R

Język programowania

Andrzej Kostański

Warszawa 2013

Kilka faktów o R:

- Rozwijany na licencji GNU
- Paradygmaty: OO, imperatywny, funkcyjny, proceduralny.
- Jest implementacyjną statystycznego języka S oraz Scheme
- Wywodzi się z bioinformatyki
- Środowisko jest napisane w C, Fortranie oraz R

… w między czasie można pobrać i zainstalować: http://cran.r-project.org/

Wspierane platformy:

- Linux
- Mac OS X
- Windows

Jak korzystać

- Konsola systemowa
- R klasyczne idle
- R Studio
- Batch mode

ZALETY

- Łatwość konfiguracji
- Duże bogactwo bibliotek tworoznych przez użytkowników z różnych dziedzin
- Nie wymaga dużo pisania kodu
- Naturalne struktury danych do data miningu - data.frame
- Możliwość wykonywanie operacji na macierzach
- Duże możliwości wizualizacji danych, w tym danych przestrzennych
- Możliwość wywoływania w trakcie wykonywania kodu napisanego w C++, C, czy Javie by wykonać “na zewnątrz” ciężkie obliczeniowo zadania.

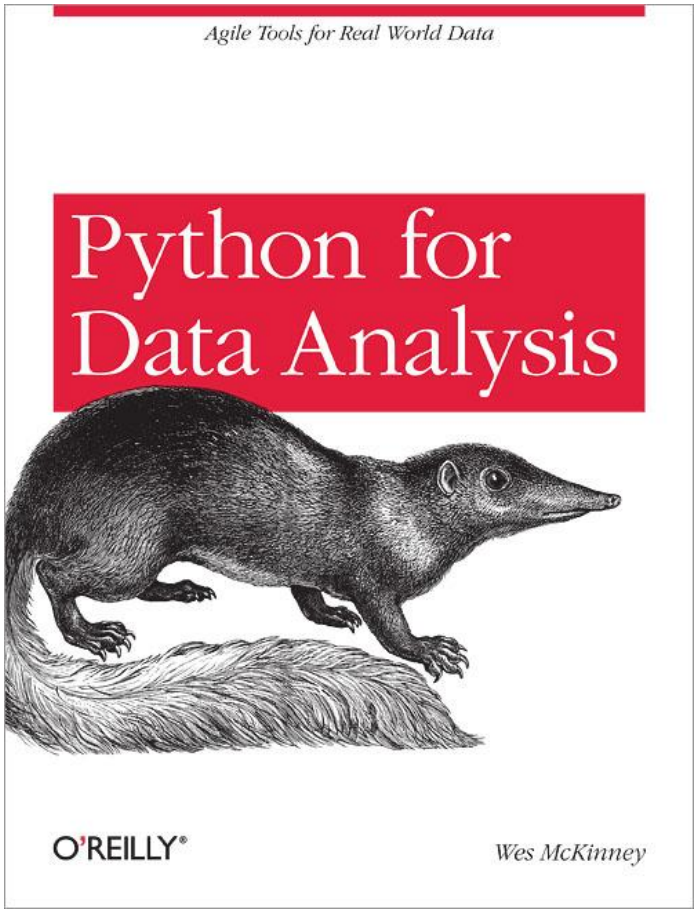
WADY

- Duże zróżnicowanie jakości bibliotek – nie zawsze zoptymalizowane dla bardzo dużych danych
- Wymaga trochę czasu by zapoznać się z bogactwem bibliotek, lub żmudnego pisania funkcji
- Trochę mniejsze community niż np. Pythona
- Inna wiedza osób w społeczności R – bardziej statystyczna, ekonometryczna niż informatyczna (co może jednak też być zaletą ;)
- Raczej bardzo bogaty pakiet do niezwykle szeroko pojętej analizy danych niż środowisko programistyczne do stworzeniajakiejś większej aplikacji, brak sensowengo ORM, czy frameworku webowego
- Niezbyt miła składnia

R a Python

Biblioteki pythona na które warto rzucić okiem:

- numpy,
- scipy,
- matplotlib,
- pandas,
- scikit-learn



R na Stackoverflow

5,800,656 questions

38,057
questions tagged

r about »

