

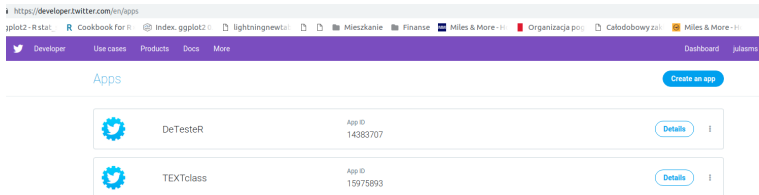
Eksploracja danych i wyszukiwanie informacji w mediach społecznościowych

Wykład 9-10 – analiza danych on-line

dr inż. Julian Sienkiewicz

13 listopada 2019

Nie da się ukryć, że zawsze najciekawsze dane dotyczą naszej aktywności w mediach społecznościowych. Jeszcze dwa lata temu można było dostać się bez większych problemów do wielu danych – obecnie w efekcie RODO oraz after typu Cambridge Analytica jest to **bardzo okrojone**. Z tego też powodu lepiej raczej korzystać z Twittera niż z Facebooka...



Aby skorzystać z danych Twittera niezbędne jest utworzenie aplikacji na własnym koncie. Dzięki temu otrzymamy klucz API i będziemy mogli swobodnie używać takich aplikacji jak `rtweet` pod R.

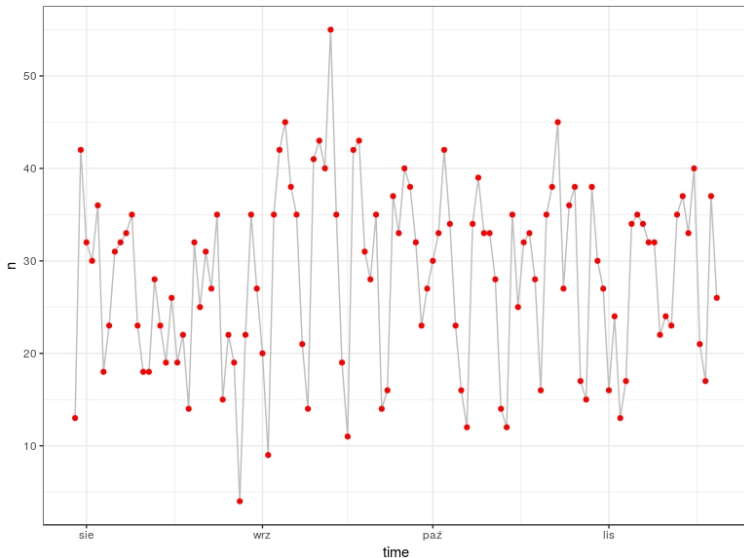
Poniżej zrzut z prostego timeline'u konta Twittera Gazety Wyborczej

