon.social"

$version
[1] "4.4.0-nightly.2025-02-07"

$urls
$urls$streaming_api
[1] "wss://streaming.mastodon.social"


$stats
$stats$user_count
[1] 2507533
```

```
$stats$status_count
[1] 119425663

$stats$domain_count
[1] 81542


$thumbnail
[1] "https://files.mastodon.social/site_uploads/files/000/000/001/@1x/57c12f441d083cde.png"

$languages
$languages[[1]]
[1] "en"


$registrations
[1] TRUE

$approval_required
[1] FALSE

$invites_enabled
[1] TRUE

$configuration
$configuration$accounts
$configuration$accounts$max_featured_tags
[1] 10


$configuration$statuses
$configuration$statuses$max_characters
[1] 500

$configuration$statuses$max_media_attachments
[1] 4

$configuration$statuses$characters_reserved_per_url
[1] 23


$configuration$media_attachments
```

```
$configuration$media_attachments$supported_mime_types
$configuration$media_attachments$supported_mime_types[[1]]
[1] "image/jpeg"

$configuration$media_attachments$supported_mime_types[[2]]
[1] "image/png"

$configuration$media_attachments$supported_mime_types[[3]]
[1] "image/gif"

$configuration$media_attachments$supported_mime_types[[4]]
[1] "image/heic"

$configuration$media_attachments$supported_mime_types[[5]]
[1] "image/heif"

$configuration$media_attachments$supported_mime_types[[6]]
[1] "image/webp"

$configuration$media_attachments$supported_mime_types[[7]]
[1] "image/avif"

$configuration$media_attachments$supported_mime_types[[8]]
[1] "video/webm"

$configuration$media_attachments$supported_mime_types[[9]]
[1] "video/mp4"

$configuration$media_attachments$supported_mime_types[[10]]
[1] "video/quicktime"

$configuration$media_attachments$supported_mime_types[[11]]
[1] "video/ogg"

$configuration$media_attachments$supported_mime_types[[12]]
[1] "audio/wave"

$configuration$media_attachments$supported_mime_types[[13]]
[1] "audio/wav"

$configuration$media_attachments$supported_mime_types[[14]]
[1] "audio/x-wav"
```

```
$configuration$media_attachments$supported_mime_types[[15]]
[1] "audio/x-pn-wave"

$configuration$media_attachments$supported_mime_types[[16]]
[1] "audio/vnd.wave"

$configuration$media_attachments$supported_mime_types[[17]]
[1] "audio/ogg"

$configuration$media_attachments$supported_mime_types[[18]]
[1] "audio/vorbis"

$configuration$media_attachments$supported_mime_types[[19]]
[1] "audio/mpeg"

$configuration$media_attachments$supported_mime_types[[20]]
[1] "audio/mp3"

$configuration$media_attachments$supported_mime_types[[21]]
[1] "audio/webm"

$configuration$media_attachments$supported_mime_types[[22]]
[1] "audio/flac"

$configuration$media_attachments$supported_mime_types[[23]]
[1] "audio/aac"

$configuration$media_attachments$supported_mime_types[[24]]
[1] "audio/m4a"

$configuration$media_attachments$supported_mime_types[[25]]
[1] "audio/x-m4a"

$configuration$media_attachments$supported_mime_types[[26]]
[1] "audio/mp4"

$configuration$media_attachments$supported_mime_types[[27]]
[1] "audio/3gpp"

$configuration$media_attachments$supported_mime_types[[28]]
[1] "video/x-ms-asf"
```

```
$configuration$media_attachments$image_size_limit
[1] 16777216

$configuration$media_attachments$image_matrix_limit
[1] 33177600

$configuration$media_attachments$video_size_limit
[1] 103809024

$configuration$media_attachments$video_frame_rate_limit
[1] 120

$configuration$media_attachments$video_matrix_limit
[1] 8294400


$configuration$polls
$configuration$polls$max_options
[1] 4

$configuration$polls$max_characters_per_option
[1] 50

$configuration$polls$min_expiration
[1] 300

$configuration$polls$max_expiration
[1] 2629746



$contact_account
$contact_account$id
[1] "13179"

$contact_account$username
[1] "Mastodon"

$contact_account$acct
[1] "Mastodon"

