# Zgłębianie danych - Raport z projektu

Mateusz Miotk 195025

9 maja 2014

# Spis treści

# 1 Treść projektu

Przeprowadzić analizę opinii (sentiment mining) użytkowników Twittera na temat budzący różne emocje: polityka, politycy (dla kilku – zrobić sondaż) lub bieżące wydarzenia. Wykorzystać pakiet R z doinstalowaną paczką twitteR lub program RapidMiner z rozszerzeniem TextMining. Sklasyfikować kilkaset/tysiecy tweetów jako negatywne i pozytywne (w różnym stopniu) na podstawie odpowiednio dobranej listy słów kluczowych. Nastepnie przanalizuj i zinterpretuj wyniki: częstość występowania opinii, średnią, odchylenie standardowe, wykres/histogram.

# 2 Wstęp

Semantic analysis inaczej też nazywana opinion mining jest to obliczeniowe przetwarzanie emocji, opinii, postaw, ocen itd. na podstawie tekstów, którymi mogą być między innymi: wywiady, blogi, dyskusje, wiadomości, komentarze itd.
Metoda ta w ogólności polega na podziału opinii na trzy podstawowe grupy: pozytywne,neutralne oraz negatywne. Ma to ogromny wpływ w biznesie, ponieważ na przykład dzięki opiniom klientów danego produktu możemy stwierdzić czy produkt ten spełnia oczekiwania, jakie założyliśmy.
W raporcie zostanie pokazane wykorzystanie tej techniki, opierając się na komentarzach znanego medium społecznościowego jakim jest twitter, badając emocje dotyczące różnych aktualnych wydarzeń ze świata.

# 3 Narzędzia jakie zostały użyte.

Eksperyment został wykonany na systemie operacyjnym Ubuntu 14.04 x64 za pomocą pakietu statystycznego R który można pobrać ze strony: `http://www.r-project.org/`.
Poza tym będziemy potrzebować pakietu **libcurl4-openssl-dev** który możemy zainstalować za pomocą terminala poleceniem **sudo apt-get install libcurl4-openssl-dev**. Pakiet ten jest nam potrzebny do połączenia pakietu R z twitterem.

# 4 Schemat algorytmu wykonania sentiment analysis

Badanie opinii będzie ogólnie wyglądało następująco:

**Algorytm 1** Schemat algorytmu wykonania sentiment analysis

1: **procedure** SENTIMENTANALYSISTWITTER
2: Dokonaj połączenia pakietu R z twitterem.
3: Pobierz dane z twittera.
4: Dokonaj wyczyszczenia oraz poprawienia otrzymanych danych poprzez wyrażenia regularne.
5: Wykonaj funkcję sentiment dla wyczyszczonych danych.
6: **end procedure**

# 5 Opis poszczególnych metod algorytmu

## 5.1 Połączenie pakietu R z twitterem

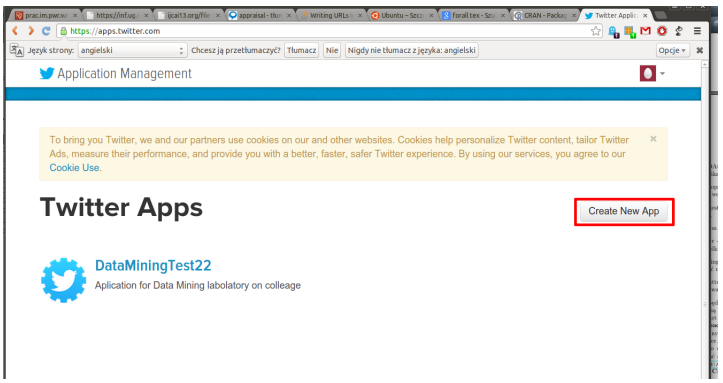
Do połączenia pakietu R z twitterem będziemy musieli zainstalować następujące dodatkowe pakiety w pakiecie R:

- RCurl - pakiet, który pozwala nam połączyć pakiet R ze stroną http//: lub ftp//:
- ROAuth - pakiet zawierający interfejs połączenia aplikacji wedle specyfikacji OAuth 1.0 .
- bitops - pakiet zawierający bitowe operację, które możemy wykonywać na wektorach.
- digest - pakiet zawierający implementację funkcji skrotów MD5, sha1 itd.
- rjson - pakiet pozwalający czytać/zapisywać w formacie json.
- plyr - pakiet zawierający mnóstwo funckji, służący do rozwiązywania wielkich problemów poprzez dzielenie je na podproblemy.
- stringr - pakiet zawierający ulepszone funkcję, które możemy wykonywać na napisach.

- twitteR - pakiet zawierający funkcję, poprzez które możemy wykorzystywać twittera w pakiecie R.

