

Eksploracja danych i wyszukiwanie informacji w mediach społecznościowych

Wykład 7 - analiza danych on-line

dr inż. Julian Sienkiewicz

26 listopada 2018

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...

The screenshot shows the Twitter Developer website at <https://developer.twitter.com/en/apps>. The top navigation bar includes links for Rplot2-Rstat, Cookbook for R, Index, ggplot2, lightningnewtai, Mieszkanie, Finanse, Miles & More - H, Organizacja poziomu, Celodobowy załącznik, Miles & More - H, Developer, Use cases, Products, Docs, More, Dashboard, and julasms. Below the navigation is a purple header with tabs for Apps, Developers, and Create an app. The main content area displays two registered applications:

App Name	App ID	Actions
DeTesterR	14383707	Details
TEXTclass	15975893	Details

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.

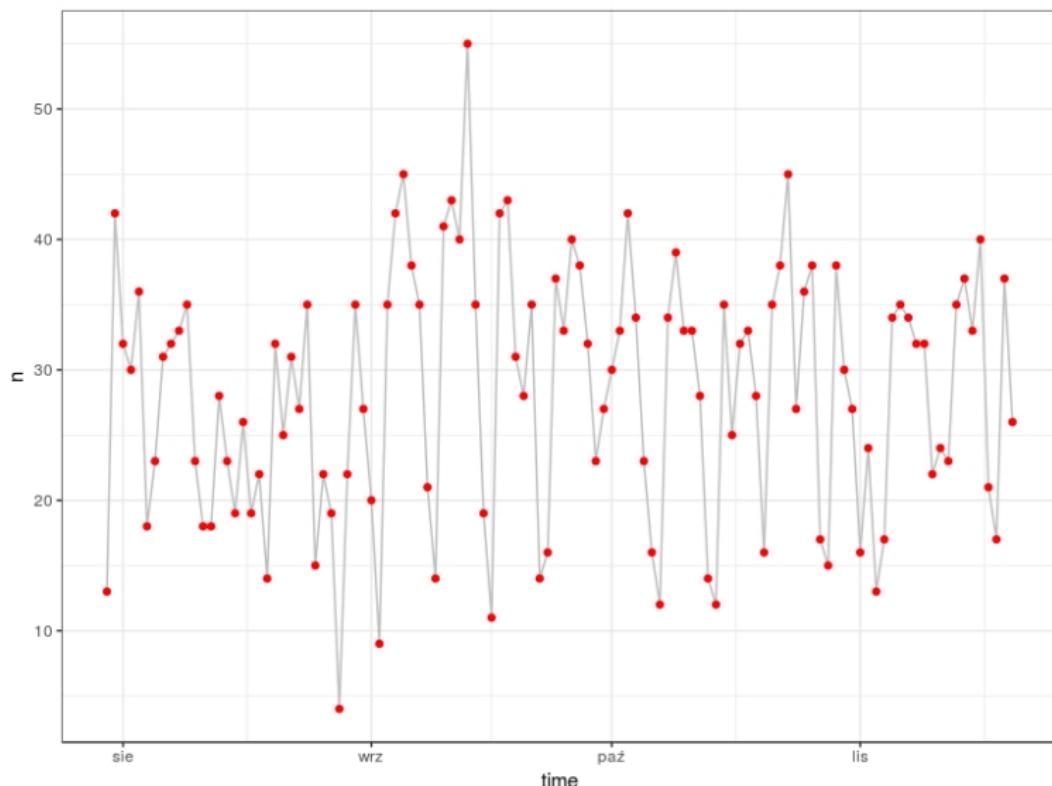
Poniżej zrzut z prostego timeline'u konta Twittera Gazety Wyborczej

