Scraping Twitter

Carolina A. de Lima Salge
Assistant Professor
Terry College of Business
University of Georgia

Business Intelligence
Spring 2021


```
install.packages("httr")

require("httr")

install.packages("jsonlite")

require("jsonlite")
```

```
Console ~/Google Drive/Intrinio:Focus/Marketing/CUB Lecture/
        /var/folders/15/2z483nk52b92jnsbcs58_15c0000gn/T//RtmpJBua8N/downloaded_packages
> require("httr")
> install.packages("jsonlite")
Error in install.packages : Updating loaded packages
> install.packages("jsonlite")
Installing package into '/Users/Andrew/Library/R/3.4/library'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/macosx/el-capitan/contrib/3.4/jsonlite_1.5.tgz'
Content type 'application/x-gzip' length 1114207 bytes (1.1 MB)
==================================================
downloaded 1.1 MB


The downloaded binary packages are in
        /var/folders/15/2z483nk52b92jnsbcs58_15c0000gn/T//RtmpJBua8N/downloaded_packages
> require("jsonlite")
>
```

Terry College of Business
UNIVERSITY OF GEORGIA

Web Scraping

Three common ways to scrape data

- Basic (HTML)
- APIs (Twitter, Yelp)
- Advanced (Javascript)
  - Rselenium: https://rpubs.com/johndharrison/RSelenium-Basics

Web Scraping

Three common ways to scrape data

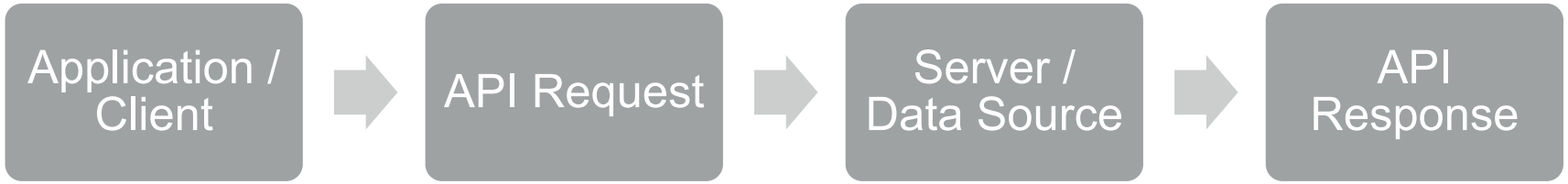
- Basic (HTML)
- APIs (Twitter, Yelp)
- Advanced (Javascript)
  - Rselenium: https://rpubs.com/johndharrison/RSelenium-Basics

API

Stands for **Application Programming Interface**



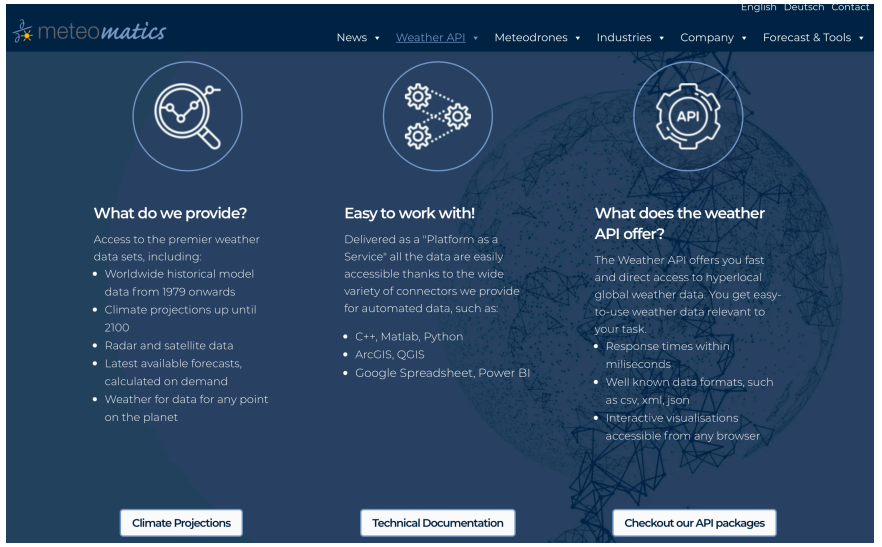
- Written code providing users access to software
- What people use to create interactions with external systems