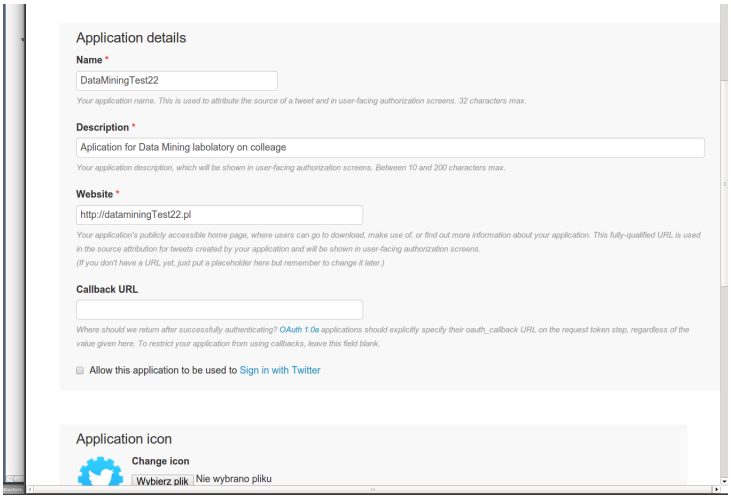
Nie będziemy korzystać z wszystkich wyżej wymienionych pakietów, aczkolwiek są one wymagane do uruchomienia pakietu twitteR. Aby zainstalować pakiet w języku R należy w konsoli wpisać polecenie:
**install.packages**(”*nazwa*”)**.**

Kolejnym krokim jakim musimy wykonać jest stworzenie aplikacji na stronie twitter.com.
Aby tego dokonać należy wejść na stronę `https://dev.twitter.com/` i zalogować się poprzez konto na twitterze. Następnie należy wejść na stronę `https://apps.twitter.com/` i stworzyć aplikacje poprzez kliknięcie na przycisk **Create a new app**.



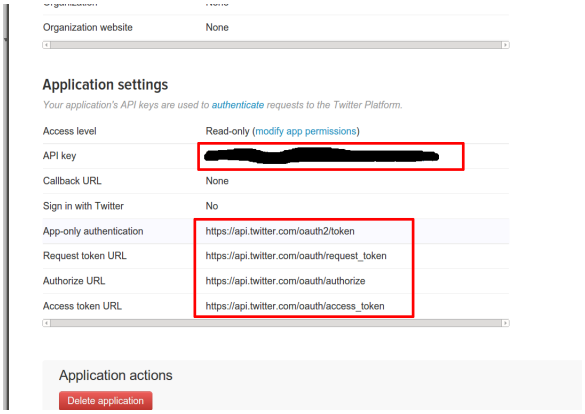

Rysunek 1: Tworzenie nowej aplikacji w twitterze.

Następnie należy wypełnić pola obowiązkowe i utworzyć aplikację:



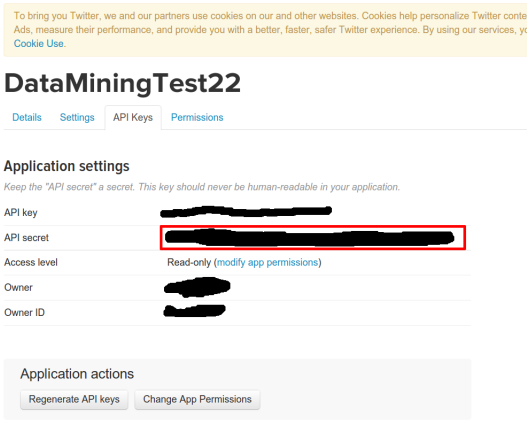

Rysunek 2: Opis tworzonej aplikacji

Po utworzeniu aplikacji wchodzimy do niej, otwieramy zakładkę **Details** i kopiujemy następujące elementy:




Rysunek 3: Potrzebne elementy z naszej twitterowej aplikacji

Ostatnią rzecz jaką będziemy potrzebować to **apiSecret**, który znajdziemy w zakładce **API Keys**.




Rysunek 4: API secret naszej aplikacji

.