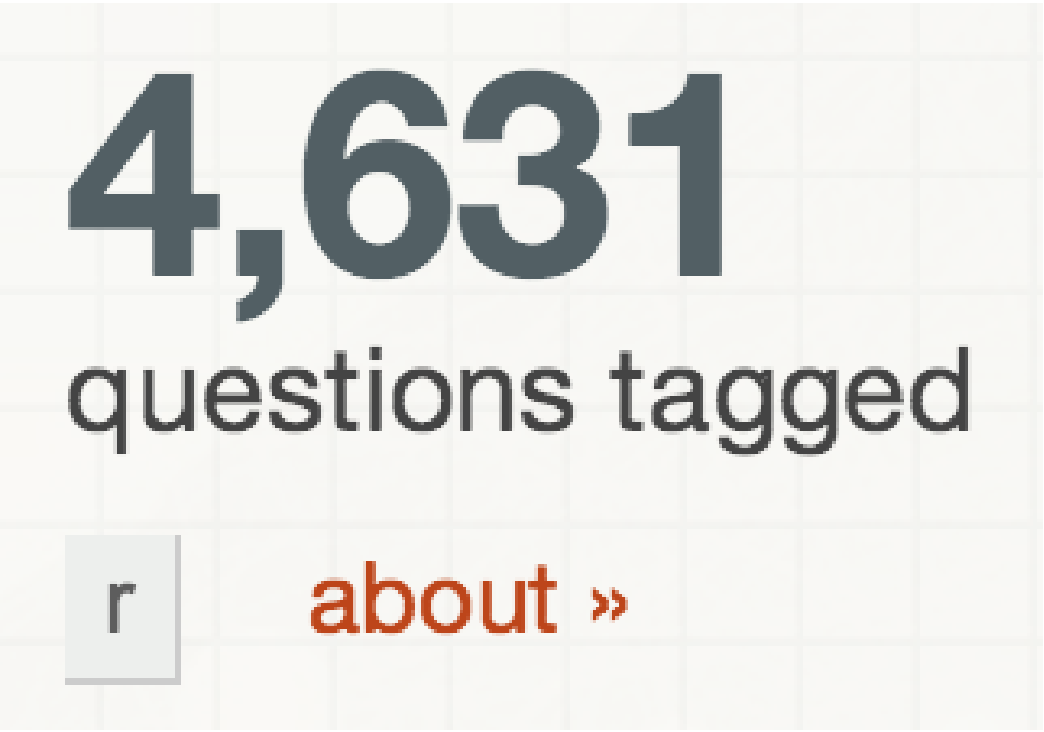
224,364
questions tagged

python about »

stackoverflow

R na CrossValidated

26,153 questions



Dlaczego warto znać R?

- Możliwość porozumienia się nt. danych i przeprowadzanych na nich analiz z osobami **nie** z branży IT, które nie *programują* na codzień: ekonomistami, ekonometrykami, statystykami, specjalistami od optymalizacji, osobami prowadzącymi badania ilościowe.
- Szybka konfiguracja, szybkie widoczne efekty

[R]ozgrzewka:

- >idle
- Instalowanie bibliotek
- Ustawianie sprawdzanie *working directory*
- Uruchamianie zapisanych skryptów (cmd+return)
- BATCH MODE, STANDARD INPUT

PRZYKŁAD:

- BATCH MODE with STANDARD INPUT:

```
RScript batch.R "Jakis tytul" 1 2 3 424 242 323 2.33 323 > output.log
```

```r
argv <- commandArgs(TRUE)
title <-  as.character(argv[1])
vars <- c(as.numeric(argv[2:length(argv)]))

cat("title =", title, "\n")
for (i in 1:length(vars)){ cat (i,"\n")}

today <-format(Sys.Date(), format="%d-%m-%Y")
png(paste("raport-",today,".png",sep=""))
barplot(vars,main=paste(title,today))
```

batch.r

```
import os
comand='''RScript batch.R "Jakis tytul" 1 2 3 424 242 323 2.33 323 > output.log'''
os.system(comand)
```

Wczytujemy dane:

- CSV
- XLS
- XML
- MySQL
- JSON

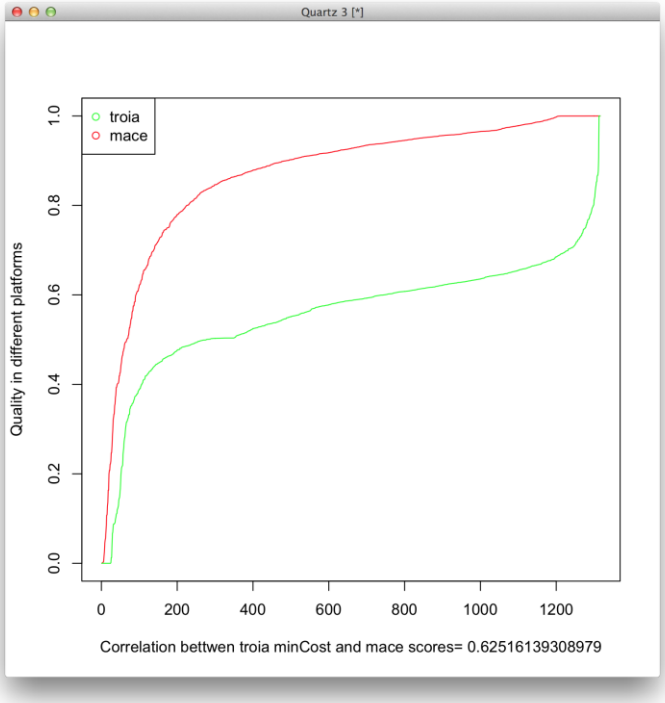
PRZYKŁAD

- Wyczyść wszystki dane / run garbage collector

```
1   #wyczyść wszystkie zmienne z pamięci
2   rm(list=ls(all=TRUE))
3   #run garbage collector
4   gc()
```

PRZYKŁAD:

- Wczytaj CSV z dysku lub z WWW + ploty




```
csv-plot.r
```

PRZYKŁAD:

- Wczytaj konkretne arkusze XLS, po pobraniu pliku z sieci

```
Xls.r
```

PRZYKŁAD:

- Zapisuj do/wczytuj z bazy danych MySQL

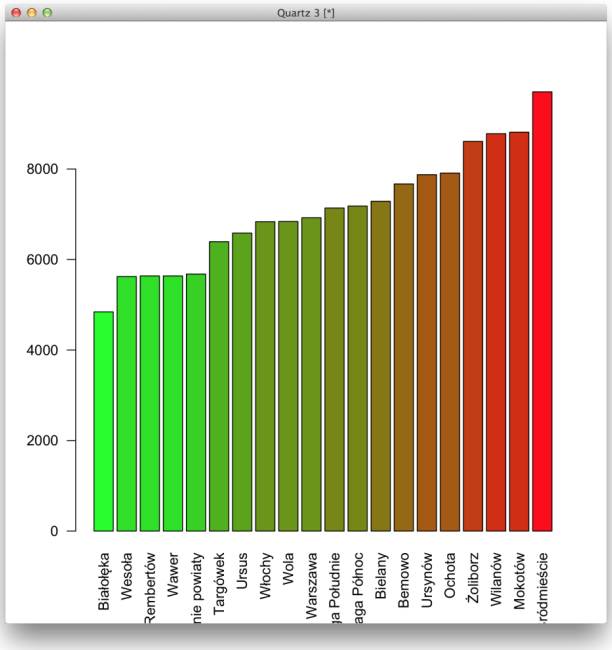
```r
install.packages("RMySQL")
library(RMySQL)
NAZWA_TABELI = "andi"
#połączenie
mydb = dbConnect(MySQL(), user='andi_10c',password='P@ssw0rd',dbname='andi_10c',host='mysql.superhost.pl
#tworzymy w bazie nowa tabele o nazwie iris_table, zapisujac do niej data frame iris (defaultowo dostępn
dbWriteTable(mydb, name=NAZWA_TABELI, value=iris)
#listujemy tabele w bazie
dbListTables(mydb)
#listujemy pola tabeli iris_table
dbListFields(mydb, NAZWA_TABELI)
#zapuszczamy sobie dowolne pożądane queries
rs=dbSendQuery(mydb, paste('select * from ',NAZWA_TABELI,' iris where Species = "setosa"'))
#przechwytujemy z querysetu dane do data framu
dane<-fetch(rs, n=-1)
```

