2021

Eindrapport

INDICIA

STAN GOMMANS, MARK LIGTENBERG, ANTHONY VAN ACHT, MIKAIL ÜNLÜ, TIGO JACOBS & EMIEL VERHOEVEN.

Inhoud

Introductie

” Hoe kunnen we op basis van Big Data waarde creëren / toevoegen aan het platform van Let’s Roar ” dit is de hoofdvraag die in overleg is opgesteld. Hier zijn we als groep dan ook mee aan de slag gegaan. Aangezien er geen verdere kennis was in ICT hebben de het project ingedeeld in een aantal sprints. Zo hebben we elke nieuwe sprints verdere diepgang kunnen realiseren en realistische vervolg doelen kunnen stellen.

Wij zijn met 6 studenten aan de slag gegaan met dit project met ieder een andere achtergrond. Zo komen wij van 5 verschillende opeldingen af en zijn wij met verschillende leerdoelen aan deze minor begonnen. Wij hebben zelf echt gekozen voor dit project, omdat we hier meerdere leerdoelen in kwijt konden en hier gelijk een associatie bij voelde. Hierbij zijn Mark en Stan zich vooral bezig gaan houden met het data gedeelte, Mikail, Tigo en Anthony met het analytics gedeelte en Emiel met UX Design.

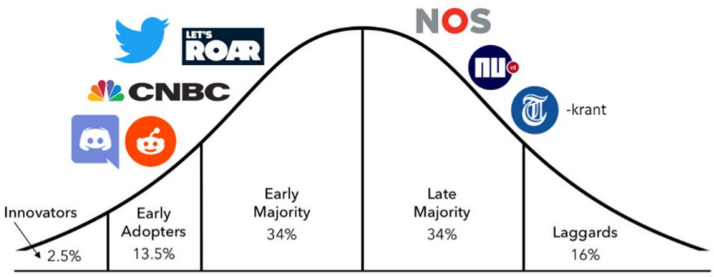
Sprint 1

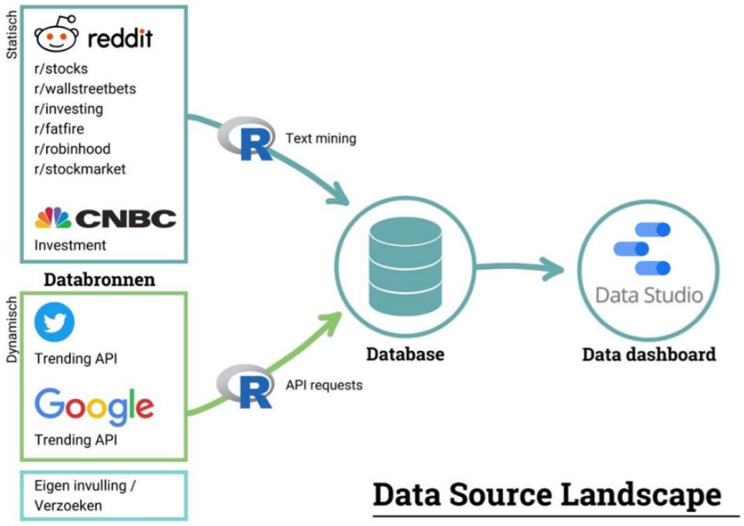
In deze sprint is de groep onderzoek gaan doen naar het onderwerp. Data en programmeren zijn beide topics die nog geen bekend terrein waren bij de groepsleden. Dit is versneld uitgevoerd middels cursussen van DataCamp en Google Analytics.

Tijdens de eerste sprint zijn er een aantal Google cursussen gevolgd en behaald. Aan de hand van deze cursussen is er voldoende startkennis op gedaan om voor Indicia aan de slag te gaan in Google Analytics en Google Data studio. De cursussen die zijn behaald zijn: Google Analytics voor beginners, Google Analytics voor gevorderden, Individuele kwalificatie voor Google Analytics en als laatste introduction to Data studio.

Als resultaat van sprint is er een data landscape opgesteld en is er geprobeerd inzicht te krijgen in de verschillende ‘nieuws’ bronnen en hun plek op de trend curve.