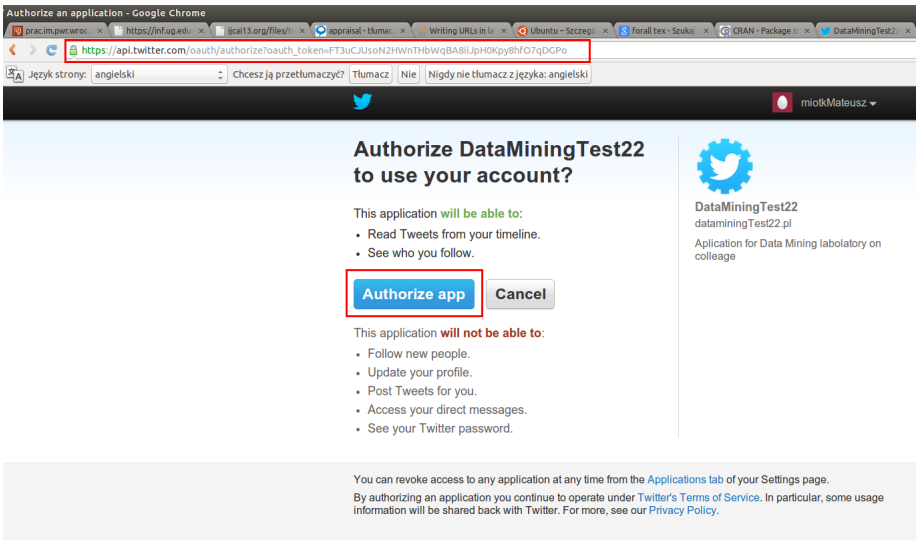
Mając wszystkie te rzeczy możemy przystąpić do napisania kodu w języku R, która połączy się z twitterem poprzez naszą utworzoną aplikację.

```
1
2  library(RCurl)
3  options(RCurlOptions = list(cainfo = system.file("CurlSSL","cacert.pem",package="RCurl")))
4  require(twitteR)
5  reqURL = "https://api.twitter.com/oauth/request_token"
6  accessURL = "https://api.twitter.com/oauth/access_token"
7  authURL = "https://api.twitter.com/oauth/authorize"
8  apiKey = [redacted]
9  apiSecret = [redacted]
10 twitCred = OAuthFactory$new(consumerKey = apiKey,consumerSecret = apiSecret,requestURL = reqURL, accessURL = accessURL,authURL=authURL)
11 twitCred$handshake(cainfo=system.file("CurlSSL","cacert.pem",package = "RCurl"))
12 registerTwitterOAuth(twitCred)
13 save(twitCred, file="~/Dropbox/Studia/Zglebianie_Danych/Labs/Project/twitCred.RData")
```

Rysunek 5: Kod autoryzujący pakiet R z naszą aplikacja w twitterze.

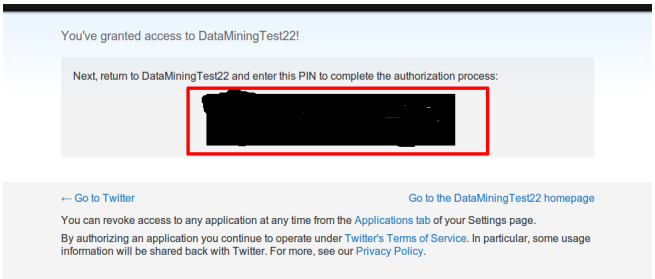
Polecenie **library** ładuje pakiet do środowiska. Polecenie **options** pobiera standardowy certyfikat cacert.pem do wykonania autoryzacji aplikacji. Polecenie **require** działa tak samo jak library z tą różnicą, że ładuje wszystkie dodatkowe potrzebne pakiety do środowiska. Polecenie **OAuthFactory** łączy środowisko pakietu R z naszą aplikacją twitterową, która jest zaakceptowana poprzez polecenie **registerTwitterOAuth**. Polecenie **save** zapisuje nam plik, dzięki któremu będziemy mogli w każdej chwili połączyć się z naszą aplikacją używając wyłącznie polecenie **registerTwitterOAuth**.
Gdy wywołamy powyższy skrypt to środowisko pakietu R wygeneruje nam link dzięki któremu będziemy musieli autoryzować naszą aplikację stworzoną

na twitterze. Aby tego dokonać należy skopiować link oraz kliknąć przycisk **Authorize app**




Rysunek 6: U góry otrzymany link z konsoli R, poniżej przycisk autoryzujący aplikację.

Po kliknięciu przycisku **Authorize app** wyświetli się nam numer pin który będziemy musieli wpisać w konsoli środowiska pakietu R.



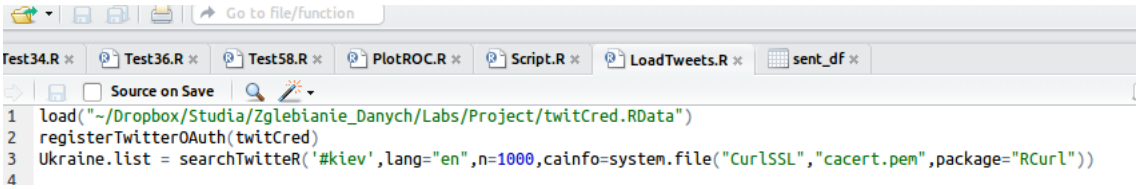

Rysunek 7: Kod pin autoryzujący naszą aplikację z twitterem

```
> source('~/Studia/Zglebianie_Danych/Labs/Project/Script.R')
To enable the connection, please direct your web browser to:
https://api.twitter.com/oauth/authorize?oauth_token=FT3uCJUsoN2HWnTHbWqBA8iiJpH0Kpy8hfO7qDGPo[redacted]
When complete, record the PIN given to you and provide it here:
```

Rysunek 8: Wpisanie pinu do konsoli pakietu R

## 5.2 Pobieranie danych z twittera

Jeśli już mamy aplikację zautoryzowaną z twitterem możemy przejść do przeglądania tweetów na dany temat.