$contact_account$display_name
[1] "Mastodon"
```

```
$contact_account$locked
[1] FALSE

$contact_account$bot
[1] FALSE

$contact_account$discoverable
[1] TRUE

$contact_account$indexable
[1] FALSE

$contact_account$group
[1] FALSE

$contact_account$created_at
[1] "2016-11-23T00:00:00.000Z"

$contact_account$note
[1] "<p>Free, open-source decentralized social media platform.</p>"

$contact_account$url
[1] "https://mastodon.social/@Mastodon"

$contact_account$uri
[1] "https://mastodon.social/users/Mastodon"

$contact_account$avatar
[1] "https://files.mastodon.social/accounts/avatars/000/013/179/original/b4ceb19c9c54ec7e.png

$contact_account$avatar_static
[1] "https://files.mastodon.social/accounts/avatars/000/013/179/original/b4ceb19c9c54ec7e.png

$contact_account$header
[1] "https://files.mastodon.social/accounts/headers/000/013/179/original/1375be116fbe0f1d.png

$contact_account$header_static
[1] "https://files.mastodon.social/accounts/headers/000/013/179/original/1375be116fbe0f1d.png

$contact_account$followers_count
[1] 837054
```

```
$contact_account$following_count
[1] 4

$contact_account$statuses_count
[1] 288

$contact_account$last_status_at
[1] "2025-01-27"

$contact_account$hide_collections
[1] FALSE

$contact_account$noindex
[1] FALSE

$contact_account$emojis
list()

$contact_account$roles
list()

$contact_account$fields
$contact_account$fields[[1]]
$contact_account$fields[[1]]$name
[1] "Homepage"

$contact_account$fields[[1]]$value
[1] "<a href=\"https://joinmastodon.org\" target=\"_blank\" rel=\"nofollow noopener me\" tra

$contact_account$fields[[1]]$verified_at
[1] "2018-10-31T04:11:00.076+00:00"


$contact_account$fields[[2]]
$contact_account$fields[[2]]$name
[1] "Patreon"

$contact_account$fields[[2]]$value
[1] "<a href=\"https://patreon.com/mastodon\" target=\"_blank\" rel=\"nofollow noopener me\"

$contact_account$fields[[2]]$verified_at
NULL
```

```
$contact_account$fields[[3]]
$contact_account$fields[[3]]$name
[1] "GitHub"

$contact_account$fields[[3]]$value
[1] "<a href=\"https://github.com/mastodon\" target=\"_blank\" rel=\"nofollow noopener me\" t

$contact_account$fields[[3]]$verified_at
[1] "2023-07-21T13:27:45.996+00:00"




$rules
$rules[[1]]
$rules[[1]]$id
[1] "1"

$rules[[1]]$text
[1] "Sexually explicit or violent media must be marked as sensitive or with a content warning

$rules[[1]]$hint
[1] "This includes content that is particularly provocative even if it may not show specific


$rules[[2]]
$rules[[2]]$id
[1] "2"

$rules[[2]]$text
[1] "No racism, sexism, homophobia, transphobia, ableism, xenophobia, or casteism."

$rules[[2]]$hint
[1] "Transphobic behavior such as intentional misgendering and deadnaming is strictly prohibi


$rules[[3]]
$rules[[3]]$id
[1] "3"

$rules[[3]]$text
[1] "No incitement of violence or promotion of violent ideologies"
```

```
$rules[[3]]$hint
[1] "Calling for people or groups to be assassinated, murdered, or attacked physically is str


$rules[[4]]
$rules[[4]]$id
[1] "4"

$rules[[4]]$text
[1] "No harassment, block evasion, dogpiling, or doxxing of others"

$rules[[4]]$hint
[1] "Repeat attempts to communicate with users who have blocked you or creation of accounts s


$rules[[5]]
$rules[[5]]$id
[1] "7"

$rules[[5]]$text
[1] "Do not share information widely-known to be false and misleading"

$rules[[5]]$hint
[1] "False and misleading information and links from low-quality sources may not be posted, e


$rules[[6]]
$rules[[6]]$id
[1] "1008"

$rules[[6]]$text
[1] "Content created by others must be attributed, and use of generative AI must be disclosed