Trend curve topic- Investment





Sprint 2

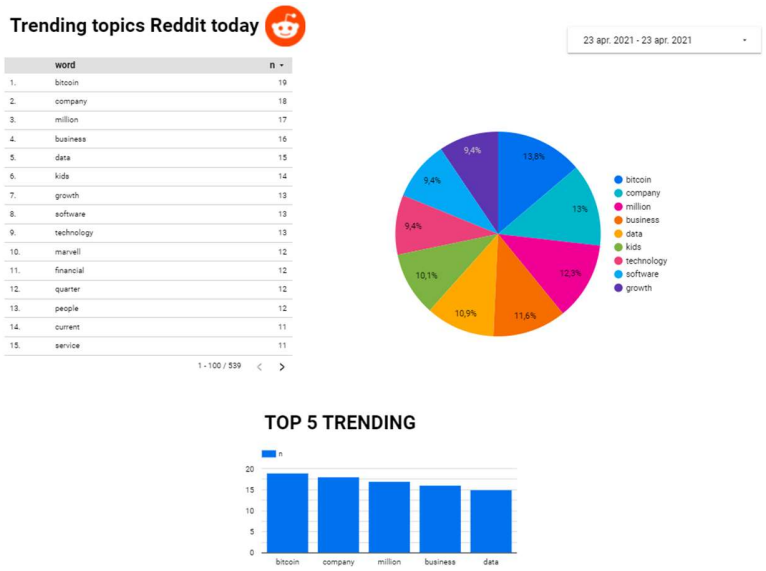
In sprint twee is er een eerste opzet gemaakt in R. Doormiddel van textmining in R hebben we webpagina's van Reddit kunnen downloaden en uitlezen. Hieruit konden we de irrelevante woorden filteren om zo relevante, veelgebruikte woorden te identificeren.

In deze stap zijn de volgende sub Reddits gebruikt:

- r/stocks
- r/wallstreetbets
- r/investing
- r/fatfire
- r/robinhood
- r/stockmarket

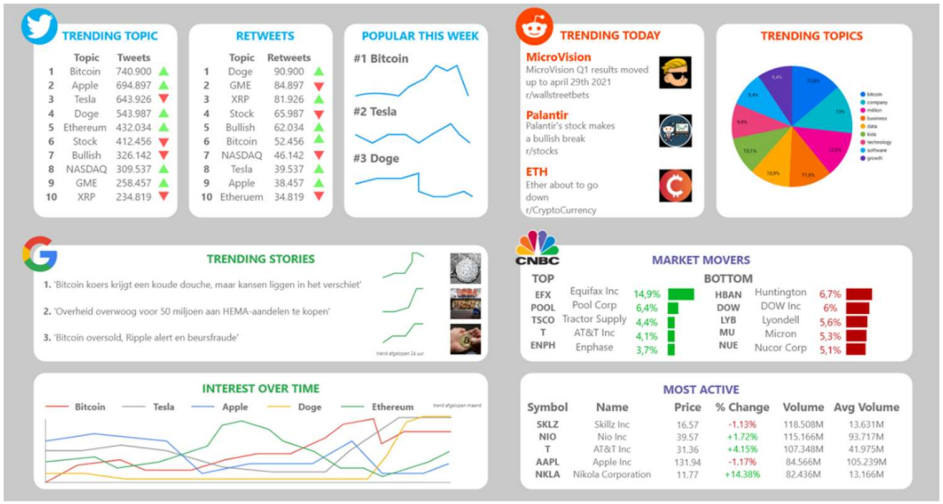
Google Data Studio

Om de verkregen data vanuit R goed weer te geven hebben we samen met de opdrachtgever besloten dit weer te geven in Google Data Studio. Deze is eenvoudig te koppelen met verschillende data types en geeft veel mogelijke filteropties. Toentertijd zag het data dashboard er nog uit zoals hieronder weergegeven. Dit was meer om een beeld te geven van hoe we het data dashboard voor ogen hadden en welke gegevens erin zouden staan.



Prototype Dashboard

Om wat houvast te creeëren bij het maken van een dashboard in Google data studio hebben we met behulp van het vak UX-design een prototype gemaakt die het gewenste resultaat weergeeft. Met inspiratie vanuit dit gewenste resultaat kunnen we het dashboard in Google data studio op gaan bouwen. Door het dashboard op de juiste manier in te richten speel je in op de wens van de gebruiker die in een blik wilt zien wat er momenteel trending is. De juiste data weergave methodes en kleur gebruik kunnen hier bij helpen.

