

Raport pracy projektowej nr 2

Mariola Bartosik

Wstęp

W raporcie przedstawię wyniki swoich badań prowadzonych na ramkach danych, dostępnych na stronie internetowej <http://www.gagolewski.com/resources/data/>. Korzystałam z danych przedstawionych poniżej.

```
options(stringsAsFactors=FALSE)
Tags <- read.csv("~/R/travel_stackexchange_com/Tags.csv")
Badges <- read.csv("~/R/travel_stackexchange_com/Badges.csv")
Comments <- read.csv("~/R/travel_stackexchange_com/Comments.csv")
Posts <- read.csv("~/R/travel_stackexchange_com/Posts.csv")
Users <- read.csv("~/R/travel_stackexchange_com/Users.csv")
Votes <- read.csv("~/R/travel_stackexchange_com/Votes.csv")
PostLinks <- read.csv("~/R/travel_stackexchange_com/PostLinks.csv")
```

Każde zadanie wykonane jest na 4 sposoby, za pomocą

- funkcji bazowych R,
- funkcji z biblioteki sqldf,
- funkcji z biblioteki dplyr,
- funkcji z biblioteki data.table.

Dodatkowo porównam czasy wykonania napisanych przeze mnie funkcji przy użyciu jednego wywołania `microbenchmark::microbenchmark()`.

Zadanie 1

Zadanie nr 1 bazowało na ramce danych Posts. Wybrałam z kolumny PostTypeId wiersze, z wartościami równymi 1, z kolumny FavoriteCount wiersze z wartościami większymi równymi 25 oraz z kolumny ViewCount wiersze z wartościami większymi równymi 10000. Na samym końcu wyodrebiłam kolumny o nazwach: "Title, Score, ViewCount, FavoriteCount".

Wyniki w 4 przypadkach były identyczne. Uzyskałam ramkę z czterema kolumnami i dziewiętnastoma wierszami.

```
head(df_base_1(), 5)
```

```
##
## 9      When traveling to a country with a different currency, how should you take your
## 683                                           How can I do a "broad" search for fl
## 1132                                           Tactics to avoid getting harassed by corrupt p
## 2394                                           How to avoid drinking
## 2788                                           Flight tickets: buy two weeks before even during holiday se
##      Score ViewCount FavoriteCount
## 9      136      16838              35
## 683     95      33554              49
## 1132    156      13220              42
## 2394    149      15197              29
## 2788    109      49440              36
```

Porównałam czasy wykonania napisanych przeze mnie funkcji w tym zadaniu przy użyciu wywołania `microbenchmark::microbenchmark()`.

Zadanie 2

Zadanie nr 2 polegało na wybraniu odpowiednich wartości z kolumny OwnerUserId (różnych od -1) z ramki Tags. Korzystałam dwukrotnie z `inner join`.

Złączenie tego typu zachowuje tylko wiersze występujące w obydwu złączonych zbiorach.

Złączeniu uległy kolumny WikiPostId oraz Posts z ramek Tags i Posts, a także kolumny AccountId i OwnerUserId z ramek Tags i Users. Posortowałam malejaco kolumny powstała po zliczeniu oraz wybrałam kolumny TagName, Count, OwnerUserId, Age, Location, DisplayName.

```
head(df_base_2(), 5)
```

```
##      TagName Count OwnerUserId Age      Location      DisplayName
## 711    canada   802         101  34    Mumbai, India          hitec
## 740   europe   681         583  35 Philadelphia, PA      Adam Tuttle
## 751 visa-refusals 554        1737  34    New York, NY Benjamin Pollack
## 710  australia   411         101  34    Mumbai, India          hitec
## 712 new-york-city 204         101  34    Mumbai, India          hitec
```