$rules[[6]]$hint
[1] "Content created by others must clearly provide a reference to the author, creator, or so



attr(,"headers")
# A tibble: 1 x 3
  rate_limit rate_remaining rate_reset
  <chr>      <chr>          <dttm>
```

```
1 300        299            2025-02-10 02:25:00
```

```
mastoactivity <- rtoot::get_instance_activity(instance = "mastodon.social") # get activity fi
mastoactivity
```

```
# A tibble: 12 x 4
   week                statuses logins registrations
   <dttm>                 <int>  <int>         <int>
 1 2025-02-09 03:35:57    18611   9599           317
 2 2025-02-02 03:35:57  1030481 208226         27037
 3 2025-01-26 03:35:57   424169 152011         12396
 4 2025-01-19 03:35:57        0      0             0
 5 2025-01-12 03:35:57        0      0             0
 6 2025-01-05 03:35:57        0      0             0
 7 2024-12-29 03:35:57        0      0             0
 8 2024-12-22 03:35:57        0      0             0
 9 2024-12-15 03:35:57        0      0             0
10 2024-12-08 03:35:57        0      0             0
11 2024-12-01 03:35:57        0      0             0
12 2024-11-24 03:35:57        0      0             0
```

```
mastotrends <- rtoot::get_instance_trends(instance = "mastodon.social") # get trends from Mas
mastotrends[25,]
```

```
# A tibble: 1 x 6
  id       name               url                    day        accounts  uses
  <chr>    <chr>              <chr>                  <date>        <int> <int>
1 12661389 eggcornamovieorplay https://mastodon.socia~ 2025-02-07        0     0
```

```
id <- "211346"
#rtoot::get_account_followers(id)
#rtoot::get_account_following(id)
#rtoot::get_account_statuses(id)

# keeps returning an error 404 code
```

5 Part 5: Analyzing Secondary Social Media Data

5.1 Deliverable 14: Create the IRA Data Object

```
library(tidyverse) # library tidyverse
```

```
Warning: package 'lubridate' was built under R version 4.3.3
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v forcats   1.0.0     v readr     2.1.5
v ggplot2   3.5.1     v stringr   1.5.1
v lubridate 1.9.4     v tibble    3.2.1
v purrr     1.0.2     v tidyr     1.3.1
-- 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 becor
```

```
ira_tweets <- readr::read_csv("/Users/coniecakes/Library/CloudStorage/OneDrive-Personal/001.
```

```
Rows: 243891 Columns: 21
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (13): author, content, region, language, publish_date, harvested_date, p...
dbl  (8): external_author_id, following, followers, updates, retweet, new_ju...

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.
```

```
utils::head(ira_tweets) # view the first few rows of the data frame
```

```
# A tibble: 6 x 21
  external_author_id author content  region language publish_date harvested_date
               <dbl> <chr>  <chr>    <chr>  <chr>    <chr>        <chr>
1           9.06e17 10_GOP "\"We h~ Unkno~ English  10/1/2017 1~ 10/1/2017 19:~
2           9.06e17 10_GOP "Marsha~ Unkno~ English  10/1/2017 2~ 10/1/2017 22:~
3           9.06e17 10_GOP "Daught~ Unkno~ English  10/1/2017 2~ 10/1/2017 22:~
4           9.06e17 10_GOP "JUST I~ Unkno~ English  10/1/2017 2~ 10/1/2017 23:~
5           9.06e17 10_GOP "19,000~ Unkno~ English  10/1/2017 2~ 10/1/2017 2:13
```

```
6            9.06e17 10_GOP "Dan Bo~ Unkno~ English  10/1/2017 2~ 10/1/2017 2:47
# i 14 more variables: following <dbl>, followers <dbl>, updates <dbl>,
#   post_type <chr>, account_type <chr>, retweet <dbl>, account_category <chr>,
#   new_june_2018 <dbl>, alt_external_id <dbl>, tweet_id <dbl>,
#   article_url <chr>, tco1_step1 <chr>, tco2_step1 <chr>, tco3_step1 <chr>
```