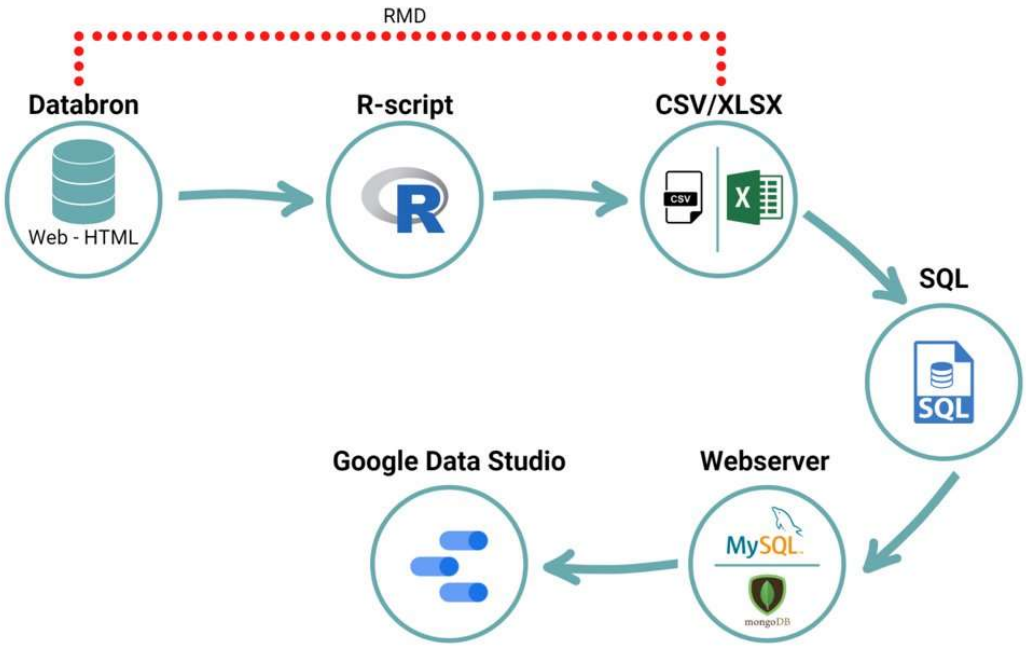


Google Analytics

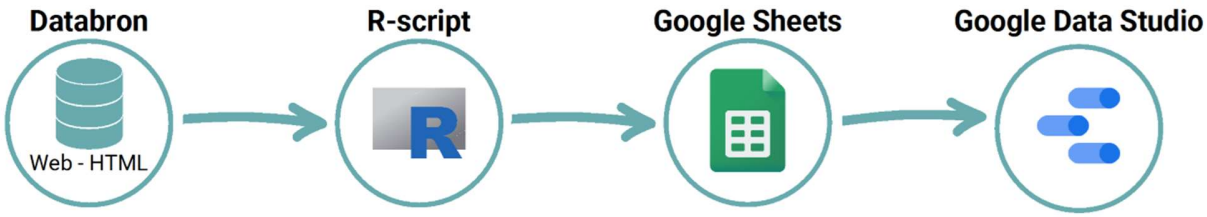
Twee groepsleden zijn in het Google Analytics account van Let's Roar gedoken. Analytics zou een toevoeging kunnen bieden aan het uiteindelijke Data Dashboard door bijvoorbeeld reactieve bezoekerspatronen weer te geven: wanneer er over een trending topic wordt geschreven, heeft dit dan invloed op de website bezoeken en conversie?

Sprint 3

Voor sprint 3 hebben we prioriteit geplaatst bij het automatiseren van de dataverzameling ten opzichte van het uitbreiden van de dataverzameling. Hierbij is onderzoek gedaan naar verschillende databases en mogelijke data verrijking met datum, tijd en bron. Aan het begin van deze sprint is er een versimpelde data process landscape ontworpen:



Achteraf gezien bleek deze al snel niet meer compleet. Binnen R hebben we namelijk gebruik gemaakt van de Googlesheets4 library, welke gebruik maakt van de Google Sheets V4 API. Zo kunnen we een Google Sheet (of meerdere) gebruiken als database, zonder hierbij een self-hosted database zoals MySQL en en het SQL framework te gebruiken. Het nieuwe te hanteren datalandschap ziet er dan als volgt uit:

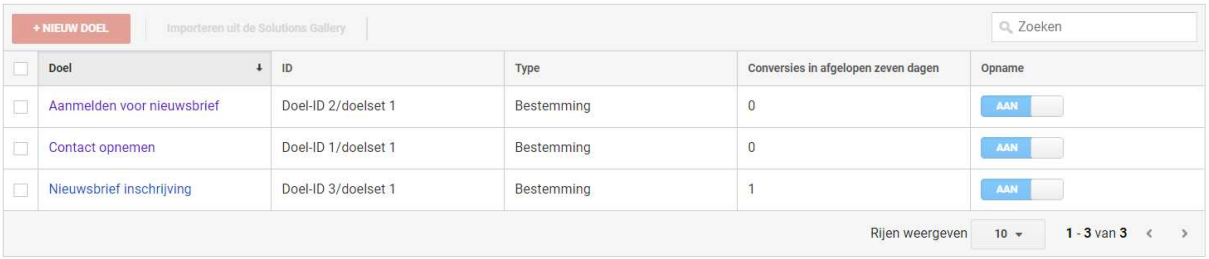


In de loop van sprint 3 is er in combinatie met R ook een start gemaakt met werken in GitHub, om zo eenvoudiger versies te beheren.

Doordat we alle data binnen Google Data Studio ophalen via een geautomatiseerde sheet in Google Sheets, word deze data om de 15 minuten vernieuwd. Hierdoor is de data die zichtbaar is in het data dashboard maximaal 15 minuten verouderd.

Google Analytics

In sprint 3 is er ook gewerkt met de websitedata van Let's Roar in Google Analytics. Helaas kregen wij niet de volledige toegang voor Google Analytics waardoor wij per mail moesten doorgeven wat er in Google Analytics gedaan moest worden. De bedoeling was om een drietal doelen op te stellen en daarvoor een funnel te maken aan de hand van de trechterweergave. Dit verliep niet vlekkeloos omdat er op het begin een aantal punten ontbraken in het conversiedoel en dan met name waardoor de funnel niet volledig zichtbaar was. Door de fout in het conversiedoel bleef de eindconversie na twee weken nog steeds op nul conversies staan.



Google Cloud platform

Om gebruik te maken van de Google Sheets API is er een Google Cloud service account nodig. Vanuit hier is het mogelijk om API tokens te genereren en te downloaden in een bestand. Dit bestand wordt gebruikt in het script om het lezen en schrijven te autoriseren. Voor nu is er gebruik gemaakt van het google account van de projectgroep, maar dit is om te zetten naar andere google accounts.

Op console.cloud.google.com moet een project worden aangemaakt met hierin de nodige gegevens van onder andere het doel van de applicatie. Links onder het IAM menu kan bij service accounts een nieuwe service account worden aangemaakt. Bij voorkeur wordt deze service account als 'Owner' ingesteld. Vervolgens heeft dit service account een mailadres aangemaakt. Noteer dit om hiermee later Sheets toegang te geven.

Bij de nieuw aangemaakte service account kan nu onder 'Actions' API keys worden gegenereerd bij 'manage keys'. Klik op 'Add key' en 'Create key'. In R maken we gebruik van een JSON bestand, dus deze kan hier worden gegenereerd. Sla vervolgens het bestand op in de map waar het R script staat. Het is belangrijk het bestandspad aan te passen in het script.

Vervolgens moet voor het Google account de Google Sheets API worden toegevoegd. Links in het menu onder het kopje 'API & Services' -> 'API library' kunnen nieuwe API's worden toegevoegd. Zoek op 'Sheets' en activeer de API.

Het project moet nog gekoppeld worden aan het gemaakte service account en de API 'scopes'. Dit kan links in het menu onder 'APIs & Services' -> '0Auth consent screen'. Er moet een external app worden ingesteld. Vul alleen de vereiste gegevens in. Bij stap 2 dienen de volgende scopes toegevoegd te worden:

- https://www.googleapis.com/auth/drive
- https://www.googleapis.com/auth/drive.file
- https://www.googleapis.com/auth/drive.readonly
- https://www.googleapis.com/auth/spreadsheets
- https://www.googleapis.com/auth/spreadsheets.readonly

Test users zijn niet nodig om in te stellen. Rond de app registratie af. Nu is het service account nodig om toegang te geven tot de ingestelde Google Sheets. Voor nu is er gebruik gemaakt van gsheets-link@lets-roar-dashboard-indicia.iam.gserviceaccount.com. Ga naar de ingestelde sheets en klik rechtsboven op de 'Share' knop. Plak het service account adres en zorg ervoor dat dit account kan bewerken, zonder dit account te 'notifyen'. Controleer vervolgens de link van de sheet door op delen te klikken (en ontvangers van de link bewerk toegang te geven). Deze links staan in het R script in de Google Sheets functies achter 'ss ='.