APIs as Products

Weather Underground packaged its API into a product, selling access to its weather data through the API

https://www.meteomatics.com/en/weather-api/?gclid=Cj0KCQjwuL_8BRCXARIsAGiC51BEYx9zvaM0DPFSwBkDLvju_7u1DzTZkdkBHCUEvQAkIknn-bC7ECcaAubqEALw_wcB



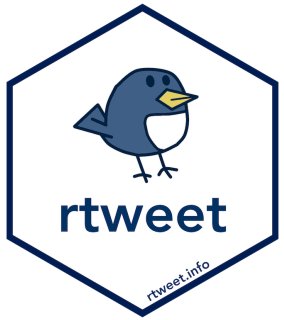
https://towardsdatascience.com/https-medium-com-hiren787-patel-web-scraping-applications-a6f370d31614

Web Scraping APIs in R

- Register an application, generate consumer keys as well as an access token and an access secret
- Use your application to create an authorization token and to send API requests for data
- Transform and store response data in tables

Web Scraping APIs – R & Twitter

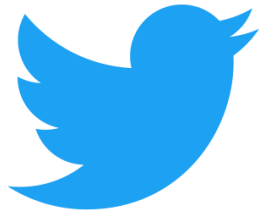
We are the client using R packages as the applications to send out HTTP requests to Twitter's server. The raw API responses we get from Twitter are usually in JSON or XML but one of the packages we will use (rtweet) transforms these files into data frames, making our lives easier



GET request

Response code & data

Server / Data Source



https://www.business-science.io/business/2017/12/27/six-reasons-to-use-R-for-business.html

Scraping Twitter Data

```
library(httr)
library(rtweet)

twitter_endpts <- oauth_endpoints("twitter")
twitter_endpts
```

```
twitter_key <- ""
twitter_secret <- ""
access_token <- ""
access_secret <- ""
```

Put information from your Twitter app here

```
twitter_token <- create_token(app = "twitter",
              consumer_key = twitter_key,
              consumer_secret = twitter_secret,
              access_token = access_token,
              access_secret = access_secret)
```

Connect with Twitter here

Scraping Twitter Data

```
# Randomly sample (approximately 1%) from the live stream of all tweets for 10 secs
sample <- stream_tweets(
  q = "",
  timeout = 10)

# Stream Tweets that contain specific keywords for 30 seconds
big4 <- stream_tweets(
  q = "EY, PwC, Deloitte, KPMG",
  timeout = 30)

# Search for Twitter statuses containing a keyword, phrase, or multiple keywords.
# ONLY RETURNS DATA FROM THE PAST 6-9 DAYS.
# To return more than 18,000 statuses in a single call, set "retryonratelimit" to TRUE.
# Default is to return 100 tweets - to return more, set n to a higher value
# search for a keyword
deloitte_en_tweets <- search_tweets("Deloitte", lang = "en", include_rts = FALSE, n = 5000)
save_as_csv(deloitte_en_tweets, "deloitte_en_tweets.csv", prepend_ids = TRUE, na = "", fileEncoding = "UTF-
8")
```