Rysunek 9: Połączenie się z twitterem oraz przykładowa funkcja wyszukująca tweety.

Wyszukiwanie tweetów weykonuje funckja **searchTwitteR**. Jej parametrami są w naszym przypadku ciąg jaki szukamy, w jakim języku, ile chcemy uzyskać tweetów oraz certyfikat przez który się łączymy z api Twittera. Uzyskane dane wyglądają obecnie następująco:

| | text | favorited | fa |
|---|---|---|---|
| 1 | RT @arabresistance: Pro-#kiev armed men operating freely in cities & kidnapping anti-#Kiev activists. http://t.co/d0AvWKLo1U http://t.co/a... | FALSE | 0 |
| 2 | RT @sygigurit: Video shows a man shooting at people in Odessa's burning House of Trade Unions http://t.co/Ho4PZfOiw1 #Kiev | FALSE | 0 |
| 3 | RT @lzvieta: Woman survived from #odessafire sharing terrifying expirience. #English subtitles http://t.co/uOUPXBhXww #Odessa #Ukraine #Kie... | FALSE | 0 |
| 4 | RT @Serebryany: @LowMaintainLife #Odessa #House of #Trade #Unions #AXE of #Euromaidan - #nazi http://t.co/UYBeFu0nMt #Kiev #punishers http:... | FALSE | 0 |
| 5 | #German views on the #Ukraine #Russia crisis - #Kiev #Moscow #Berlin #Frankfurt #Munich #Germany #EU http://t.co/kBPSasQuNu | FALSE | 0 |
| 6 | RT @Serebryany: @LowMaintainLife #Odessa #House of #Trade #Unions #AXE of #Euromaidan - #nazi http://t.co/UYBeFu0nMt #Kiev #punishers http:... | FALSE | 0 |
| 7 | RT @UkrToday: Urgent! #Ukraine #Kiev regime has rejected #Russia's #Putin's dialogue proposal http://t.co/pe8o7ehtee via @FarEasterner | FALSE | 0 |
| 8 | RT @DrMarcusP: Peace in #Ukraine will only come when #Kiev pulls its forces back from the south and east of the country, including its extr... | FALSE | 0 |
| 9 | RT @DrMarcusP: Peace in #Ukraine will only come when #Kiev pulls its forces back from the south and east of the country, including its extr... | FALSE | 0 |
| 10 | RT @N1EUWS: #Obama's #Bloodbath in #Odessa: As #Guilty as Anyone in #Kiev: "As the building... http://t.co/r8juS3DLV9 # #nieuws #news | FALSE | 0 |

Rysunek 10: Przykładowe dane pobrane za pomocą paczki twitteR.

Aby otrzymać powyższą tabelę z danymi musimy te dane przerobić z naszej listy do obiektu typu **data frame**. Wykonujemy to poleceniem:
**twListToDF**(*nazwa*)
Otrzymane tweety zawsze możemy zapisać w postaci **.csv** używając polecenia:
**write.csv**(**obiekt,ścieżka,row.names=F**).

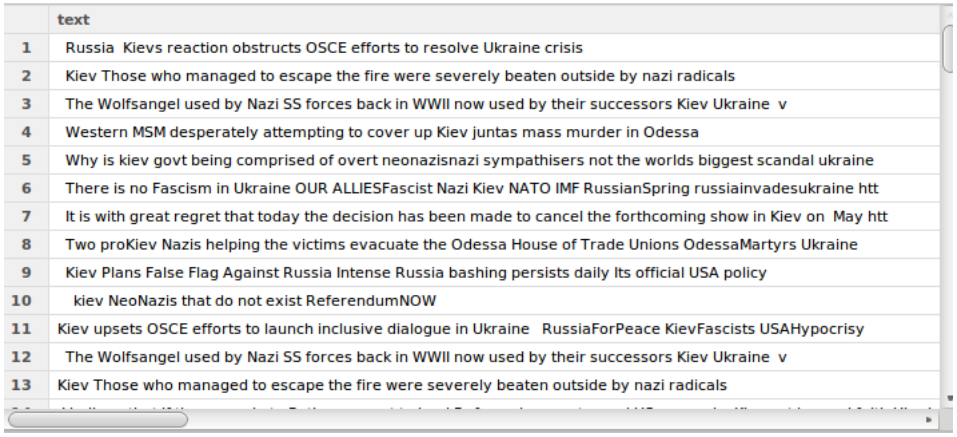
## 5.3 Wyczyszczenie i uporządkowanie danych za pomocą wyrażeń regularnych