```
utils::tail(ira_tweets) # view the last few rows of the data frame
```

```
# A tibble: 6 x 21
  external_author_id author  content region language publish_date harvested_date
               <dbl> <chr>   <chr>   <chr>  <chr>    <chr>        <chr>
1         2497991305 AUSTIN~ Former~ Unite~ English  3/8/2017 8:~ 3/8/2017 8:59
2         2497991305 AUSTIN~ BREAKI~ Unite~ English  3/8/2017 8:~ 3/8/2017 8:59
3         2497991305 AUSTIN~ Why me~ Unite~ English  3/8/2017 8:~ 3/8/2017 9:00
4         2497991305 AUSTIN~ How we~ Unite~ English  3/8/2017 8:~ 3/8/2017 8:59
5         2497991305 AUSTIN~ John H~ Unite~ English  3/8/2017 8:~ 3/8/2017 8:59
6         2497991305 AUSTIN~ Fossil~ Unite~ English  3/8/2017 8:~ 3/8/2017 8:59
# i 14 more variables: following <dbl>, followers <dbl>, updates <dbl>,
#   post_type <chr>, account_type <chr>, retweet <dbl>, account_category <chr>,
#   new_june_2018 <dbl>, alt_external_id <dbl>, tweet_id <dbl>,
#   article_url <chr>, tco1_step1 <chr>, tco2_step1 <chr>, tco3_step1 <chr>
```

```
class(ira_tweets) # check the class of the data frame
```

```
[1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

5.2 Deliverable 15: Using the count() function, determine how many tweets are from each region in the dataset?

```
ira_tweets %>%
  dplyr::count(region) # count number of tweets per region
```

```
# A tibble: 17 x 2
   region                n
   <chr>             <int>
 1 Azerbaijan         6238
 2 Egypt                 1
 3 France               22
 4 Germany             656
```

```
 5 Iraq                     17
 6 Israel                  675
 7 Italy                  6278
 8 Japan                     2
 9 Russian Federation     1295
10 Serbia                    2
11 Turkey                    9
12 Ukraine                1232
13 United Arab Emirates   9598
14 United Kingdom          600
15 United States        159369
16 Unknown               57859
17 <NA>                     38
```

5.3 Deliverable 16: How many tweets are from each language in the dataset?

```
ira_tweets %>%
  dplyr::count(language) # count number of tweets per language
```

```
# A tibble: 49 x 2
   language       n
   <chr>      <int>
 1 Albanian      51
 2 Arabic       570
 3 Bulgarian    549
 4 Catalan       76
 5 Croatian      46
 6 Czech          5
 7 Danish        10
 8 Dutch        169
 9 English   190252
10 Estonian      96
# i 39 more rows
```

5.4 Deliverable 17: What are the account types in the dataset, and how many tweets are from each account type in the dataset?

```
ira_tweets %>%
  dplyr::count(account_type) # count number of tweets per account type
```

```
# A tibble: 14 x 2
   account_type      n
   <chr>         <int>
 1 ?               608
 2 Arabic          475
 3 Commercial      339
 4 French           22
 5 German         1297
 6 Hashtager     27349
 7 Italian        6278
 8 Koch            384
 9 Left          36072
10 Right        114810
11 Russian       44964
12 Spanish           1
13 Ukranian          1
14 local         11291
```

5.5 Deliverable 18: What are the account categories in the dataset, and how many tweets are from each account category in the dataset?

```
ira_tweets %>%
  dplyr::count(account_category) # count number of tweets per account category
```

```
# A tibble: 8 x 2
  account_category      n
  <chr>             <int>
1 Commercial          339
2 Fearmonger          384
3 HashtagGamer      27349
4 LeftTroll         36072
5 NewsFeed          11291
6 NonEnglish        53038
7 RightTroll       114810
8 Unknown             608
```

6 Part 6: Webscraping and REGEX in Python

6.1 Deliverable 19: Using Regex in Python

```
import re
```

```
text = "The rain in Spain"

if re.search("rain", text): # search for text in string
  print("Yes, there is at least one match!")
else:
  print("No match")
```

```
Yes, there is at least one match!
```

```
text = "The rain in Spain falls mainly in the plain!"
matches = re.findall("ain", text) # search for string in text string
print(matches)
```

```
['ain', 'ain', 'ain', 'ain']
```

```
text = "The rain in Spain"
split_text = re.split("", text) # split string into list of substrings based on pattern
print(split_text)
```

```
['', 'T', 'h', 'e', ' ', 'r', 'a', 'i', 'n', ' ', 'i', 'n', ' ', 'S', 'p', 'a', 'i', 'n', ''
```

```
text = "The rain in Spain"
replaced_text = re.sub("Spain", "France", text) # substitute string with another string
print(replaced_text)
```

```
The rain in France
```