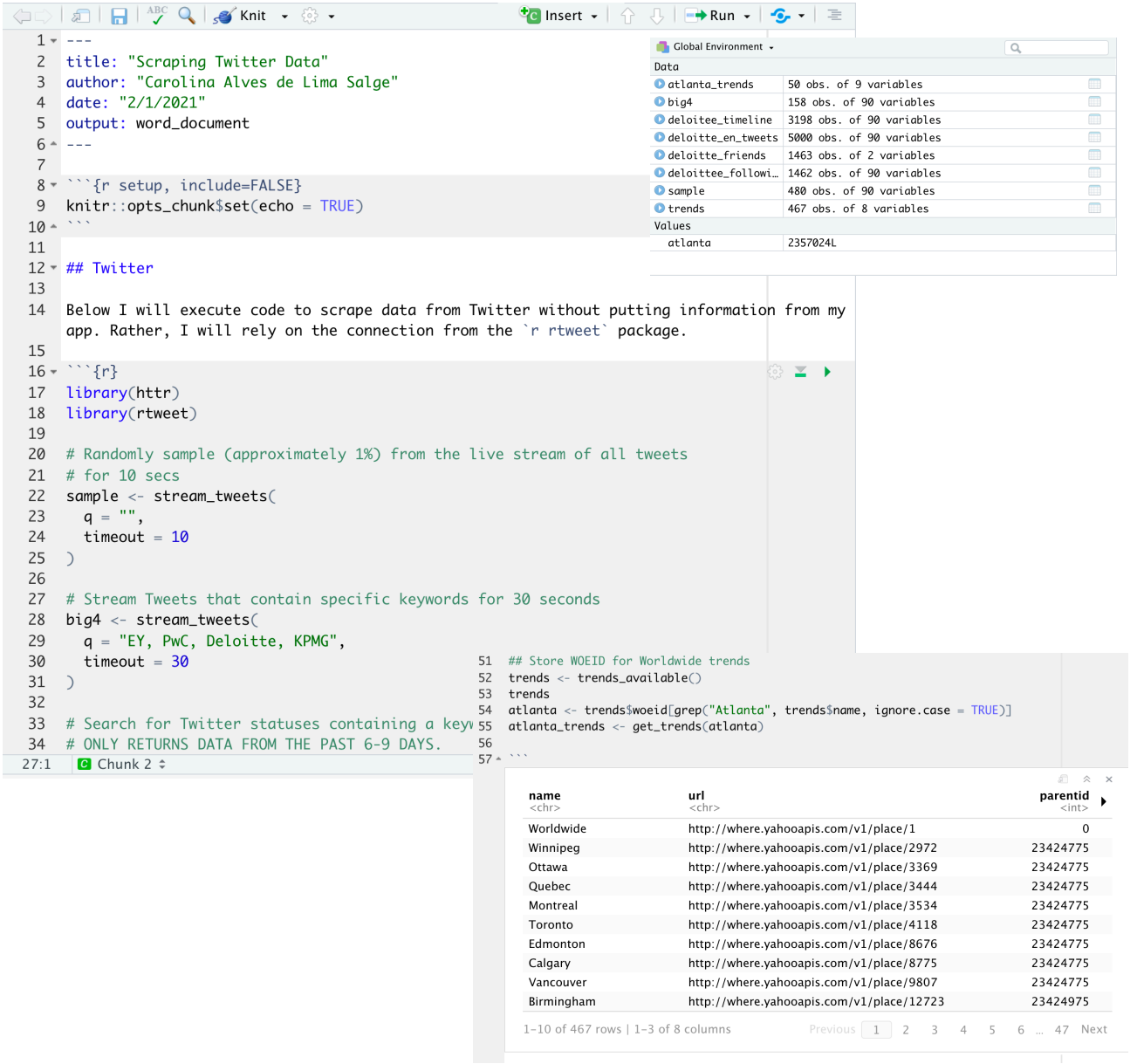
Scraping Twitter Data

```
# Search for the timeline of users
# Provide a user ID or screen name and specify the number of tweets (max of 3,200).
deloitee_timeline <- get_timeline("@Deloitte", n = 3200)

# Get the list of account followed by @Deloitte
# i.e., who does @Deloitte follow?
deloitte_friends <- get_friends("@Deloitte")

# Lookup function and Twitter trends
deloittee_following <- lookup_users(deloitte_friends$user_id)

## Store WOEID for Worldwide trends
trends <- trends_available()
trends
atlanta <- trends$woeid[grep("Atlanta", trends$name, ignore.case = TRUE)]
atlanta_trends <- get_trends(atlanta)
```

Ready to scrape some Twitter data?