	user_id	status_id	created_at	screen_name	text	source	d
gazeta_wyborcza.1	19179390	1064896545261277185	2018-11-20 15:01:35	gazeta_wyborcza	Prokuratorzy nie mogą uczyć dzieci o konstytucji htt...	IFTTT	
gazeta_wyborcza.2	19179390	1064887886561099782	2018-11-20 14:27:11	gazeta_wyborcza	Wniosek Brudzińskiego oddalony. Sąd nie zgodził się...	IFTTT	
gazeta_wyborcza.3	19179390	1064867469544833024	2018-11-20 13:06:03	gazeta_wyborcza	Wyborcza sprawdza co daje #wymaginanawspól...	IFTTT	
gazeta_wyborcza.4	19179390	1064867112198524928	2018-11-20 13:04:38	gazeta_wyborcza	Jaroslav Kulisz w @clouapp: Niech się ci stary wysz...	Twitter Web Client	
gazeta_wyborcza.5	19179390	1064866351108493315	2018-11-20 13:01:37	gazeta_wyborcza	Potężny krater uderzeniowy pod lodem Grenlandii. ...	IFTTT	
gazeta_wyborcza.6	19179390	1064860978733551616	2018-11-20 12:40:16	gazeta_wyborcza	W cyklu co daje #wymaginanawspólnota przysz...	Twitter Web Client	
gazeta_wyborcza.7	19179390	1064852288437710848	2018-11-20 12:05:44	gazeta_wyborcza	Beata Mazurek: Nie poprzemy wniosku o komisję śle...	IFTTT	
gazeta_wyborcza.8	19179390	1064850886504185856	2018-11-20 12:00:09	gazeta_wyborcza	Wdowa wychłostała się sama, czyli antynowiczok [C...	IFTTT	
gazeta_wyborcza.9	19179390	1064842092550197248	2018-11-20 11:25:13	gazeta_wyborcza	Zanikający antysemityzm w Polsce? Specjaliści: Cz...	IFTTT	
gazeta_wyborcza.10	19179390	1064838410177847296	2018-11-20 11:10:35	gazeta_wyborcza	Jak PiS tłumaczy się z nepotyzmu? Podobnie jak PSL...	IFTTT	
gazeta_wyborcza.11	19179390	1064833383740698624	2018-11-20 10:50:36	gazeta_wyborcza	TVP Info broni ambasadora Przylebskiego. Ale my m...	IFTTT	
gazeta_wyborcza.12	19179390	1064830868433371138	2018-11-20 10:40:37	gazeta_wyborcza	RMF FM: CBA w mieszkaniu Marka Chrzanowskiego ...	IFTTT	
gazeta_wyborcza.13	19179390	1064818205074571264	2018-11-20 09:50:18	gazeta_wyborcza	W Hongkongu najważniejszy proces polityczny od cz...	IFTTT	
gazeta_wyborcza.14	19179390	1064813246731956225	2018-11-20 09:30:35	gazeta_wyborcza	SKOK-I kontra "Wyborcza". Kasa Krajowa zapowiada ...	IFTTT	
gazeta_wyborcza.15	19179390	1064791839889342464	2018-11-20 08:05:32	gazeta_wyborcza	Dziś Polska zmierzy się z Portugalią. Na boisku zabra...	IFTTT	
gazeta_wyborcza.16	19179390	1064791830267613184	2018-11-20 08:05:29	gazeta_wyborcza	#Repost @andrzejrysuj • • • • • Dziś skarbowka. ...	IFTTT	
gazeta_wyborcza.17	19179390	1064762862881701888	2018-11-20 06:10:23	gazeta_wyborcza	Strzelanina w szpitalu w Chicago. Cztery osoby nie z...	IFTTT	

Generalnie taki pakiet umożliwia nam:

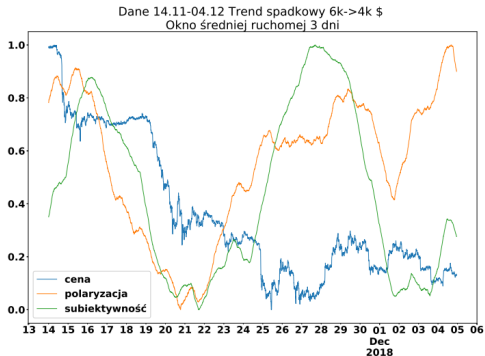
- wyszukiwanie dowolnego łańcucha (6-9 ostatnich dni),
- pobieranie informacji o liczbie retweetów etc,
- obserwacje aktywności itp.