Sprint 4

In sprint 4 stonden de volgende punten centraal: optimaliseren van automatisering, overdraagbaarheid en toevoegen van databronnen.

Uitbreiden databronnen

Voor het uitbreiden van databronnen is er gekozen voor een uitbreiding naar Reddit Crypto, subreddits waar over cryptocurrencies in het algemeen gesproken wordt. Hierbij is gekeken naar aantal leden, post volume en relevantie. We hebben hierbij vermeden subreddits als r/Bitcoin te gebruiken om een bias vooraf te voorkomen.

- r/cryptocurrency
- r/crypto
- r/cryptomarkets
- r/cryptomoonshots
- r/cryptocurrencies
- r/altcoin

Twitter API

Een tweede uitbreiding van databron is gericht op de Twitter API. Er is een begin gemaakt met toegang van het Twitter Developer Platform en de nodige authenticatiestappen in R. Vervolgens is er getest met verschillende functies uit de library Rtweet. Om de toevoeging van deze databron waardevol te maken is er ook gekeken naar een juiste strategie om data te verzamelen en te verwerken.

We hebben vanuit het twitter account van FlyingFish gezorgt voor de benodigde authenticatie. Je hebt hier een twitter developer account nodig waarmee je de benodigde key en tokens kunt toevoegen aan je R script. Vanuit hier kunnen we door de library Rtweet tweets genereren van twitter. We zoeken door middel van search_twitter naar tweets gerelateerd aan "investing". Dit gaf naar onze mening de meest

geschikte data voor het dashboard. We filteren het aantal tweets nu op 150 woorden zodat we ongeveer elk kwartier een nieuw script kunnen uitdraaien. We filteren alleen tweets met de taal Engels. De teksten die hieruit voort zijn gekomen zijn uit elkaar gehaald, gefilterd, datum aan toegevoegd en worden geteld zodat ze visueel in het dashboard kunnen worden weergegeven.

Om gebruik te kunnen maken van de twitter API moesten we met een twitter accoun inloggen bij developer.twitter.com om vervolgens een aanvraag te doen voor een developer account. Hier hebben we vervolgens antwoorden gegeven op de vragen die werden voorgelegd. De antwoorden die we gaven waren gerelateerd aan ons en voor dit project. We hebben hier vervolgens een project aangemaakt waarna we de benodigde customer keys en Authentication tokens ontvingen. Deze kun je meermaals verversen, dit hebben we dan ook moeten veranderen in het R-script.

Optimaliseren script

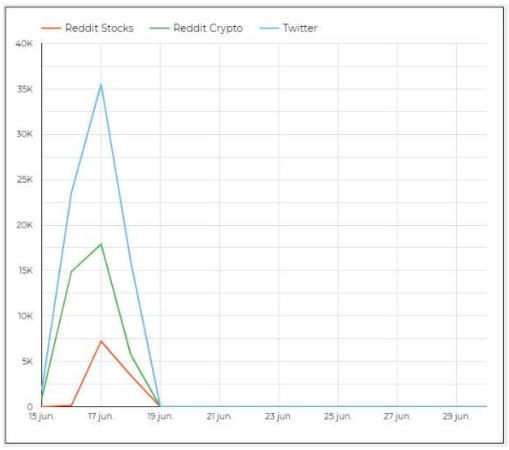
In een eerdere versie van het tekst mining script werd gebruik gemaakt van een CSV-bestand in de deduplicatie strategie. Voor het wegschrijven werd een vector gehanteerd, maar na het importeren werd hier een tabel van gemaakt. Dit gaf dus verschillende resultaten. R biedt een base functie welke een bestand wegschrijft en deze als zelfde R object terug in kan lezen; writeRDS en readRDS. Zit verkleint de foutmarge in dit deel van het script.

Om ervoor te zorgen dat de Google Sheets niet te vol komen is er een opschoon element verwerkt in het begin van het script. Dagelijks tussen 03:00 en 03:15 wordt de masterset gedownload uit Sheets en wordt opgeschoond op basis van datum. Deze datum en duur van databewaring is bepaald aan de hand van de limieten van Google Sheets. Dit limiet is 5 miljoen cellen (ongeacht kolommen en rijen). Hierdoor hebben we het aantal kolommen teruggebracht van 5 naar 3. Er is gekozen om de data 7 dagen te bewaren. Wanneer dit meer moet worden is het advies een MySQL of soortgelijke database te hanteren.