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 ht...	IFTTT
gazeta_wyborcza.2	19179390	106488788651099782	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 #wyimaginowanawspół... Jarosław Kuisz w @clouapp: Niech się ci starzy wysz...	IFTTT
gazeta_wyborcza.4	19179390	1064867112198524928	2018-11-20 13:04:36	gazeta_wyborcza	Potężny krater uderzeniowy pod lodem Grenlandii ... Twitter Web Client	
gazeta_wyborcza.5	19179390	1064866351108493315	2018-11-20 13:01:37	gazeta_wyborcza	W cyku co daje #wyimaginowanawspółnota przysz... Twitter Web Client	
gazeta_wyborcza.6	19179390	1064860978733551616	2018-11-20 12:40:16	gazeta_wyborcza	Beata Mazurek: Nie poprzemy wniosku o komisję śle...	IFTTT
gazeta_wyborcza.7	19179390	1064852288437710848	2018-11-20 12:05:44	gazeta_wyborcza	Wdowa wychłostała się sama, czyli antynowiczok [C... IFTTT	
gazeta_wyborcza.8	19179390	1064850886504185856	2018-11-20 12:00:09	gazeta_wyborcza	Zanikający antysemityzm w Polsce? Specjalisci: Czę... IFTTT	
gazeta_wyborcza.9	19179390	1064842092550197248	2018-11-20 11:25:13	gazeta_wyborcza	Jak PiS tłumaczy się z nepotyzmu? Podobnie jak PSL... IFTTT	
gazeta_wyborcza.10	19179390	1064838410177847296	2018-11-20 11:10:35	gazeta_wyborcza	TVP Info broni ambasadorka Przyłębskiego. Ale my m... IFTTT	
gazeta_wyborcza.11	19179390	1064833383740698624	2018-11-20 10:50:36	gazeta_wyborcza	RMP FM: CBA w mieszkaniu Marka Chrzanowskiego ... IFTTT	
gazeta_wyborcza.12	19179390	1064830868433371138	2018-11-20 10:40:37	gazeta_wyborcza	W Hongkongu najważniejszy proces polityczny od cz... IFTTT	
gazeta_wyborcza.13	19179390	1064818205074571264	2018-11-20 09:50:18	gazeta_wyborcza	SKOK-i kontra "Wyborcza". Kasa Krajowa zapowiada ... IFTTT	
gazeta_wyborcza.14	19179390	1064813246731956225	2018-11-20 09:30:35	gazeta_wyborcza	Dziś Polska zmierzy się z Portugalią. Na boisku zabra.. IFTTT	
gazeta_wyborcza.15	19179390	1064791839889342464	2018-11-20 08:05:32	gazeta_wyborcza	#Repost @andrzejrysuje • • • • Dziś skarówka. ... IFTTT	
gazeta_wyborcza.16	19179390	1064791830267613184	2018-11-20 08:05:29	gazeta_wyborcza	Strzelanina w szpitalu w Chicago. Cztery osoby nie z... IFTTT	
gazeta_wyborcza.17	19179390	1064762862881701888	2018-11-20 06:10:23	gazeta_wyborcza		

Generalnie taki pakiet umożliwia nam:

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

Poniżej aktywność konta TT GW na przestrzeni ostatnich paru miesięcy



Google

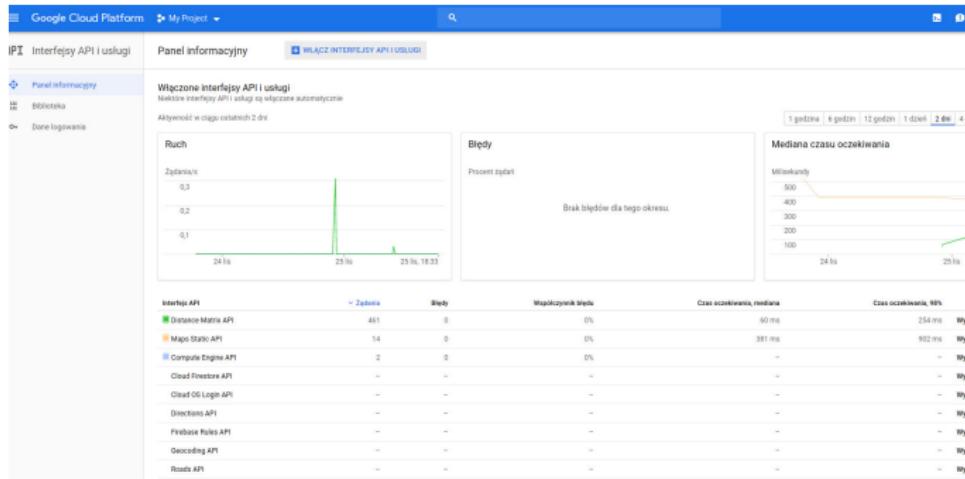
Jak wszyscy wiemy, **Google** jest prawdziwym potentatem i w zasadzie monopolistą jeśli chodzi o szereg usług – możemy albo z nich nie korzystać albo też zgadzać się na różne, delikatnie mówiąc, średnio etyczne praktyki [JS: mój osobisty pogląd]. Nie da się jednak ukryć, że poniższe usługi, do których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