Poniżej aktywność konta TT GW na przestrzeni ostatnich paru miesięcy



Przewidywanie rynku kryptowalut za pomocą analizy sentymentu mediów społecznościowych

inż. Mikołaj Koszowski (praca inżynierska 2018/19)



Przewidywanie wyników sondaży wyborczych za pomocą metod uczenia maszynowego

Robert Jankowski (praca inżynierska 2019/20)



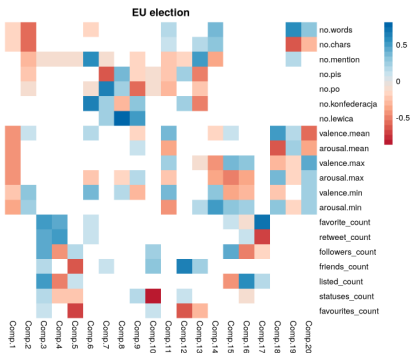
Przewidywanie wyników sondaży wyborczych za pomocą metod uczenia maszynowego

Robert Jankowski (praca inżynierska 2019/20)



Przewidywanie wyników sondaży wyborczych za pomocą metod uczenia maszynowego

Robert Jankowski (praca inżynierska 2019/20)



Google

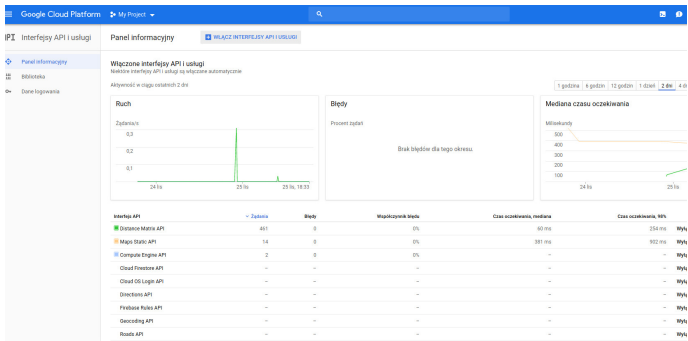
Jak wszyscy wiemy, **Google** jest prawdziwym potentatem i w zasadzie monopolistą jeśli chodzi o szereg usług – możemy albo z nich nie korzystać albo też zgadzać się na różne, delikatnie mówiąc, średnio etyczne praktyki [JS: mój osobisty pogląd]. Nie da się jednak ukryć, że poniższe usługi, do których dedykowane są konkretne API są dość przydatne:

- Distance Matrix,
- Maps Static,
- Directions,
- Geocoding,
- Maps



Konsola i pakiety R

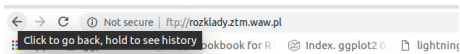
Od pewnego czasu Google wymaga udostępnienia danych związanych z płatnościami, ale ww. usługi są dalej darmowe (do osiągnięcia pewnych limitów zapytań / transferu). Konsola Google Cloud wygląda np. następująco



Za pomocą konsoli jesteśmy w stanie ustawić **klucze API**, które następnie można wykorzystać w takich pakietach R jak: `ggmap`, `gmapdistance`, `googleway`.

A detailed map of Warsaw, Poland, showing the Vistula River (Wisła) flowing through the city. Major roads and highways are marked with numbers in yellow and red. Landmarks such as the Museum of the History of Warsaw (Muzeum Historii Warszawy), the Copernicus Science Centre (Centrum Nauki Kopernik), and the Park Łazienkowski are labeled. The map includes latitude and longitude coordinates, with latitude ranging from 52.20 to 52.28 and longitude from 20.90 to 21.10. The map is credited to Google Maps.

Jednym z dostawców ciekawych danych jest ZTM Warszawa. Na serwrze <ftp://rozklady.ztm.waw.pl> Udostępnia on spakowane pliki, zawierające informacje związane z poszczególnymi przystankami (oraz rozkładami jazdy).



Index of /

Name	Size	Date Modified
☐ RA141202.7z	4.7 MB	12/2/14, 1:00:00 AM
☐ RA141203.7z	7.7 MB	12/3/14, 1:00:00 AM
☐ RA141204.7z	4.8 MB	12/4/14, 1:00:00 AM
☐ RA141205.7z	4.7 MB	12/4/14, 1:00:00 AM
☐ RA141206.7z	4.8 MB	12/4/14, 1:00:00 AM
☐ RA141207.7z	4.8 MB	12/4/14, 1:00:00 AM
☐ RA141208.7z	4.7 MB	12/5/14, 1:00:00 AM
☐ RA141209.7z	4.8 MB	12/5/14, 1:00:00 AM
☐ RA141210.7z	7.7 MB	12/8/14, 1:00:00 AM
☐ RA141211.7z	4.8 MB	12/11/14, 1:00:00 AM
☐ RA141212.7z	4.8 MB	12/11/14, 1:00:00 AM
☐ RA141213.7z	4.7 MB	12/11/14, 1:00:00 AM
☐ RA141214.7z	4.8 MB	12/11/14, 1:00:00 AM