{ db_mysql.r }

Przykład

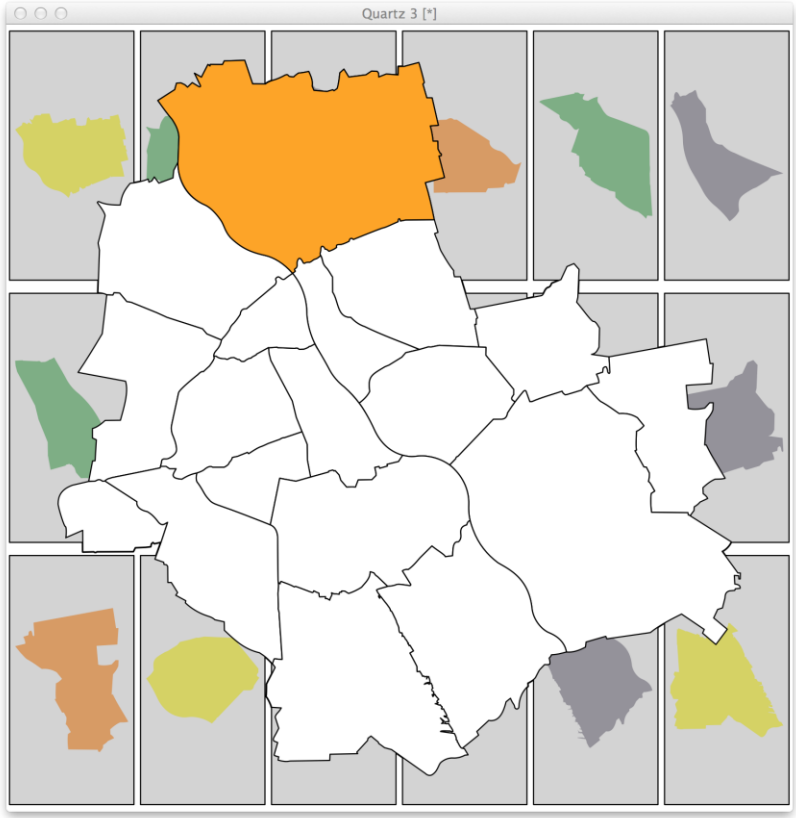
- kodowania UTF-8 dla danych z MySQL
- Zapis do CSV
- Plot z gradientem

	dzielnica	SREDNI_METRAZ	SREDNIA_CENA	SREDNIA_CENA_M	col
2	Białołęka	70	336957	4843	#B60048
15	Wesoła	81	454024	5624	#6D0091
8	Rembertów	62	349903	5638	#A3005B
14	Wawer	92	518943	5638	#5B00A3
19	Zachodnie powiaty	28	159000	5679	#FF0000
10	Targówek	51	323508	6394	#B60048
11	Ursus	49	323319	6584	#B60048
18	Włochy	71	482772	6835	#6D0091
17	Wola	64	440733	6840	#7F007F
13	Warszawa	53	370202	6923	#A3005B
7	Praga Południe	52	373585	7138	#A3005B
6	Praga Północ	44	317743	7181	#B60048
3	Bielany	52	379592	7287	#91006D
1	Bemowo	61	464775	7670	#6D0091
12	Ursynów	63	497255	7875	#5B00A3
5	Ochota	51	402098	7910	#91006D
20	Żoliborz	49	418414	8610	#7F007F
16	Wilanów	80	699073	8780	#0000FF
4	Mokotów	60	526644	8814	#4800B6
9	Śródmieście	55	531942	9705	#4800B6



mieszkania.r

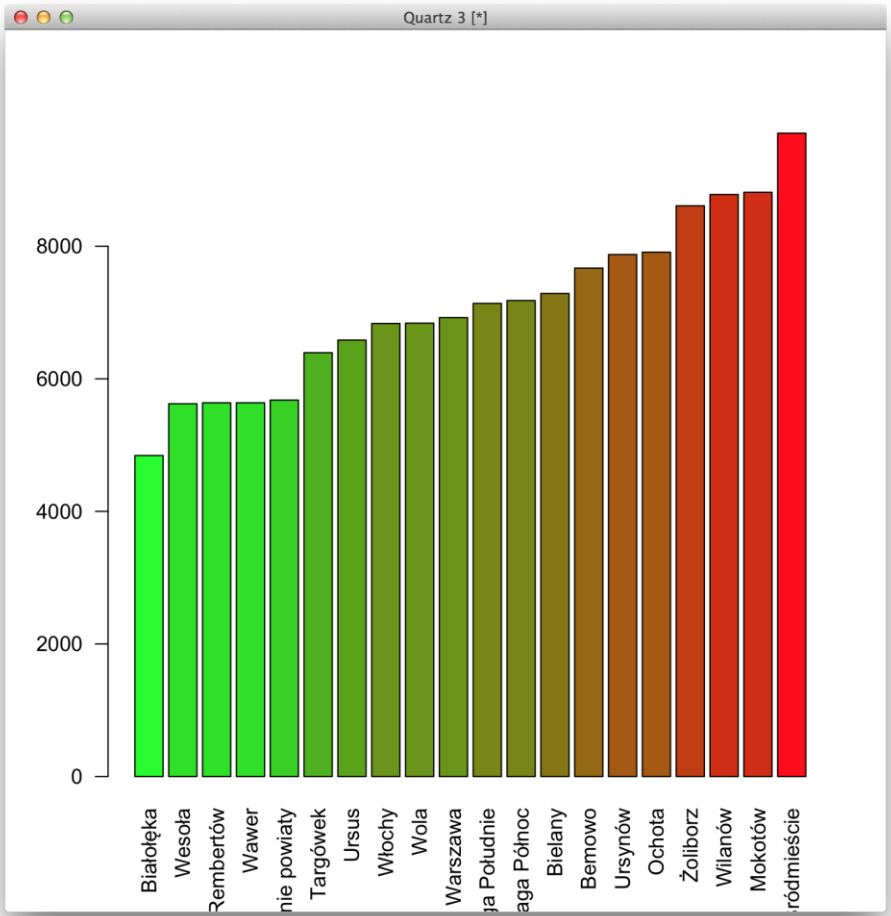
CASE: wizualizacja danych przestrzennych - wstęp