6.2 Deliverable 20: Webscraping with BeautifulSoup

```
import requests
from bs4 import BeautifulSoup
```

```
url = "http://quotes.toscrape.com/"
response = requests.get(url) # get response from website
soup = BeautifulSoup(response.text, 'html.parser') # parse text
print(soup)
```

```
<!DOCTYPE html>

<html lang="en">
<head>
<meta charset="utf-8"/>
<title>Quotes to Scrape</title>
<link href="/static/bootstrap.min.css" rel="stylesheet"/>
<link href="/static/main.css" rel="stylesheet"/>
</head>
<body>
<div class="container">
<div class="row header-box">
<div class="col-md-8">
<h1>
<a href="/" style="text-decoration: none">Quotes to Scrape</a>
</h1>
</div>
<div class="col-md-4">
<p>
<a href="/login">Login</a>
</p>
</div>
</div>
<div class="row">
<div class="col-md-8">
<div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
<span class="text" itemprop="text">"The world as we have created it is a process of our think
<span>by <small class="author" itemprop="author">Albert Einstein</small>
<a href="/author/Albert-Einstein">(about)</a>
</span>
<div class="tags">
            Tags:
            <meta class="keywords" content="change,deep-thoughts,thinking,world" itemprop="ke
```

```
<a class="tag" href="/tag/change/page/1/">change</a>
<a class="tag" href="/tag/deep-thoughts/page/1/">deep-thoughts</a>
<a class="tag" href="/tag/thinking/page/1/">thinking</a>
<a class="tag" href="/tag/world/page/1/">world</a>
</div>
</div>
<div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
<span class="text" itemprop="text">"It is our choices, Harry, that show what we truly are, fa
<span>by <small class="author" itemprop="author">J.K. Rowling</small>
<a href="/author/J-K-Rowling">(about)</a>
</span>
<div class="tags">
            Tags:
            <meta class="keywords" content="abilities,choices" itemprop="keywords"/>
<a class="tag" href="/tag/abilities/page/1/">abilities</a>
<a class="tag" href="/tag/choices/page/1/">choices</a>
</div>
</div>
<div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
<span class="text" itemprop="text">"There are only two ways to live your life. One is as thou
<span>by <small class="author" itemprop="author">Albert Einstein</small>
<a href="/author/Albert-Einstein">(about)</a>
</span>
<div class="tags">
            Tags:
            <meta class="keywords" content="inspirational,life,live,miracle,miracles" itempro
<a class="tag" href="/tag/inspirational/page/1/">inspirational</a>
<a class="tag" href="/tag/life/page/1/">life</a>
<a class="tag" href="/tag/live/page/1/">live</a>
<a class="tag" href="/tag/miracle/page/1/">miracle</a>
<a class="tag" href="/tag/miracles/page/1/">miracles</a>
</div>
</div>
<div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
<span class="text" itemprop="text">"The person, be it gentleman or lady, who has not pleasure
<span>by <small class="author" itemprop="author">Jane Austen</small>
<a href="/author/Jane-Austen">(about)</a>
</span>
<div class="tags">
            Tags:
            <meta class="keywords" content="aliteracy,books,classic,humor" itemprop="keywords
<a class="tag" href="/tag/aliteracy/page/1/">aliteracy</a>
<a class="tag" href="/tag/books/page/1/">books</a>
```

```
<a class="tag" href="/tag/classic/page/1/">classic</a>
<a class="tag" href="/tag/humor/page/1/">humor</a>
</div>
</div>
<div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
<span class="text" itemprop="text">"Imperfection is beauty, madness is genius and it's bette
<span>by <small class="author" itemprop="author">Marilyn Monroe</small>
<a href="/author/Marilyn-Monroe">(about)</a>
</span>
<div class="tags">
            Tags:
            <meta class="keywords" content="be-yourself,inspirational" itemprop="keywords"/>
<a class="tag" href="/tag/be-yourself/page/1/">be-yourself</a>
<a class="tag" href="/tag/inspirational/page/1/">inspirational</a>
</div>
</div>
<div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
<span class="text" itemprop="text">"Try not to become a man of success. Rather become a man
<span>by <small class="author" itemprop="author">Albert Einstein</small>
<a href="/author/Albert-Einstein">(about)</a>
</span>
<div class="tags">
            Tags:
            <meta class="keywords" content="adulthood,success,value" itemprop="keywords"/>
<a class="tag" href="/tag/adulthood/page/1/">adulthood</a>
<a class="tag" href="/tag/success/page/1/">success</a>
<a class="tag" href="/tag/value/page/1/">value</a>
</div>
</div>
<div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
<span class="text" itemprop="text">"It is better to be hated for what you are than to be lov
<span>by <small class="author" itemprop="author">André Gide</small>
<a href="/author/Andre-Gide">(about)</a>
</span>
<div class="tags">
            Tags:
            <meta class="keywords" content="life,love" itemprop="keywords"/>