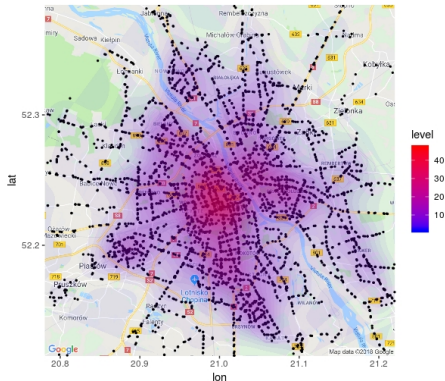
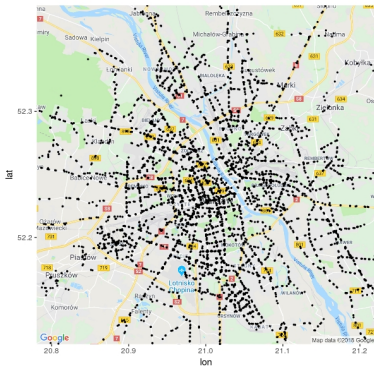


Dane nie są w szczególnie wygodnym formacie – plik tekstowy ze słabo wyróżnionymi sekcjami.

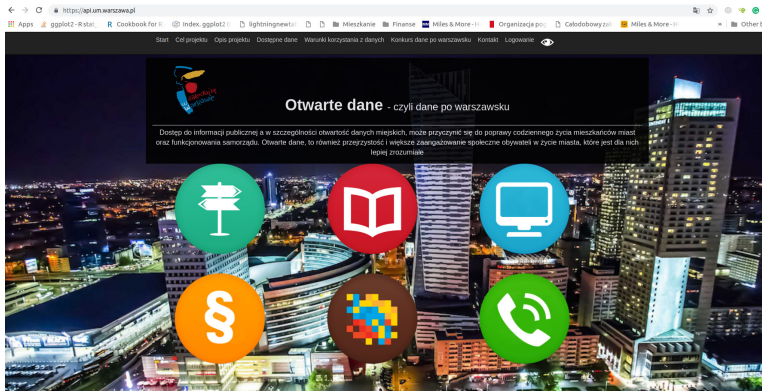
1001	KIJOWSKA,			9			--			WARSZAWA											
#PR																					
100101	2	UL./PL.: TARGOWA,								Kier.: AL. ZIELENIECKA,		Y= 52.248670	X= 21.044260								
L 6	-	stały:	125	135	138	166	509	517													
L 1	-	na zadanie:	N21																		
100102	2	UL./PL.: TARGOWA,								Kier.: ZABKOWSKA,		Y= 52.249020	X= 21.044540								
L 6	-	stały:	125	135	138	166	509	517													
L 1	-	na zadanie:	N21																		
100103	1	UL./PL.: TARGOWA,								Kier.: AL. ZIELENIECKA,		Y= 52.248998	X= 21.043983								
L 9	-	stały:	3	6	7	8^	9^	22 24^ 25 26													
100104	1	UL./PL.: TARGOWA,								Kier.: DW. WILENSKI,		Y= 52.249905	X= 21.041726								
L 8	-	stały:	3	6	13	23^	25	26 27^ 28													
100105	2	UL./PL.: KIJOWSKA,								Kier.: AL. ZIELENIECKA,		Y= 52.250350	X= 21.043860								
L 3	-	stały:	123	146	147																
L 3	-	na zadanie:	N02	N03	N71																
100106	1	UL./PL.: KIJOWSKA,								Kier.: DW. WSCHODNI (KIJOWSKA),		Y= 52.250008	X= 21.043710								
L 13	-	stały:	3^	6^	7	8^	9^	13 22 23^ 24^ 25^	26^	27											
100107	2	UL./PL.: KIJOWSKA,								Kier.: ZABKOWSKA,		Y= 52.250210	X= 21.043630								
L 3	-	stały:	120	156	169																
L 3	-	na zadanie:	N02	N03	N71																
100108	2	UL./PL.: KIJOWSKA,								Kier.: DW. WSCHODNI (KIJOWSKA),		Y= 52.250060	X= 21.044240								
L 6	-	stały:	120	123	146	147	156	169													
L 3	-	na zadanie:	N02	N03	N71																