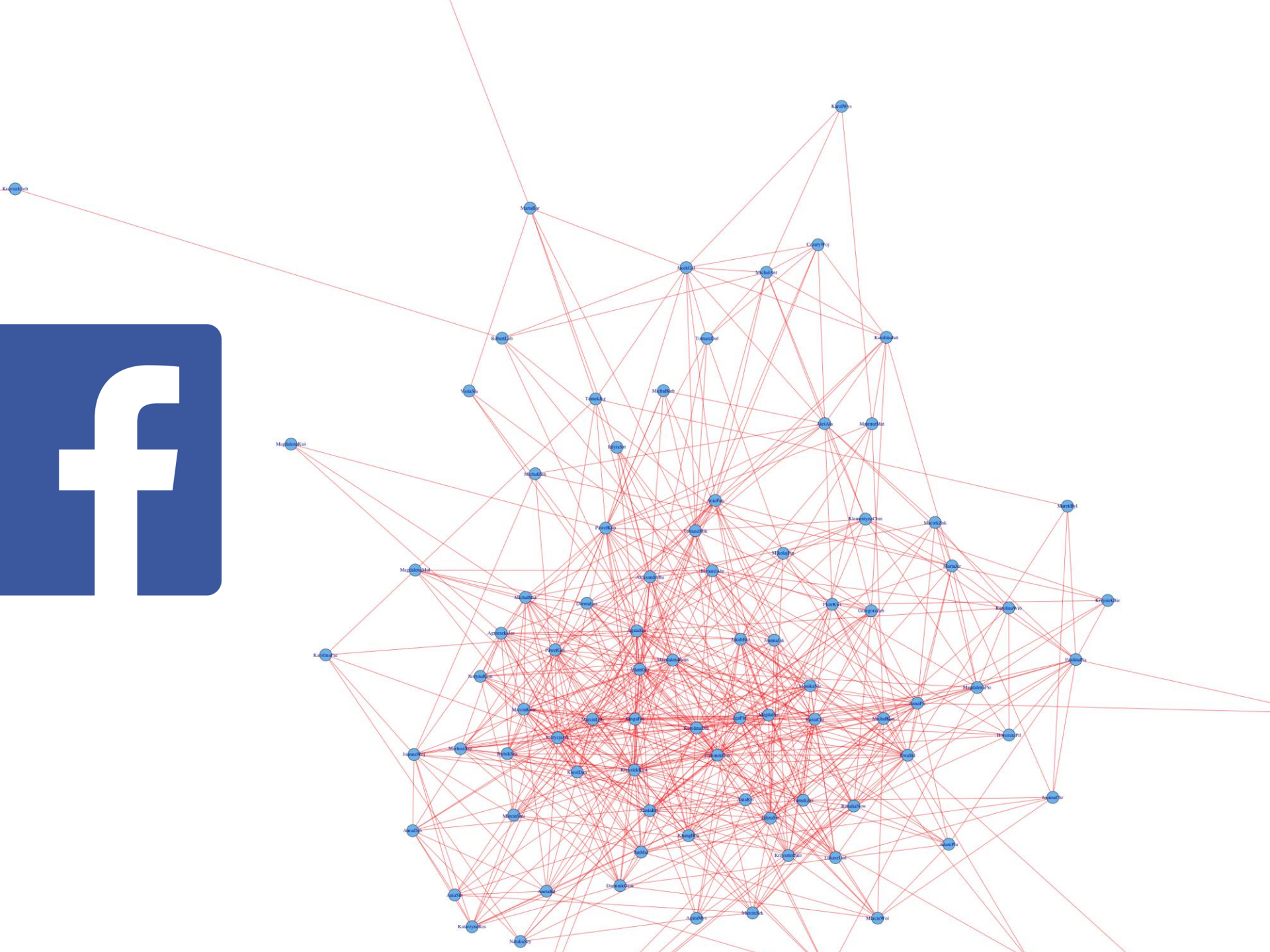
`Waw_map.r`

Mapa, to plik SVG wzięty z Wikimedia, trzeba go przekształcić do formaty PS. Można z pomocą