<a class="tag" href="/tag/life/page/1/">life</a>
<a class="tag" href="/tag/love/page/1/">love</a>
</div>
</div>
<div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
<span class="text" itemprop="text">"I have not failed. I've just found 10,000 ways that won't
```

```
<span>by <small class="author" itemprop="author">Thomas A. Edison</small>
<a href="/author/Thomas-A-Edison">(about)</a>
</span>
<div class="tags">
            Tags:
            <meta class="keywords" content="edison,failure,inspirational,paraphrased" itempro
<a class="tag" href="/tag/edison/page/1/">edison</a>
<a class="tag" href="/tag/failure/page/1/">failure</a>
<a class="tag" href="/tag/inspirational/page/1/">inspirational</a>
<a class="tag" href="/tag/paraphrased/page/1/">paraphrased</a>
</div>
</div>
<div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
<span class="text" itemprop="text">"A woman is like a tea bag; you never know how strong it
<span>by <small class="author" itemprop="author">Eleanor Roosevelt</small>
<a href="/author/Eleanor-Roosevelt">(about)</a>
</span>
<div class="tags">
            Tags:
            <meta class="keywords" content="misattributed-eleanor-roosevelt" itemprop="keywo
<a class="tag" href="/tag/misattributed-eleanor-roosevelt/page/1/">misattributed-eleanor-roos
</div>
</div>
<div class="quote" itemscope="" itemtype="http://schema.org/CreativeWork">
<span class="text" itemprop="text">"A day without sunshine is like, you know, night."</span>
<span>by <small class="author" itemprop="author">Steve Martin</small>
<a href="/author/Steve-Martin">(about)</a>
</span>
<div class="tags">
            Tags:
            <meta class="keywords" content="humor,obvious,simile" itemprop="keywords"/>
<a class="tag" href="/tag/humor/page/1/">humor</a>
<a class="tag" href="/tag/obvious/page/1/">obvious</a>
<a class="tag" href="/tag/simile/page/1/">simile</a>
</div>
</div>
<nav>
<ul class="pager">
<li class="next">
<a href="/page/2/">Next <span aria-hidden="true">→</span></a>
</li>
</ul>
</nav>
```

```
</div>
<div class="col-md-4 tags-box">
<h2>Top Ten tags</h2>
<span class="tag-item">
<a class="tag" href="/tag/love/" style="font-size: 28px">love</a>
</span>
<span class="tag-item">
<a class="tag" href="/tag/inspirational/" style="font-size: 26px">inspirational</a>
</span>
<span class="tag-item">
<a class="tag" href="/tag/life/" style="font-size: 26px">life</a>
</span>
<span class="tag-item">
<a class="tag" href="/tag/humor/" style="font-size: 24px">humor</a>
</span>
<span class="tag-item">
<a class="tag" href="/tag/books/" style="font-size: 22px">books</a>
</span>
<span class="tag-item">
<a class="tag" href="/tag/reading/" style="font-size: 14px">reading</a>
</span>
<span class="tag-item">
<a class="tag" href="/tag/friendship/" style="font-size: 10px">friendship</a>
</span>
<span class="tag-item">
<a class="tag" href="/tag/friends/" style="font-size: 8px">friends</a>
</span>
<span class="tag-item">
<a class="tag" href="/tag/truth/" style="font-size: 8px">truth</a>
</span>
<span class="tag-item">
<a class="tag" href="/tag/simile/" style="font-size: 6px">simile</a>
</span>
</div>
</div>
</div>
<footer class="footer">
<div class="container">
<p class="text-muted">
                Quotes by: <a href="https://www.goodreads.com/quotes">GoodReads.com</a>
</p>
<p class="copyright">
                Made with <span class="zyte"> </span> by <a class="zyte" href="https://www.zy
```

```
</p>
</div>
</footer>
</body>
</html>
```

```
quotes = soup.find_all("span", class_="text") # find all span elements with class "text"
for quote in quotes:
  print(quote.text) # print the text of each quote
```

```
"The world as we have created it is a process of our thinking. It cannot be changed without c
"It is our choices, Harry, that show what we truly are, far more than our abilities."
"There are only two ways to live your life. One is as though nothing is a miracle. The other
"The person, be it gentleman or lady, who has not pleasure in a good novel, must be intoleral
"Imperfection is beauty, madness is genius and it's better to be absolutely ridiculous than a
"Try not to become a man of success. Rather become a man of value."
"It is better to be hated for what you are than to be loved for what you are not."
"I have not failed. I've just found 10,000 ways that won't work."
"A woman is like a tea bag; you never know how strong it is until it's in hot water."
"A day without sunshine is like, you know, night."
```

```
authors = soup.find_all("small", class_="author") # find all small elements with class "autho
for author in authors:
  print(author.text)
```

```
Albert Einstein
J.K. Rowling
Albert Einstein
Jane Austen
Marilyn Monroe
Albert Einstein
André Gide
Thomas A. Edison
Eleanor Roosevelt
Steve Martin
```