100188	1	UL./PL.: KIJOWSKA,								Kier.: DW. WSCHODNI (KIJOWSKA),		Y= 52.249820	X= 21.043890								
L 3	-	na zadanie:	N02	N03	N71																
#PR																					
1002	ZABKOWSKA,			5			--			WARSZAWA											
#PR																					
100201	2	UL./PL.: TARGOWA,								Kier.: KIJOWSKA,		Y= 52.251300	X= 21.038540								
L 9	-	stały:	120	125	135	138	156	166	169 509 517												
L 4	-	na zadanie:	N02	N03	N21	N71															
100202	2	UL./PL.: TARGOWA,								Kier.: DW. WILENSKI,		Y= 52.251720	X= 21.038640								
L 9	-	stały:	120	125	135	138	156	166	169 509 517												
L 4	-	na zadanie:	N02	N03	N21	N71															
100203	1	UL./PL.: TARGOWA,								Kier.: KIJOWSKA,		Y= 52.251755	X= 21.038137								
L 8	-	stały:	3	6	13	23^	25	26 27^ 28													
100204	1	UL./PL.: TARGOWA,								Kier.: DW. WILENSKI,		Y= 52.251923	X= 21.037890								
L 8	-	stały:	3	6	13	23^	25	26 27^ 28													
100206	2	UL./PL.: TARGOWA,								Kier.: DW. WILENSKI,		Y= 52.253178	X= 21.036700								
L 3	-	stały:	162	170	338																
L 4	-	na zadanie:	N11	N14	N61	N64															
#PR																					

Jednak już wykorzystanie prostych wyrażeń regularnych umożliwia uzyskania informacji nt. numeru przystanku oraz jego położenia (współrzędne geograficzne).

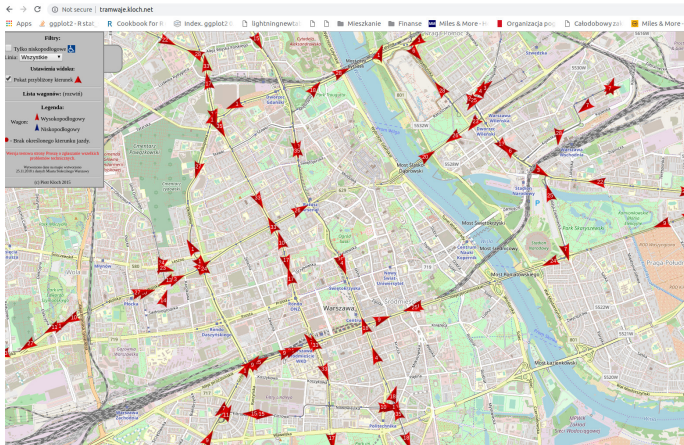
Korzystając z API Google można łatwo zwizualizować położenie przystanków oraz ich gęstość:



Miasto Warszawa prowadzi pod adresem api.um.warszawa.pl prowadzi serwis **Otwarte dane**, w którym możliwe jest otrzymanie klucza API i dostęp do wielu interesujących danych, np. **położeń tramwajów w czasie rzeczywistym**.