Naniesienie cen na mapę:



`map_by_prices_colored.r`



CASE: Facebook graph

- Jeśli chce się pracować z własnymi danymi, to trzeba skorzystać z aplikacji NETVIZZhttps://apps.facebook.com/netvizz/ i po udzieleniu jej uprawnień w sekcji Step2 kliknąć na here, a potem zapisać plik GDF, który należy podzielić na dwa pliki, jeden z osobami, a drugi z relacjami miedzy nimi.
- W dostarczonym katalogu /fb znajdują się już gotowe pliki z moimi znajomymi z Facebook'a

- Pliki fb_graph_poor.R pokazuje słaby styl programowania w R bez wykorzystania możliwości jakie dają biblioteki do pracy z data framem, zamiast tego itereując po pojedyńczych wierszach

- Pliki fb_graph_better.R stara się poprawić wspomniane powyżej słabości korzystając min. z biblioteki dapply do modyfikowania data frameu jak również z gsub do skorzystania z mocy wyrażeń regularnych przy skracaniu nazwisk

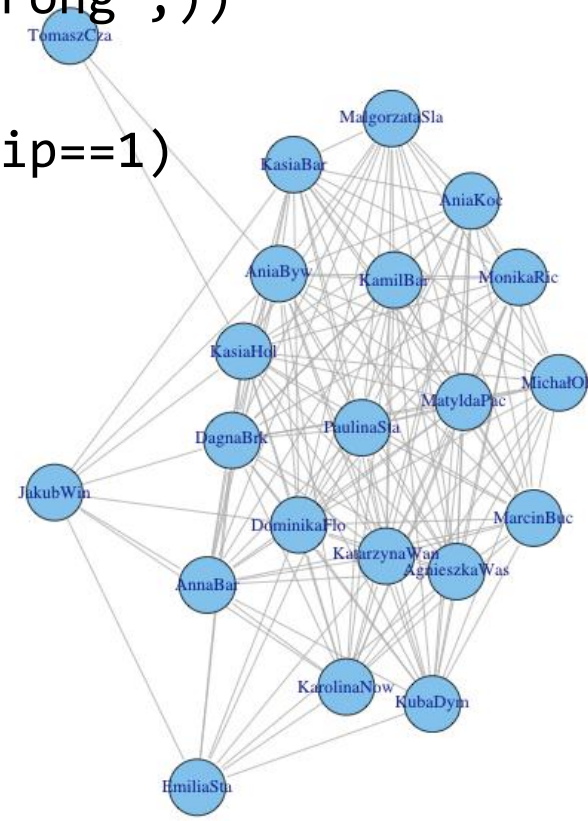
{ fb_graph_poor.R }

{ fb_graph_better.R }

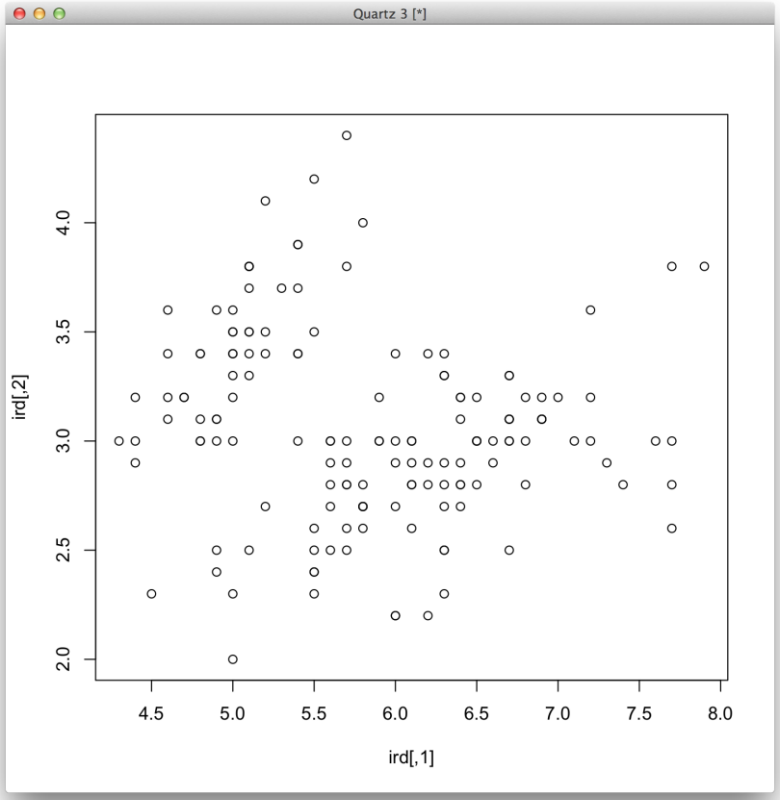
http://kostanski.wordpress.com/2012/10/13/sieci-spolecznosciowe-facebook-praca-z-grafem-znajomych-wizualizacja-szukanie-najwiekszej-kliki/

igraph – badanie właściwości.