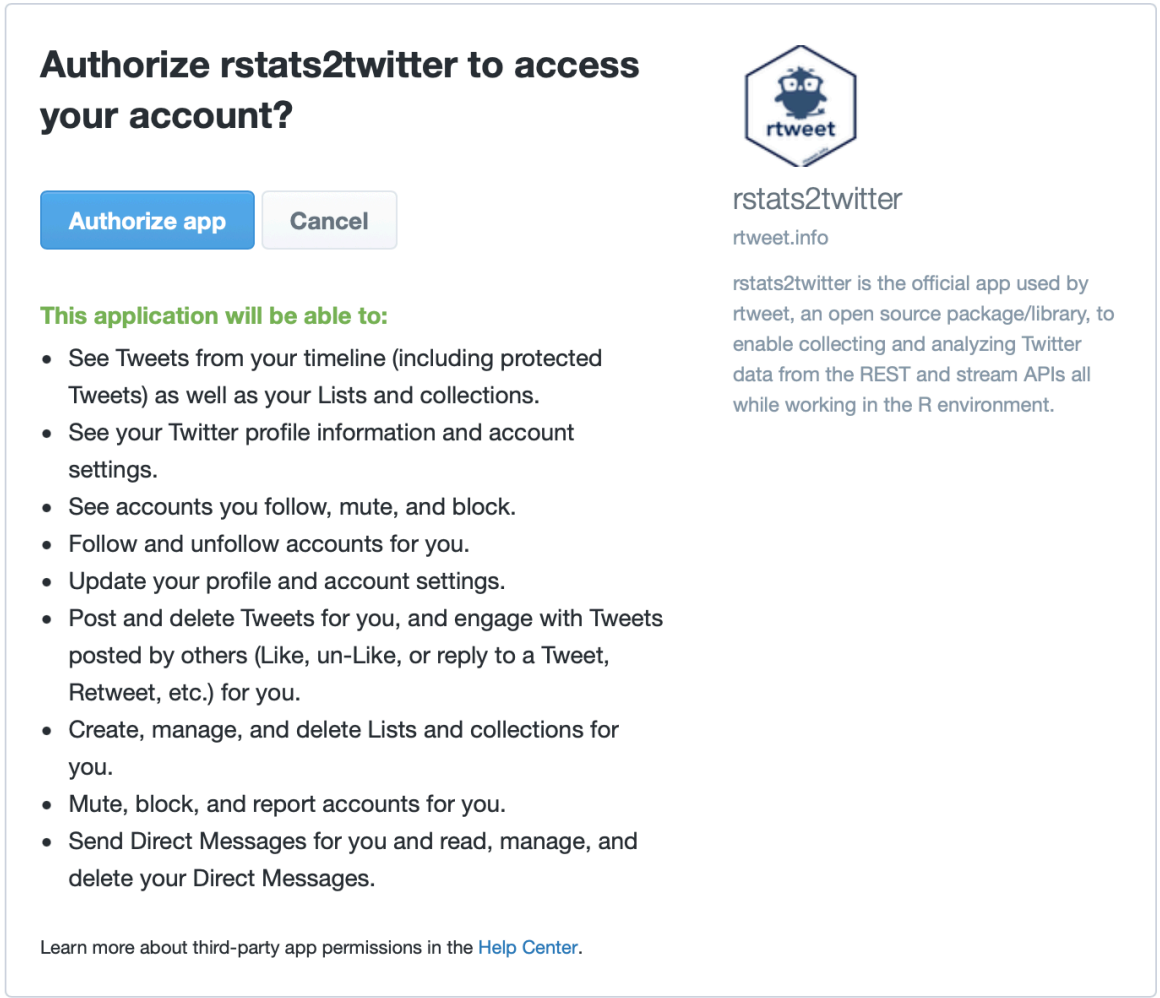
In-Class Exercise

- Open a Rmd file
- Copy and paste code from slides 8 to 10
  - Change the code to add the information from you Twitter app (if your account has not been approved yet, try using the information from rtweet (i.e., ignore code in slide 8 except for packages)
- Execute code in Markdown
- Open the `deloitte_en_tweets.csv` file you saved. How much data do you have?



In-Class Exercise

- Open a Rmd file
- Copy and paste code from slides 8 to 10
  - Change the code to add the information from you Twitter app (if your account has not been approved yet, try using the information from rtweet (i.e., ignore code in slide 8 except for packages)
- Execute code in Markdown
- Open the `deloitte_en_tweets.csv` file you saved. How much data do you have?




```
Authentication complete. Please close this page and return to R.
```

Thank You!