6.3 Deliverable 21: Combining Regex with BeautifulSoup

```
from bs4 import BeautifulSoup

html_doc = """
<html><head><title>The Dormouse's story</title></head>
<body>
<p class="title"><b>The Dormouse's story</b></p>
46
<p class="story">Once upon a time there were three little sisters; and their names were
<a href="http://example.com/elsie" class="sister" id="link1">Elsie</a>,
<a href="http://example.com/lacie" class="sister" id="link2">Lacie</a> and
<a href="http://example.com/tillie" class="sister" id="link3">Tillie</a>;
and they lived at the bottom of a well.</p>
"""

soup = BeautifulSoup(html_doc, 'html.parser')

# Find all 'a' tags with 'class' attribute containing 'sister'
for tag in soup.find_all('a', class_=re.compile("sister")):
  print(tag)
```

```
<a class="sister" href="http://example.com/elsie" id="link1">Elsie</a>
<a class="sister" href="http://example.com/lacie" id="link2">Lacie</a>
<a class="sister" href="http://example.com/tillie" id="link3">Tillie</a>
```

```
text = "Contact us at info@example.com or support@example.com"
emails = re.findall(r'\b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b', text)
print(emails)
```

```
['info@example.com', 'support@example.com']
```

6.4 Deliverable 22: Webscraping with Selenium

```
from selenium import webdriver
from selenium.webdriver.common.by import By  # Import By module

browser=webdriver.Chrome() # Launch Chrome Browser

browser.get("http://quotes.toscrape.com/") # Open the website
```

```
quotes = browser.find_elements(By.CSS_SELECTOR, ".text") # Find all elements with class "text

for quote in quotes:
  print(quote.text)
```

```
"The world as we have created it is a process of our thinking. It cannot be changed without c
"It is our choices, Harry, that show what we truly are, far more than our abilities."
"There are only two ways to live your life. One is as though nothing is a miracle. The other
"The person, be it gentleman or lady, who has not pleasure in a good novel, must be intolerab
"Imperfection is beauty, madness is genius and it's better to be absolutely ridiculous than a
"Try not to become a man of success. Rather become a man of value."
"It is better to be hated for what you are than to be loved for what you are not."
"I have not failed. I've just found 10,000 ways that won't work."
"A woman is like a tea bag; you never know how strong it is until it's in hot water."
"A day without sunshine is like, you know, night."
```

```
browser.quit()
```