Zadanie 3

Zadanie nr 3 polegało na pogrupowaniu względem RelatedPostId. Zliczyłam wartości w RelatedPostId i zapisałam kolumnę z wartościami zliczenia jako NumLinks. Poza tym należało zmienić nazwę RelatedPostId na PostId, a także zapisać wynik w postaci ramki danych o nazwie RelatedTab. Kolejnym krokiem było złączenie z ramek danych RelatedTab i Posts kolumn PostId i Id zachowując wiersze występujące w obydwu kolumnach. Następnie wybrałam wartości w kolumnie PostTypeId równe 1. Posortowałam malejąco kolumnę NumLinks oraz wybrałam poszczególne kolumny takie jak TagName, Count, OwnerUserId, Age, Location, DisplayName.

```
head(df_base_3(), 5)
```

```
##
## 2262          Is there a way to find out if I need a transit visa for a layover in
## 2165          Do I need a visa to transit (or layover) in the Schengen
## 1168          Should my first trip be to the country which issued my Schengen
## 3855 Should I submit bank statements when applying for a UK Visa? What do they say ab
## 1050    How much electronics and other valuables can I bring duty-free when going to
##      NumLinks
## 2262      594
## 2165      585
## 1168      331
## 3855      259
## 1050      197
```

Zadanie 4

Zadanie nr 4 na początku polegało na znalezieniu odpowiednich wartości Name w ramce Badges. Takich, dla których dane w kolumnie Class mają wartość równą 1. Pogrupowałam ramkę względem Name. Wybrałam wartości po zliczeniu większe od dwóch i mniejsze od 10. Potem wybrałam tylko te wartości Name z Users, które dostaliśmy po wyselekcjowaniu z Badges. Także, należało wybrać dane, dla których wartość z kolumny Class jest równa 1. Zapisałam to jako ValuableBadges. Złączyłam tak jak wyżej z ramek Users i Badges odpowiednio kolumny UserId i Id. Na samym końcu wybrałam odpowiednio unikalne wartości z kolumn Id, DisplayName, Reputation, Age, Location.

```
head(df_base_4(), 5)
```

```
##      Id      DisplayName Reputation Age      Location
## 1   19      VMAtm      18556   33      Tampa, FL, United States
## 2  101      Mark Mayo   121667   37 Sydney, New South Wales, Australia
## 4  108  Ankur Banerjee   31273   27      London, UK
## 6  466  iHaveacomputer    8360  NA      Down underer
## 7  693 RoflcoptrException  33300  NA
```

Zadanie 5

Zadanie nr 5 polegało na wybraniu z ramki Votes kolumny PostId, gdzie wartości w kolumnie VoteTypeId są równe 2. Pogrupowałam PostId. Zliczyłam wartości i zapisałam jako UpVotes. Powstała nowa ramka, którą zapisałam jako UpVotesTab. Analogicznie postepowałam przy tworzeniu DownVotesTab, gdzie zliczone wartości zapisałam jako DownVotes, a wcześniej z kolumny VoteTypeId wybrałam wartości równe 3. Złączyłam za pomocą left join kolumny PostId z obu ramek. Złączenie tego typu wybiera każdy wiersz z pierwszego zbioru, dołączając do niego wszystkie pasujące wiersze z drugiego zbioru występujące w lewym zbiorze. Wybrałam kolumny PostId, UpVotes, DownVotes, gdzie kolumnie DownVotes wszystkie wartości nieokreślone zostały zamienione zerem.

```
head(df_base_5(), 5)
```

##	PostId	UpVotes	DownVotes
## 1	1	10	2
## 2	2	32	0
## 3	3	13	1
## 4	4	9	1
## 5	5	14	0

Zadanie 6

Zadanie nr 6 było podobne do zadania 5, różnica, polegała na kolejności w złączeniach. Przez co dostałam dwie różne ramki danych. Następnie należało połączyć je i wybrać kolumny PostId i kolumnę, która powstała po odjęciu UpVotes od DownVotes.

```
head(df_sql_6(), 4)
```

```
## Warning: package 'sqldf' was built under R version 3.6.3
## Loading required package: gsubfn
## Warning: package 'gsubfn' was built under R version 3.6.3
## Loading required package: proto
## Warning: package 'proto' was built under R version 3.6.3
## Loading required package: RSQLite
## Warning: package 'RSQLite' was built under R version 3.6.3
```

##	PostId	Votes
## 1	1	8
## 2	2	32
## 3	3	12
## 4	4	8