Uzyskane powyższe tweety są bardzo nieeleganckie. Zawierają one znaki, które powodują że w algorytmie sentiment nie będą one brane pod uwagę. Musimy oczyścić otrzymane tweety za pomocą wyrażeń regularnych. Dokonujemy tego poprzez wywołanie następującego kodu:

```
Ukraine.df$text = gsub("@\\w+", "", Ukraine.df$text)
Ukraine.df$text = gsub('[[:punct:]]', '',Ukraine.df$text)
Ukraine.df$text = gsub('[[:cntrl:]]', '',Ukraine.df$text)
Ukraine.df$text = gsub('\\d+', '', Ukraine.df$text)
Ukraine.df$text = gsub("http\\w+", "", Ukraine.df$text)
Ukraine.df$text = gsub("RT|via", "", Ukraine.df$text)
Ukraine.df$text = gsub("http", "", Ukraine.df$text)
```

Rysunek 11: Polecenia służące wyczyszczeniu i uporządkowaniu danych z twittera.

Tak więc aby oczyścić nasze tweety za pomocą wyrażeń regularnych używamy funkcji **gsub**. Funkcja **gsub** zastępuje wyrażenia, który jest podany jako pierwszy argument wzorcem podanym jako drugi argument. Jako trzeci argument wymaga on podania danych na których ma wykonać zamianę. Pierwszy wiersz pozbywa się nazw loginów, z którego otrzymaliśmy dany tweet. Wyrażenie [[: *punct* :]] wyszukuje wszystkie znaki specjalne i punktowalne typu @ [ ] itd. Wyrażenie [[: *cntl* :]] wyszukuje wszyskie znaki kontrolne. Pozostałe wyrażenia są jasne do zrozumienia.
Jeśli wykonamy powyższe kroki na zbiorze naszych zebranych tweetów to otrzymamy już bardziej elegancką formę do przeczytania i wykonania sentiment analysis.



| | text |
|---|---|
| 1 | Russia Kievs reaction obstructs OSCE efforts to resolve Ukraine crisis |
| 2 | Kiev Those who managed to escape the fire were severely beaten outside by nazi radicals |
| 3 | The Wolfsangel used by Nazi SS forces back in WWII now used by their successors Kiev Ukraine v |
| 4 | Western MSM desperately attempting to cover up Kiev juntas mass murder in Odessa |
| 5 | Why is kiev govt being comprised of overt neonazisnazi sympathisers not the worlds biggest scandal ukraine |
| 6 | There is no Fascism in Ukraine OUR ALLIESFascist Nazi Kiev NATO IMF RussianSpring russiainvadesukraine htt |
| 7 | It is with great regret that today the decision has been made to cancel the forthcoming show in Kiev on May htt |
| 8 | Two proKiev Nazis helping the victims evacuate the Odessa House of Trade Unions OdessaMartyrs Ukraine |
| 9 | Kiev Plans False Flag Against Russia Intense Russia bashing persists daily Its official USA policy |
| 10 | kiev NeoNazis that do not exist ReferendumNOW |
| 11 | Kiev upsets OSCE efforts to launch inclusive dialogue in Ukraine RussiaForPeace KievFascists USAHypocrisy |
| 12 | The Wolfsangel used by Nazi SS forces back in WWII now used by their successors Kiev Ukraine v |
| 13 | Kiev Those who managed to escape the fire were severely beaten outside by nazi radicals |

Rysunek 12: Wyczyszczone i uporządkowane dane z twittera.

## 5.4 Funkcja sentiment

Mając tak przygotowane dane możemy przystąpić już teraz do realizacji sentiment analysis. Aby tego dokonać będziemy jeszcze potrzebować drobnych zmian w otrzymanych, przerobionych tekstach tekstach. Po pierwsze musimy zamienić wszystko na małe litery. Wykonujemy to za pomocą polecenia **tolower**. Następnie dla każdego otrzymanego tweeta dzielimy go na listę pojedyńczych słów. Dokonujemy tego za pomocą polecenia **strsplit**. Teraz wystarczy tylko dokonać porównania z listą pozytywnych oraz negatywnych słów. Dla języka angielskiego listę pozytywnych oraz negatywnych słów możemy ściągnąć ze strony `http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar`
Używamy funkcji **match**, która obliczy nam pozycje podzielonych słów naszego tweeta w naszym słowniku słów pozytywnych i negatywnych. Jeśli dane słowo nie występuje w żadnym ze słowników otrzymuje ono wartość zero.
Aby otrzymać czy dana opinia jest pozytywna, neutralna czy negatywna wystarczy wykonać odpowiednie działanie:

$$opinia = \sum(\text{pozytywne słowa}) - \sum(\text{negatywne słowa})$$

Uzyskany wynik należy wówczas implementować następująco:

$$\text{sentiment analysis} \begin{cases} pozytywna & opinion > 0 \\ neutralna & opinion = 0 \\ negatywna & opinion < 0 \end{cases}$$

```
41      sentence = tolower(sentence)
42
43      word.list = str_split(sentence,'\\s+')
44      words = unlist(word.list)
45
46      positive.matches = match(words,positive.words)
47      negative.matches = match(words,negative.words)
48
49      positive.matches = !is.na(positive.matches)
50      negative.matches = !is.na(negative.matches)
51
52      score = sum(positive.matches) - sum(negative.matches)
53      return (score)
```

Rysunek 13: Implementacja funckji sentiment w pakiecie R

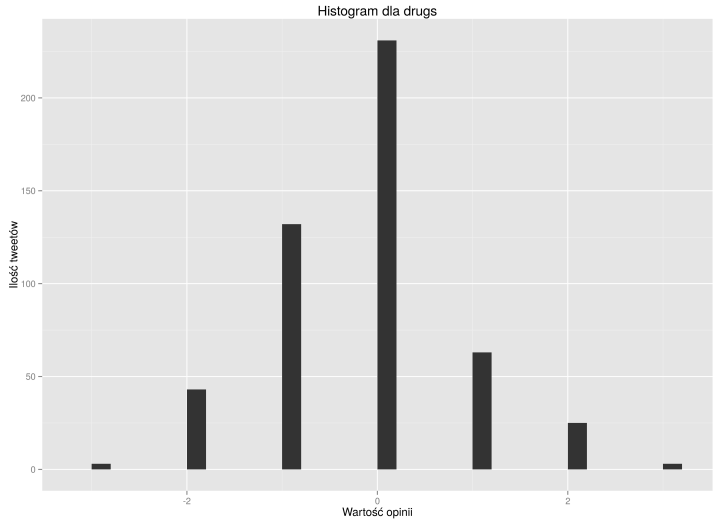
# 6 Użycie sentiment analysis do analizy danych

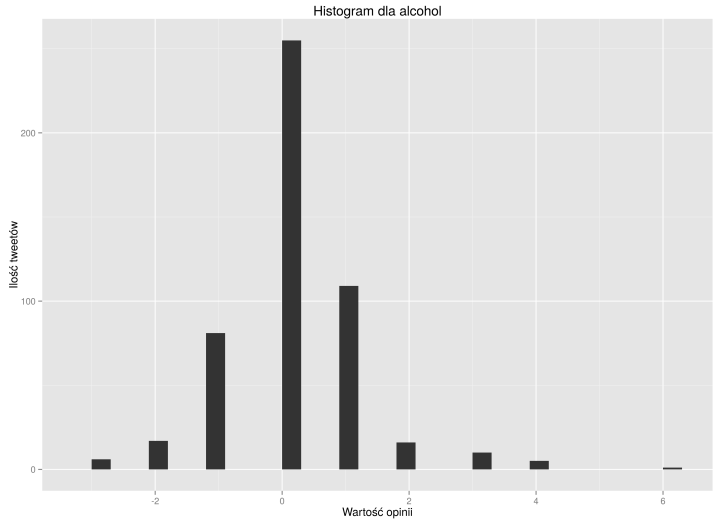
Mając tak napisany program w pakiecie R rozważmy następujące doświadczenie: Weźmy sobie kilkaset tweetów dotyczące słów wzbudzający emocję. Niech będą nimi następujące słowa: narkotyki, alkohol, rasizm, małżeństwo, praca, samochód. Dokonamy sentiment analysis na **500** tweetach zawierające powyższe nazwy. Obliczymy do tego histogram występowania wartości opinion oraz jakie ma ono odchylenie standardowe oraz średnią. Dzięki uzyskanym wynikom możemy stwierdzić jak dane słowo jest odbierane przez użytkowników twittera.
Tak więc dla danych słów uzyskaliśmy następujące wyniki:

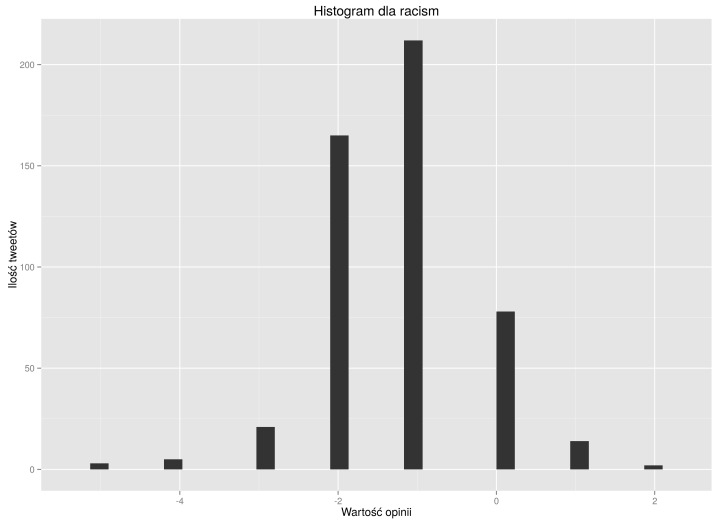
| Słowo | Pozytywne | Negatywne | Neutralne | Średnia | Odchylenie |
|---|---|---|---|---|---|
| narkotyki | 91 | 178 | 231 | -0.21 | 1 |
| alkohol | 141 | 104 | 255 | 0.13 | 1.07 |
| rasizm | 16 | 406 | 78 | -1.24 | 0.96 |
| małżeństwo | 159 | 244 | 97 | 0.04 | 1.29 |
| praca | 421 | 12 | 67 | 1.24 | 1.1 |
| samochód | 141 | 84 | 275 | 0.2 | 1.02 |

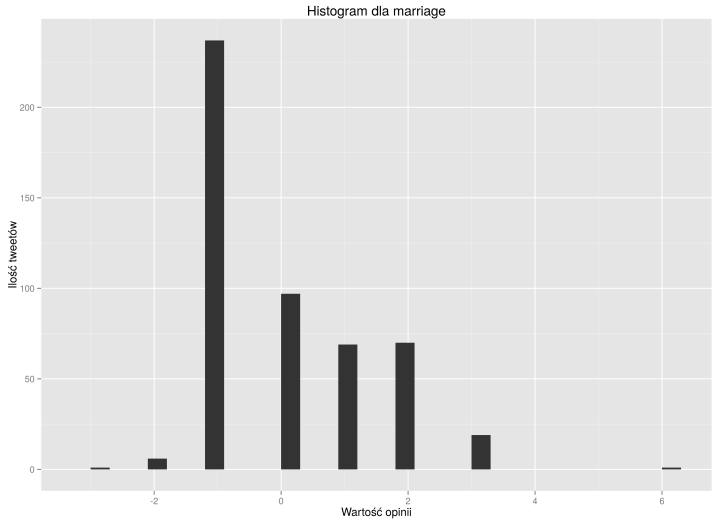
Można było spodziewać się następujących wyników. Otóż najbardziej negatywnym słowem w naszym zestawieniu jest rasizm, zaś najbardziej pozytywnym praca. Zaskakujące są wyniki w słowie narkotyki gdzie bardzo dużo opinii jest sklasyfikowane jako neutralne. Drugim zaskoczeniem jest przewaga negatywnych opinii występujące w słowie: małżeństwo.

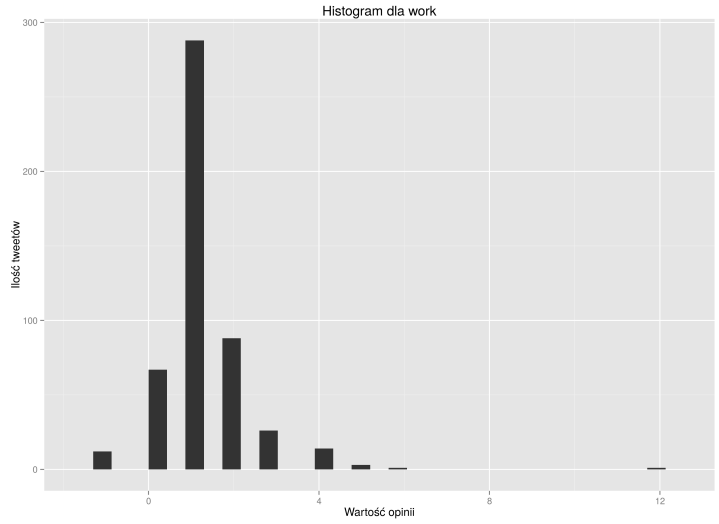
Przejdźmy do histogramów częstości dla danych słów:

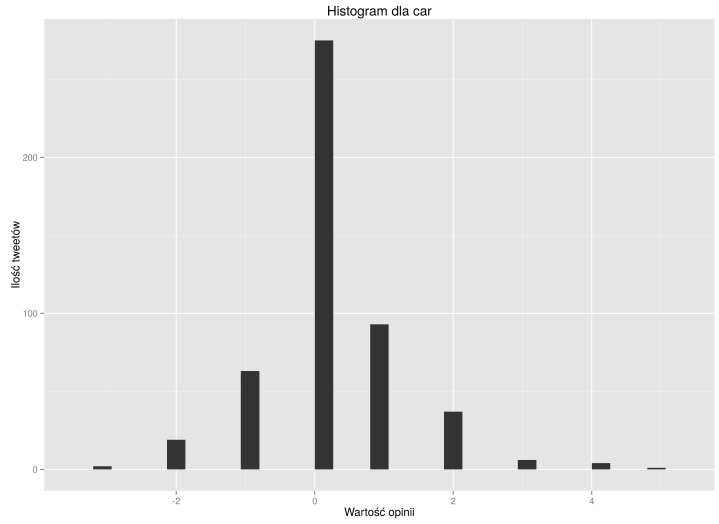



Histogram dla racism
Ilość tweetów
200
150
100
50
0
-4
-2
0
2
Wartość opinii


Histogram dla marriage
Ilość tweetów
200
150
100
50
0
-2
0
2
4
6
Wartość opinii




Powyższe wykresy ukazują częstości wartości opinion, który został przydzielony poprzez naszą funkcję sentiment.

# 7 Objaśnienie napisanego programu w języku R

## 7.1 Ogólne działanie

W pełni działający program znajduje się w repozytorium GitHub'a pod adresem: `https://github.com/miotek32/DataMiningProject`

Program został tak skonstruowany, że wymaga tylko uruchomienia skryptu znajdującego się w pliku **main.R**. Poza tym jeśli użytkownik chce zmienić dane dla którego chce obliczyć sentiment analysis wystarczy tylko zmienić linijkę **words** w pliku **main.R**. Resztę czyli pobranie i zapisanie danych skrypt robi automatycznie. Dane wraz z histogranami zapisane są w folderze **Data**. Pliki z danymi są zapisane w formacie **.csv** . Każdy z nich zawiera w pierwszej kolumnie wartość opinii a następnie samą opinie, ale w swej pierwotnej postaci.

## 7.2 Plik CreateData.R

Plik **CreateData.R** zawiera funkcję createData, która za swoje argumenty przyjmuje hasło, które ma poszukiwać na twitterze oraz liczbę tweetów do pobrania. Jego działanie polega na pobraniu tweetów z serwisu oraz zapisanie ich pod nazwą: **hasło.csv**.

## 7.3 Plik LoadTweets.R

Plik **LoadTweets.R** zawiera dwie funkcję: **score.sentiment** oraz **sentimentFunction**. Pierwsza z nich realizuje algorytm sentiment analysis, który został wyjaśniony na początku dokumentu. Druga funkcja, która za argument przyjmuje nazwę pliku otwiera ten plik oraz wywołuje funkcję **score.sentiment** i zapisuje wynik w tym samym pliku. Poza tym generuje on histogram w postaci **Histhasło.png** oraz wyświetla w konsoli wartości: ilość opinii pozytywnych, negatywnych, neutralnych, średnia oraz odchylenie standardowe.

## 7.4 Plik main.R

Główny plik, którego uruchomienie skryptu spowoduje automatyczne wygenerowanie plików, wyświetlenie wartości dotyczącej średniej i odchylenia standardowego oraz wykresu histogramu. Kluczowym elementem w tym pliku jest zmienna words, która zawiera listę słów, które chcemy wykonać algorytm sentiment analysis.

# 8 Ograniczenia

Pakiet **twitteR** w języku R ma pewne ograniczenie, które wynika z wykorzystania zewnętrznego API usługi twitter. Polega ona na tym, że niekoniecznie możemy uzyskać żądaną liczbę tweetów dla danego słowa. Bardzo często

zdarza się że funkcja **searchTwitter**(**słowo,n=1000**) może wyświetlić nam komunikat, że niestety żądaliśmy 1000 tweetów, ale jedynie udało się uzyskać mniejszą ilość. Jest to spowodowane ograniczeniami, które znajdują się w API serwisu twitter. Drugim ograniczeniem jest czas działania programu. Ponieważ język R jest jednowątkowy to wykonanie napisanego tutaj programu trwa około 2 minut.

# 9 Podsumowanie

Przedstawione tutaj rozwiązanie nie jest jedyne. Istnieje na przykład metoda, która potrafi oprócz sklasyfikowania opinii jako pozytywna czy negatywna, przydzielić która opinia wyrażona jest przez szczeście, złość itd. Pakiet R posiada takie rozwiązanie, które zawarte jest w pakietach: **tm** oraz **sentiment**. Jednak pakiety te są w fazie testowej, nie są one wydane w wersjach produkcyjnych, ale możliwości ich są obiecujące.

# Literatura


[1] http://davetang.org/muse/2013/04/06/using-the-r_twitter-package/

[2] Video kurs: P.Anderson *How to use R in Mining Twitter*

*[3] http://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html*

*[4] http://www.regular-expressions.info/posixbrackets.html*

*[5] http://www.r-project.org/*

*[6] http://cran.r-project.org/web/packages/available_packages_by_name.html*