Oczywiście dane są wykorzystywane przez zewnętrzne serwisy, np.
tramwaje.koch.net.



My chcemy zobaczyć, jakie informacje da się “wyciągnąć” z tych danych.

Kod (tutaj R) służący do pobrania danych jest dość prosty:

```
rm(list = ls())
## API UM do oznaczenia pozycji tramwajow w Warszawie
library(jsonlite)
library(dplyr)

get.trams <- function(i, steps, delay, url, api) {

  cat("Step ", i, "of", steps, "Waiting ", delay, "secs.\n")
  Sys.sleep(delay)
  url.api <- paste(url, api, sep = "")
  x <- fromJSON(url.api)

  return(x$result %>% mutate(call = i))
}

um.waw.api <- "https://api.um.warszawa.pl/api/action/wsstore_get?id=c7238cfe-8b1f-4c38-bb4a-de386db7e776&apikey="
um.waw.url <- "https://api.um.warszawa.pl/api/action/wsstore_get?id=c7238cfe-8b1f-4c38-bb4a-de386db7e776&apikey="
delay <- 60 # sekundy
steps <- 30

data.trams <- do.call(rbind, lapply(1:steps, function(i) get.trams(i, steps, delay, um.waw.url, um.waw.api)))
```

W efekcie można otrzymać np. następującą strukturę:

```
> data.trams %>% print(n=20)
# A tibble: 6,770 x 9
  Status FirstLine Lon Lines Time Lat LowFloor Brigade call
  <chr> <int> <dbl> <chr> <chr> <dbl> <lgl> <int> <int>
1 RUNNING 6 20.9 "6" "2018-11-24T14:35:43" 52.3 FALSE 2 1
2 RUNNING 28 21.1 "28" "2018-11-24T14:34:06" 52.3 FALSE 3 1
3 RUNNING 6 21.0 "6" "2018-11-24T14:35:43" 52.3 FALSE 4 1
4 RUNNING 6 21.1 "6" "2018-11-24T14:35:39" 52.2 FALSE 8 1
5 RUNNING 27 20.9 "27" "2018-11-24T14:35:17" 52.2 FALSE 2 1
6 RUNNING 28 21.0 "28" "2018-11-24T14:35:41" 52.3 FALSE 4 1
7 RUNNING 27 21.0 "27" "2018-11-24T14:35:44" 52.2 FALSE 5 1
8 RUNNING 27 21.0 "27" "2018-11-24T14:35:41" 52.2 FALSE 3 1
```

Dane

Dane zawierają informacje dotyczące następujących pól:

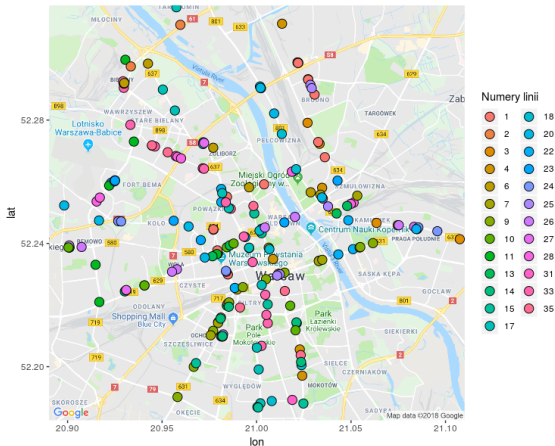
- numer linii,
- numer brygady (czyli tak jakby podlinia),
- czas, w którym dokonano lokalizacji,
- położenie (długość i szerokość geograficzną)

Widok poszczególnych brygad, kolorem oznaczono linie

Dane

Dane zawierają informacje dotyczące następujących pól:

- numer linii,
- numer brygady (czyli tak jakby podlinia),
- czas, w którym dokonano lokalizacji,
- położenie (długość i szerokość geograficzną)