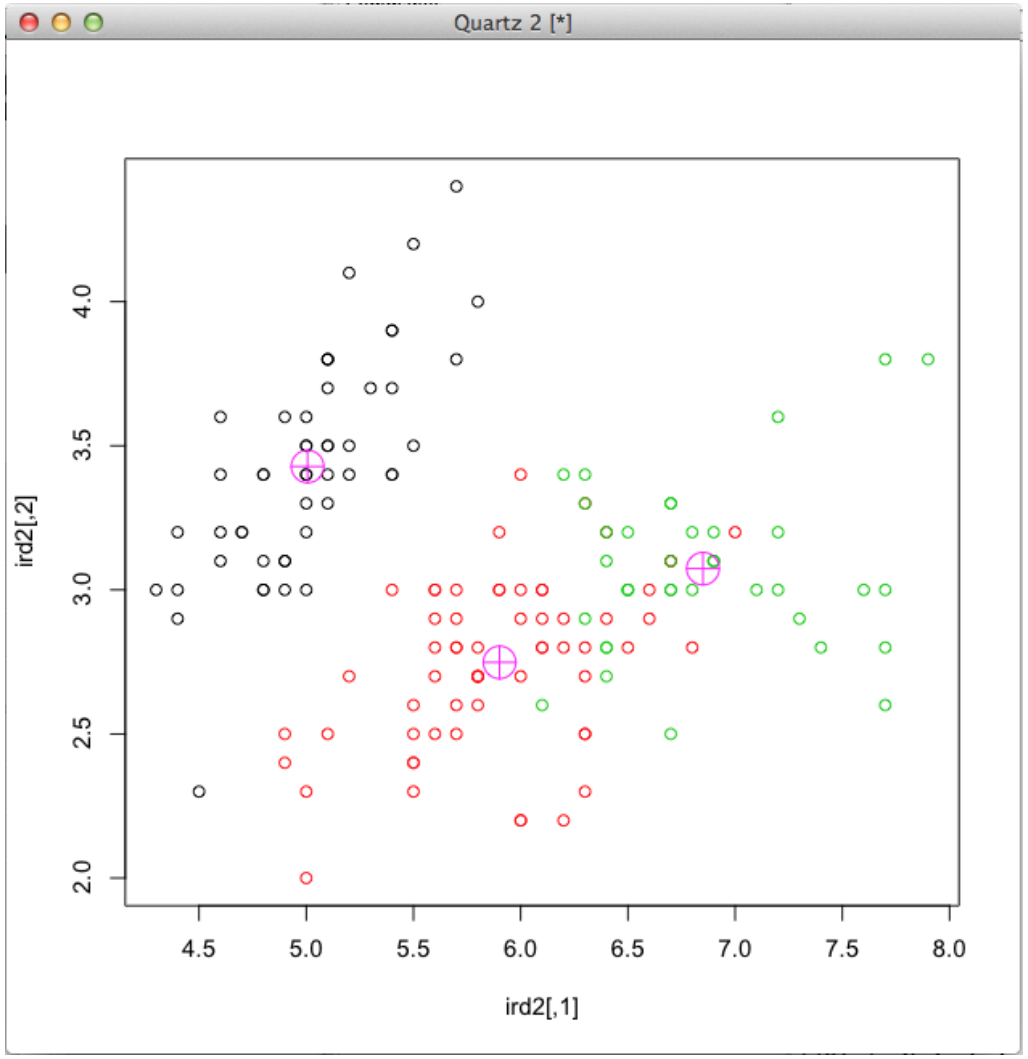
- `cliques(g, min=5)`
- `largest.cliques(g)`
- `is.connected(g)`
- `klastry<-clusters(g, mode=c(“weak”, “strong”;))`
- `table(klastry$membership)`
- `g1<-induced.subgraph(g,klastry$membership==1)`