In de 4e week is er hard gewerkt met de verkregen data in Google Data Studio visueel weer te geven. Het doel was om in een grafiek per resource de trendlijn te zien van de afgelopen 7 dagen en dit is uiteindelijk gelukt. Dit was moeilijker dan verwacht omdat de trendlijn steeds alleen maar zichtbaar was van de data uit de Reddit aandelen. Ook kwamen er steeds fouten in de grafiek op het moment dat de data ververst werd. Dit is opgelost door de data in hetzelfde databestand steeds bij te werken.

Google Data Studio

Om alle drie de bronnen uiteindelijk weer te kunnen geven in de trendlijn grafiek is de data aan elkaar gekoppeld als één databundel. In Google Data Studio is het mogelijk om meerdere gegevens aan elkaar te koppelen en uiteindelijk weer te kunnen geven als ‘gemengde gegevens’. Toen dit was gebeurd lukte het nog steeds niet om alle drie de databronnen weer te geven, dit kwam omdat er maar één statistiek was toegevoegd. Door alle drie de Record Counts toe te voegen aan de statistiek was het vervolgens gelukt om alle drie de trendlijnen weer te geven. Nu was het alleen nog noodzakelijk om te kijken welke trendlijn bij welke databron hoorde en er vervolgens een naam en kleur aan te geven.



Het is nu mogelijk om zelf een specifieke periode te kiezen en vervolgens daaruit te zoeken op een woord. Bij het zoeken naar woorden is het mogelijk om uit 5 verschillende zoekfilters te kiezen: Gelijk aan, Bevat, Begint met, Regex en In.

Livedashboard:https://datastudio.google.com/embed/reporting/3639f603-2e5c-48e6-903f-1b558b58b889/page/gV6PC

Volledig bestand: https://datastudio.google.com/s/hxU9My7P0Io

Aanbevelingen

Script

Gaandeweg hebben we ondervonden dat de tekst mining niet altijd even betrouwbaar is. Het is een mogelijkheid de HTML tekst mining te vervangen met de Reddit API, om zo rechtstreeks de nodige posts binnen te halen.

Verbreding van de data waarde

Op dit moment worden er platte wordcounts toegepast in het script. Als de gebruiker op de hoogte is van de context en bronnen kunnen hier conclusief uit getrokken worden. De data kan wel waardevoller worden door bijvoorbeeld een Bag-of-words principe toe te passen.

Daarnaast is het een mogelijke uitbreiding om in het dashboard de rechtstreekse bronnen aan woorden te koppelen; als men op een word klikt kun je doorklikken op de posts/tweets waar deze worden uit gevonden worden.

Als derde aanbeveling op het gebied van de waarde van de data kan er gebruikt gemaakt worden van huidige Google Analytics of Hubspot data, om zo reactiepatronen vast te stellen en te verwerken in het data dashboard.

Als laatste kunnen er natuurlijk altijd bronnen bijgevoegd worden, zoals meer subreddits en andere Twitter listening kanalen. Op breder gebied kan er ook uitgebreid worden naar andere platformen om zo een breder vangnet te creëren.

Database

Als database hebben we Google Sheets gebruikt. In onze laatste versie van het script hebben we voor een databewaring van 7 dagen gekozen. Deze bleek achteraf wat terughoudend en kan vrij snel opgeschroefd worden naar 14 of 21 dagen. Mocht er in de toekomst behoefte zijn aan een langere bewaring kan er gekeken worden naar een andere type database zoals een SQL of BugQuery.

Bronnenlijst

https://developers.google.com/identity/protocols/oauth2/scopes#sheets

https://googlesheets4.tidyverse.org/

https://www.oreilly.com/library/view/mining-the-social/9781449368180/ch01.html

https://developer.twitter.com/en/docs/twitter-api/v1/trends/locations-with-trending-topics/api-reference/get-trends-available

https://developer.twitter.com/en/docs/tutorials/getting-started-with-r-and-v2-of-the-twitter-api

https://www.earthdatascience.org/courses/earth-analytics/get-data-using-apis/use-twitter-api-r/

https://cran.r-project.org/web/packages/rtweet/vignettes/auth.html

https://www.rdocumentation.org/packages/rtweet/versions/0.7.0/topics/search_tweets