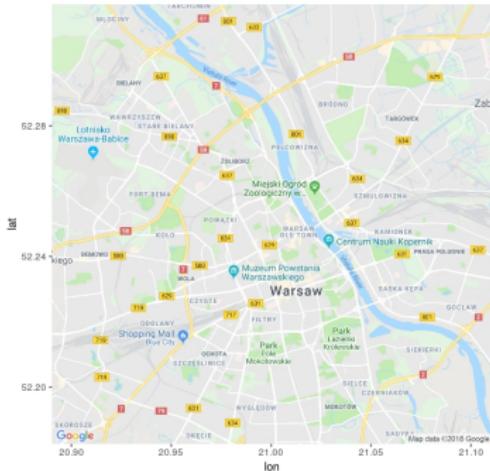
Za pomocą konsoli jesteśmy w stanie ustawić **klucze API**, które następnie można wykorzystać w takich pakietach R jak: ggmap, gmapdistance, googleway.

Konsola i pakiety R

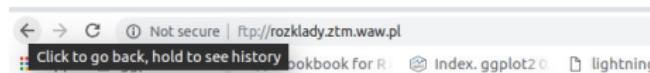
W efekcie poniższy prosty kod

```
library(ggmap)  
register_google("REDACTED")  
map <- get_googlemap(center = c(lon = 21.00, lat = 52.25), zoom = 12, maptype = "road")  
ggmap(map)
```

umożliwa otrzymanie prostej mapy, na której można nanosić następne informacje.



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



Dane

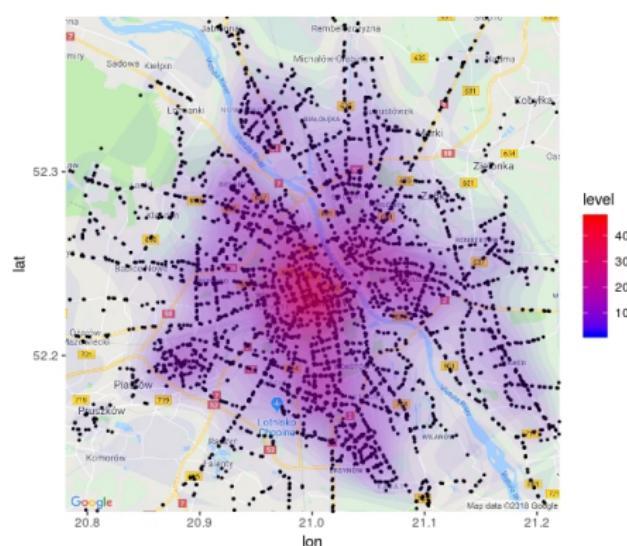
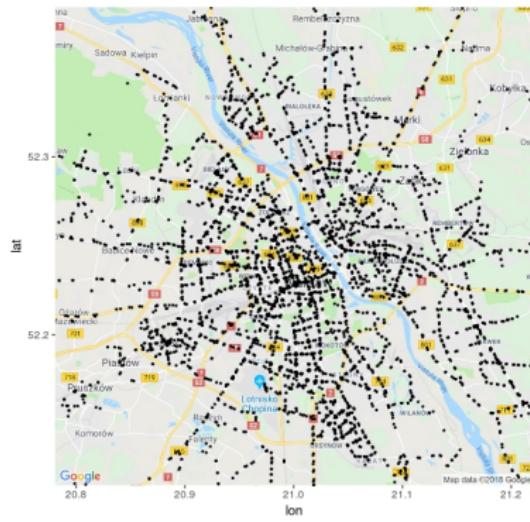
Dane nie są w szczególnie wygodnym formacie – plik tekstowy ze słabo wyróżnionymi sekcjami.