CASE: k-means

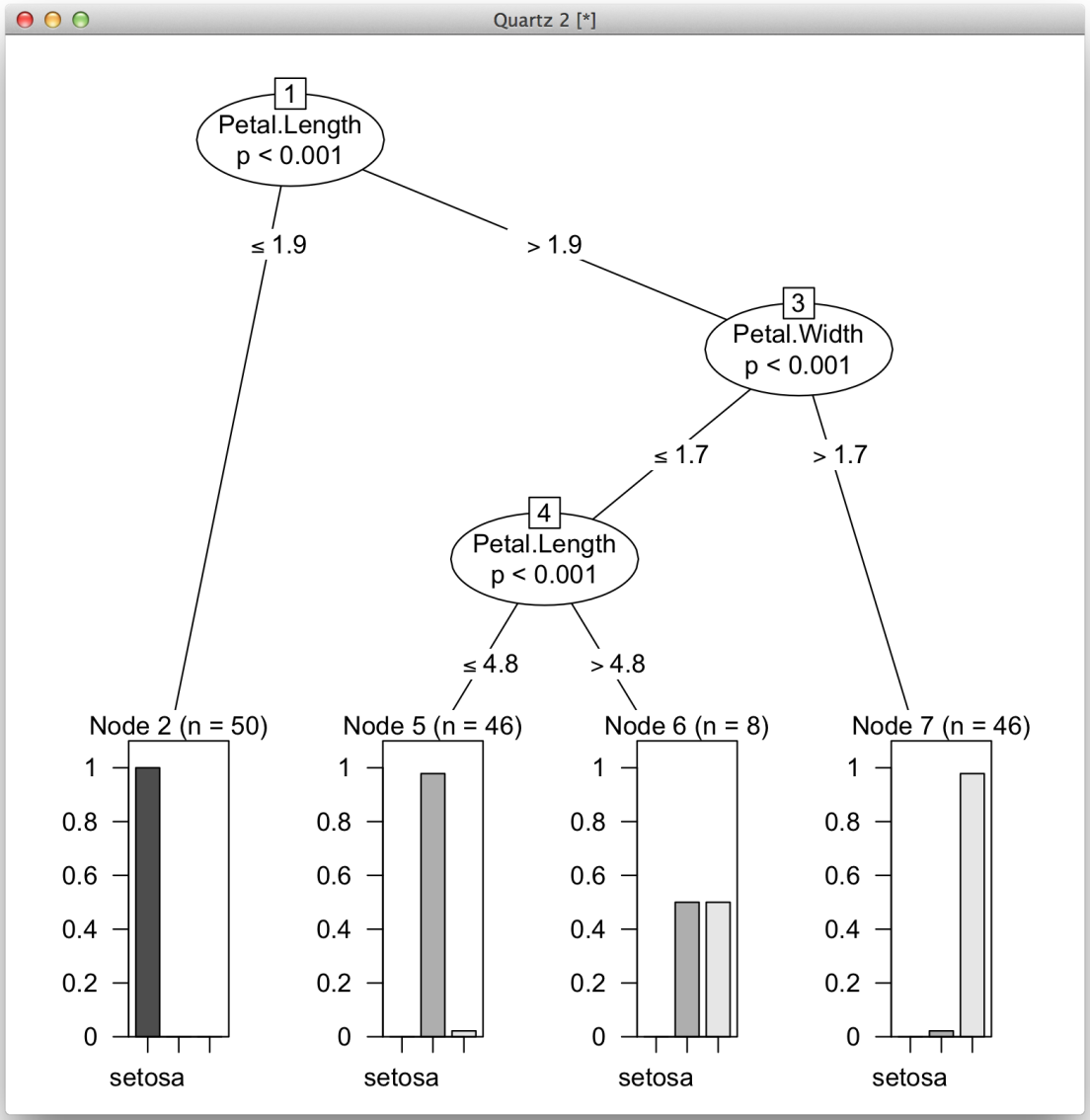


K-means.r



http://kostanski.wordpress.com/2012/10/11/macierze-w-r-oraz-algorytm-kmeans-k-srednich-uczenie-bez-nadzoru/

CASE: drzewo decyzyjne



{ decision-tree.r }

Bibliografia:

- Comprehensive R Archive Networkhttp://cran.r-project.org/
- R doc: http://cran.r-project.org/doc/manuals/R-admin.pdf
- 1h7min wprowadzenia do R by Google: http://www.youtube.com/playlist?list=PLOU2XLYxmsIK9qQfztXeybpHvru-TrqAP
- MYSQL in R: http://www.r-bloggers.com/accessing-mysql-through-r/
- Batch mode: https://www.inkling.com/read/r-cookbook-paul-teetor-1st/chapter-3/recipe-3-13
- Don't use foR:http://nsaunders.wordpress.com/2010/08/20/a-brief-introduction-to-apply-in-r/
- http://scikit-learn.org/stable/
- http://matplotlib.org/
- http://pandas.pydata.org/
- http://docs.scipy.org/doc/numpy/reference/
- http://docs.scipy.org/doc/scipy/reference/