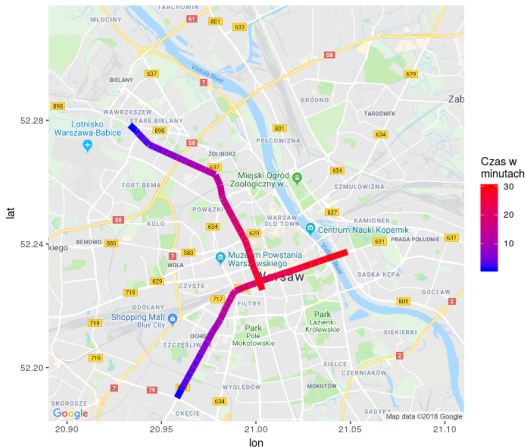
Dane

- można również ocenić jak zmienia się system w czasie,
- próbkujemy układ z częstotliwością jednej minuty,
- na wykresie przedstawiono pojedynczą brygadę dla linii 7 oraz 33

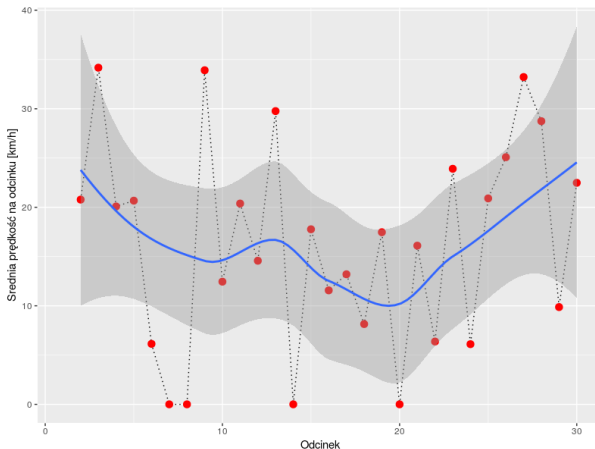
Linie 7 i 33, kolorem oznaczono położenie po x minutach od wykonania pierwszego zapytania

Dane

- można również ocenić jak zmienia się system w czasie,
- próbujemy układ z częstotliwością jednej minuty,
- na wykresie przedstawiono pojedynczą brygadę dla linii 7 oraz 33



Korzystając z Google Distance Matrix API można wyznaczyć odległości pomiędzy kolejnymi położeniami, co w połączeniu z informacją o czasie przejazdu daje nam prędkość (średnią)



Tak wyznaczone wartości można też nanieść na mapę, pokazując w których miejscach tramwaj zwalnia.



W ostatnich latach kluczowym problem staje się kwestia zanieczyszczenia powietrza, a dokładniej tzw. **smogu**. Jest to wypadkowa związana z przestrzałami urządzeniami grzewczymi w domach oraz transportem indywidualnym. Od ok. roku działa platforma airly.eu, która udostępnia dane dotyczące poziomów zanieczyszczenia powietrza w wybranych punktach Polski.

airly

Strona główna Mapa Czujnik Dane O Airly Blog Kariera Projekty Kontakt EN

Czy wiesz czym oddychasz na co dzień?

Budujemy sieć sensorów powietrza dla miast i gmin. Dzięki naszemu rozwiązaniu możliwe jest monitorowanie stanu powietrza w czasie rzeczywistym za pomocą mapy online.

Naszym celem jest zbudowanie gęstej sieci sensorów w całej Polsce, co wpłynie na zwiększenie świadomości tego, czym oddychamy na co dzień. Pozwoli to także zidentyfikowanie źródeł problemów i ich dokładnej lokalizacji.

[ZOBACZ NASZ FILM](#)

Sprawdź jakość powietrza w twoim otoczeniu!