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

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

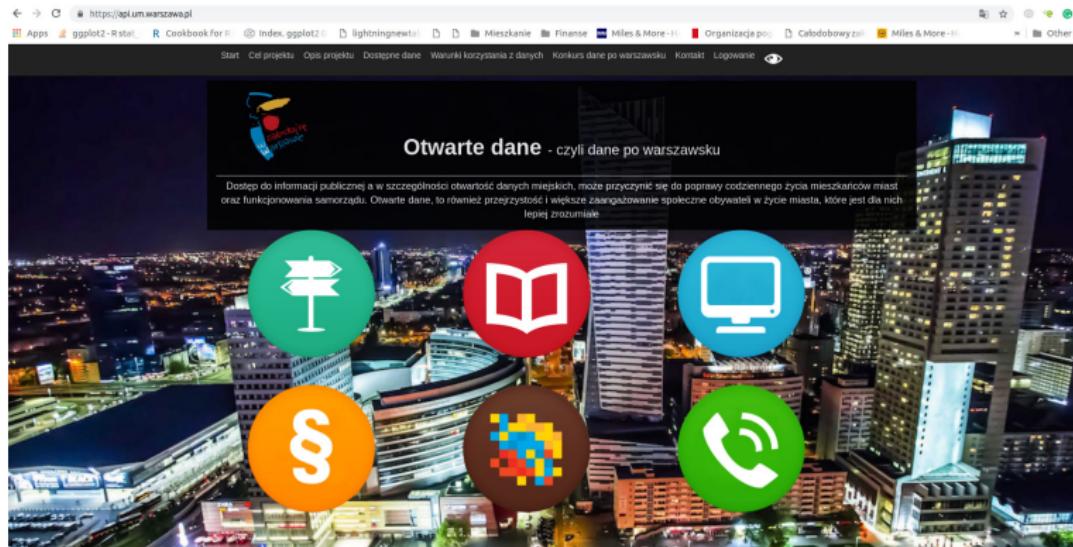
Dane

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



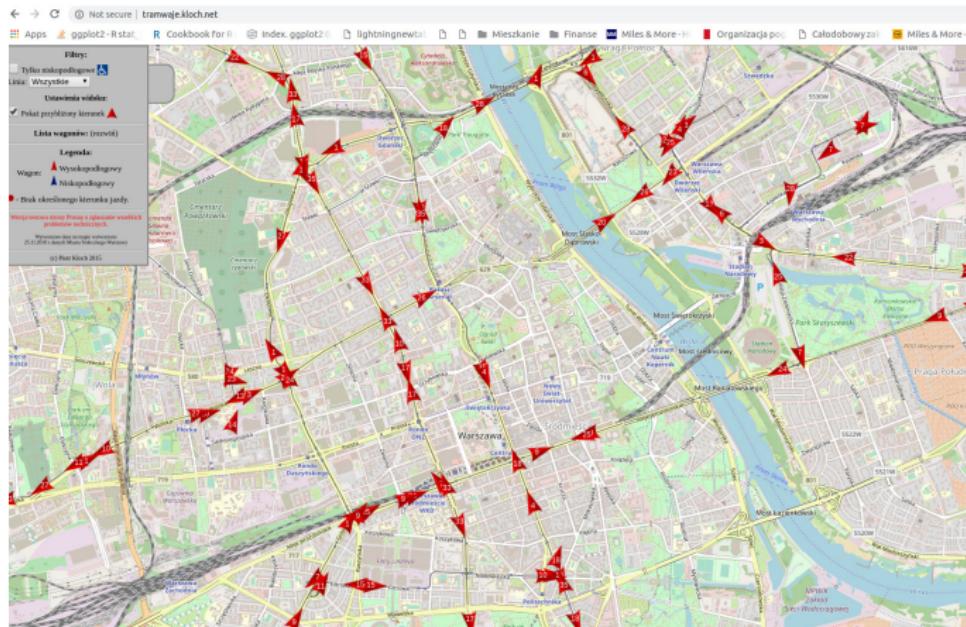
Wstęp

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. **położen tramwajów w czasie rzeczywistym**.



Wstęp

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



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

Pobieranie danych

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

```
rm(list = ls())
## API UM do oznaczenia pozycji tramwajów 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 <- [REDACTED]
um.waw.url <- "https://api.um.warszawa.pl/api/action/wsstore_get/?id=c7238cf8-8b1f-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
```

Pobieranie danych

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)