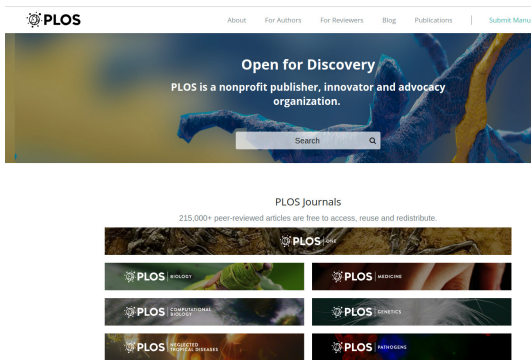
[PRZEJDŹ DO MAPY](#)

[APLIKACJA NA ANDROIDA](#)

[APLIKACJA NA IOS](#)

Airly posiada API (konieczna rejestracja), za pomocą którego można dostać się do obecnych danych oraz historycznych (24 godziny).

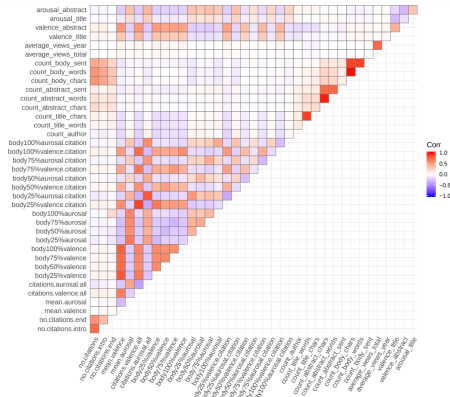
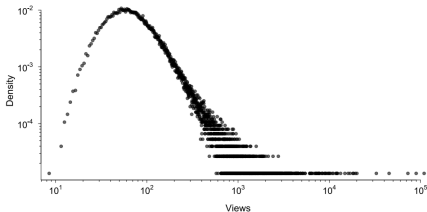
PLOS - **Public Library of Science** to zbiór dość poczytanych czasopism w standardzie OA – Open Access. Do obecnej chwili w ramach swoich kilku czasopism opublikowana tam ponad **215 tysięcy artykułów naukowych**.



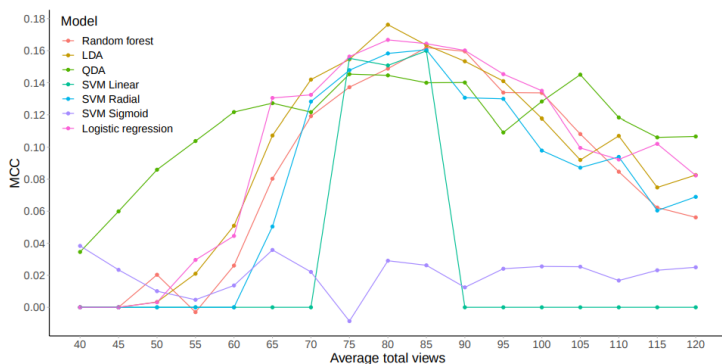
Dostępne są biblioteki w R: `rplos` oraz `alm`, które dają pełny dostęp do wszystkich artykułów oraz statystyk związanych z nimi.

Analiza artykułów naukowych PLOS ONE

Robert Jankowski (ION)



$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$



Open Academic Graph

Projekt studencki

Open Academic Graph 2019

[Overview](#)[Venue Collection](#)[Paper Collection](#)[Author Collection](#)[OAG v2 Download](#)[Data Description](#)[Evaluation](#)[Reference](#)

Overview

Open Academic Graph (OAG) is generated by linking two large academic graphs: [Microsoft Academic Graph \(MAG\)](#) and [AMiner](#). This page collects the second version of OAG which was generated in Jan 2019. The statistics of OAG 2019 is listed as the three tables below. The two large graphs are both evolving and we take MAG November 2018 snapshot and AMiner July 2018 or January 2019 snapshot for this version.

Venue Collection

Table 1: statistics of OAG venue data

Data set	#Pairs/Venues	Date
Linking relations	29,841	2018.12
AMiner venues	89,397	2018.07
MAG venues	52,678	2018.11

Paper Collection

Table 2: statistics of OAG paper data

Data set	#Pairs/Papers	Date
Linking relations	91,137,597	2018.12
AMiner papers	172,209,563	2019.01
MAG papers	208,915,369	2018.11