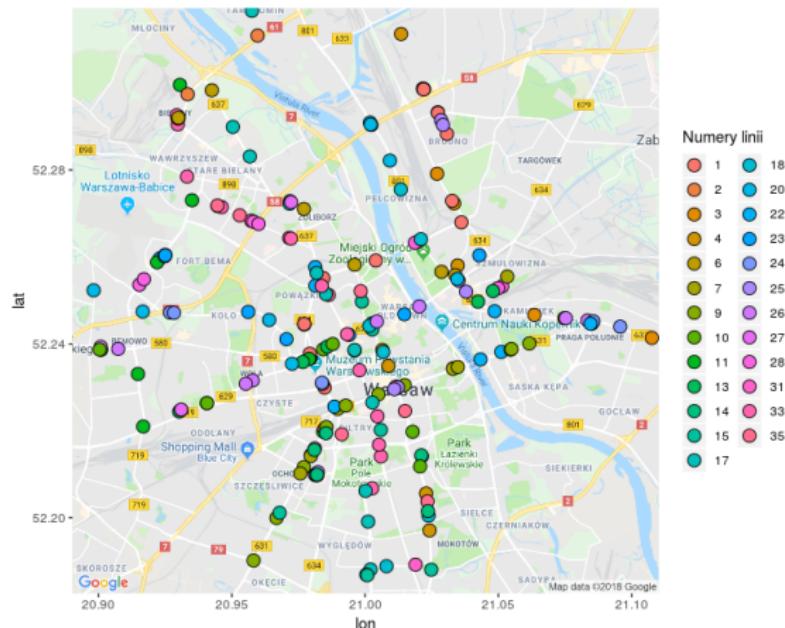
Pobieranie danych

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 w czasie

Dane

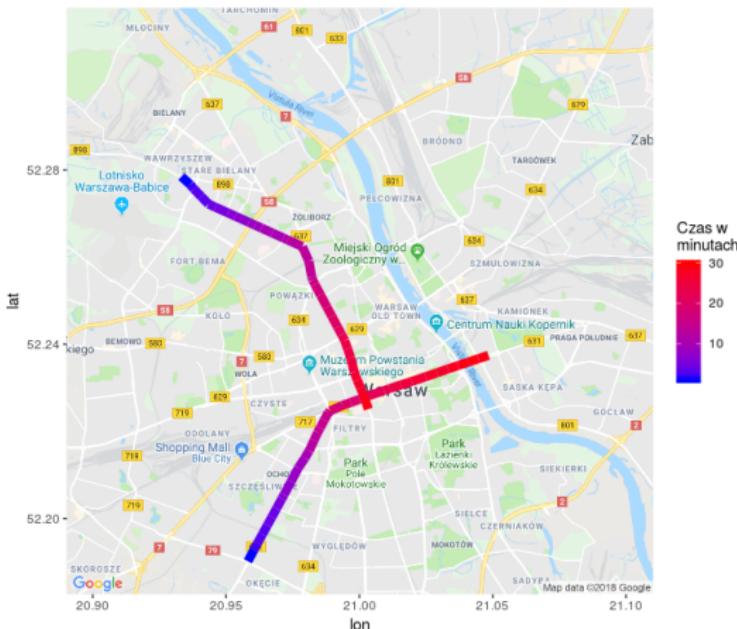
- można również ocenić jak zmienia się system w czasie,
- próbujemy układ z częstotliwością jednej minuty,
- na wykresie przedstawiono pojedynczą brygadę dla linii 7 oraz 33

Dane w czasie

Dane

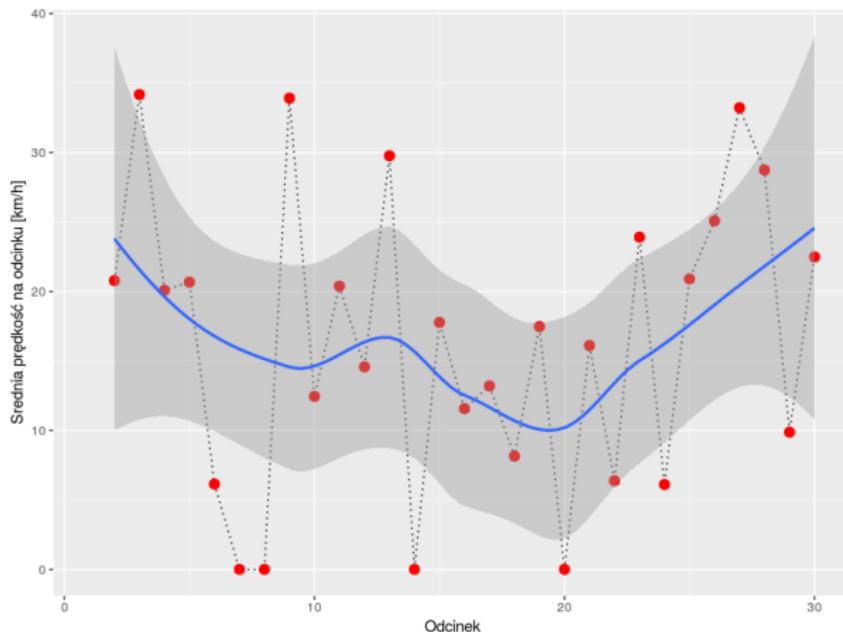
- można również ocenić jak zmienia się system w czasie,
- próbujemy układ z częstotliwością jednej minuty,
- na wykresie przedstawiono pojedynczą brygadę dla linii 7 oraz 33

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



Prędkości

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



Prędkości

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

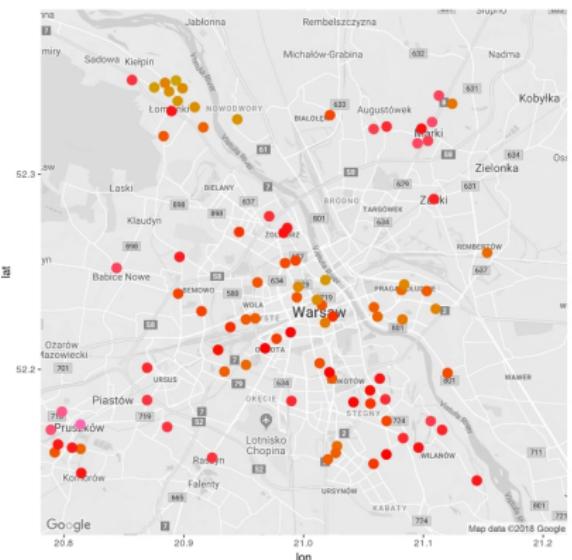


W ostatnich latach kluczowym problem staje się kwestia zanieczyszczenia powietrza, a dokładniej tzw. **smogu**. Jest to wypadkowa związana z przestrzalymi urządzeniami grzewczymi w domach oraz transportem indywidualnym. Od ok. roku działa platforma airly.eu, która udostępnia dane dotyczące poziomów zanieczyszczenia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

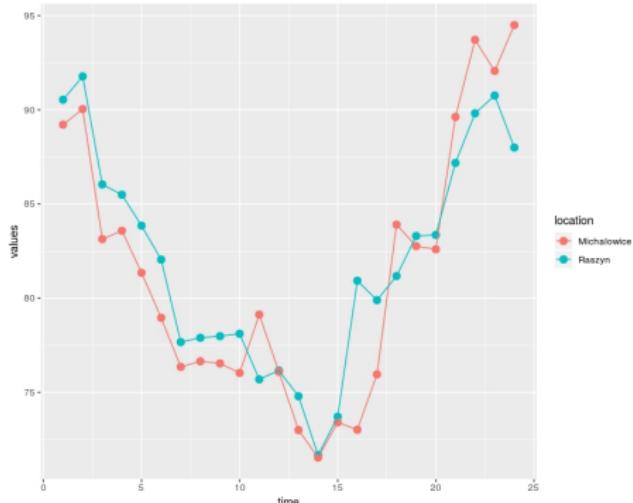
The screenshot shows the homepage of airly.eu. At the top left is the Airly logo. To its right is a horizontal navigation bar with links: Strona główna, Mapa, CzuJNIK, Dane, O Airly, Blog, Kariera, Projekty, Kontakt, and EN. The main content area features a large image of a city skyline under a blue sky. Overlaid on the image is a white rectangular box containing text and buttons. The text inside the box reads: "Czy wiesz czym oddychasz na co dzień?". Below this is a paragraph about building a sensor network across Poland. A blue button labeled "ZOBACZ NASZ FILM" is located at the bottom of this section. To the right of the main image is another white box with the text: "Sprawdź jakość powietrza w twoim otoczeniu!". Below this are three buttons: "PRZEJDŹ DO MAPY", "APLIKACJA NA ANDROIDA", and "APLIKACJA NA IOS".

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

Poniżej wartości czujników w Warszawie (niedziela, 25 listopada, godz. 20:00) oraz wykres poziomów dla dwóch wybranych lokalizacji.



Poziom AQI dla Raszyng i Michałowic za 24h, korelacja r=0.91



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**.

The screenshot shows the main landing page of the PLOS website. At the top, there's a navigation bar with links for About, For Authors, For Reviewers, Blog, Publications, and Submit Manuscript. Below the navigation is a large banner featuring a blue and yellow abstract image. The text "Open for Discovery" is prominently displayed, followed by "PLOS is a nonprofit publisher, innovator and advocacy organization." A search bar is located at the bottom of the banner. Below the banner, the heading "PLOS Journals" is shown, along with the text "215,000+ peer-reviewed articles are free to access, reuse and redistribute." Six thumbnail images of different PLOS journals are displayed below this text: PLOS ONE, PLOS BIOLOGY, PLOS MEDICINE, PLOS COMPUTATIONAL BIOLOGY, PLOS GENETICS, and PLOS NEXUS.

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