

Social network analyzes

kalganov maxim

23/11/2020

Функции для получения друзей из vk

```

api_version_checker <- function(v){
  current_version <- "5.73"

  if(is.null(v)) return(current_version)
  if(as.numeric(v) < 4) return(current_version)
  return(v)
}

vkGetUserFriends <- function(user_id = NULL,
                             access_token = NULL,
                             api_version = NULL,
                             name_case = "nom"){

  if(is.null(access_token)){
    stop("Set access_token in options, is require.")
  }

  api_version <- api_version_checker(api_version)
  # result frame
  result <- data.frame(stringsAsFactors = F)

  # query
  query <- paste0("https://api.vk.com/method/friends.get?", ifelse(is.null(user_id), "",
    paste0("user_id=", user_id)), "&hints&count=10000&fields=nickname, domain, sex, bdate, cit
y, country, timezone, photo_50, photo_100, photo_200_orig, has_mobile, contacts, education, on
line, relation, last_seen, status, can_write_private_message, can_see_all_posts, can_post, u
niversities&name_case", name_case, "&access_token=", access_token, "&v=", api_version)
  answer <- GET(query)
  stop_for_status(answer)
  dataRaw <- content(answer, "parsed", "application/json")

  # check for error
  if(!is.null(dataRaw$error)){
    stop(paste0("Error ", dataRaw$error$error_code, " - ", dataRaw$error$error_msg))
  }
  # parsing
  all_response <- dataRaw$response$items
  all_colnames <- c(
    "user_id", "first_name", "last_name", "gender", "nickname", "domain",
    "bdate", "city", "country", "photo_50", "photo_100", "photo_200_orig",
    "has_mobile", "online", "can_post", "can_see_all_posts",
    "can_write_private_message", "home_phone", "status", "last_seen_time",
    "last_seen_platform", "university", "university_name", "faculty",
    "faculty_name", "graduation", "education_form", "education_status",
    "relation")

  for(i in 1:dataRaw$response$count){
    new_friend <- data.frame(user_id = ifelse(is.null(all_respo
nse[[i]]$id), NA, all_response[[i]]$id),
                             first_name = ifelse(is.null(all_respo
nse[[i]]$first_name), NA, all_response[[i]]$first_name),
                             last_name = ifelse(is.null(all_respo
nse[[i]]$last_name), NA, all_response[[i]]$last_name),
                             gender = ifelse(is.null(all_respo
nse[[i]]$sex), NA, all_response[[i]]$sex),
                             nickname = ifelse(is.null(all_respo
nse[[i]]$nickname ), NA, all_response[[i]]$nickname),

```

```

        domain = ifelse(is.null(all_respo
nse[[i]]$domain), NA,all_response[[i]]$domain),
        bdate = ifelse(is.null(all_respo
nse[[i]]$bdate), NA,all_response[[i]]$bdate),
        city = ifelse(is.null(all_respo
nse[[i]]$city), NA,all_response[[i]]$city),
        country = ifelse(is.null(all_respo
nse[[i]]$country), NA,all_response[[i]]$country),
        photo_50 = ifelse(is.null(all_respo
nse[[i]]$photo_50), NA,all_response[[i]]$photo_50),
        photo_100 = ifelse(is.null(all_respo
nse[[i]]$photo_100), NA,all_response[[i]]$photo_100),
        photo_200_orig = ifelse(is.null(all_respo
nse[[i]]$photo_200_orig), NA,all_response[[i]]$photo_200_orig),
        has_mobile = ifelse(is.null(all_respo
nse[[i]]$has_mobile), NA,all_response[[i]]$has_mobile),
        online = ifelse(is.null(all_respo
nse[[i]]$online), NA,all_response[[i]]$online),
        can_post = ifelse(is.null(all_respo
nse[[i]]$can_post), NA,all_response[[i]]$can_post),
        can_see_all_posts = ifelse(is.null(all_respo
nse[[i]]$can_see_all_posts), NA,all_response[[i]]$can_see_all_posts),
        can_write_private_message = ifelse(is.null(all_respo
nse[[i]]$can_write_private_message), NA,all_response[[i]]$can_write_private_message),
        home_phone = ifelse(is.null(all_respo
nse[[i]]$home_phone), NA,all_response[[i]]$home_phone),
        status = ifelse(is.null(all_respo
nse[[i]]$status), NA,all_response[[i]]$status),
        last_seen_time = ifelse(is.null(all_respo
nse[[i]]$last_seen$time), NA,all_response[[i]]$last_seen$time),
        last_seen_platform = ifelse(is.null(all_respo
nse[[i]]$last_seen$platform), NA,all_response[[i]]$last_seen$platform),
        university = ifelse(is.null(all_respo
nse[[i]]$university), NA,all_response[[i]]$university),
        university_name = ifelse(is.null(all_respo
nse[[i]]$university_name), NA,all_response[[i]]$university_name),
        faculty = ifelse(is.null(all_respo
nse[[i]]$faculty), NA,all_response[[i]]$faculty),
        faculty_name = ifelse(is.null(all_respo
nse[[i]]$faculty_name), NA,all_response[[i]]$faculty_name),
        graduation = ifelse(is.null(all_respo
nse[[i]]$graduation), NA,all_response[[i]]$graduation),
        education_form = ifelse(is.null(all_respo
nse[[i]]$education_form), NA,all_response[[i]]$education_form),
        education_status = ifelse(is.null(all_respo
nse[[i]]$education_status), NA,all_response[[i]]$education_status),
        relation = ifelse(is.null(all_respo
nse[[i]]$relation), NA,all_response[[i]]$relation),
        stringsAsFactors = F)
    colnames(new_friend) <- all_colnames
    result <- rbind(result, new_friend)
  }
  # convert date
  result$last_seen_time <- as.POSIXct(as.integer(result$last_seen_time), origin="1970
-01-01")

  # return result

```

```
    return(result)
  }
```

Анализ сообществ

Получение данных для анализа сообществ

```
my_vk_communities <- vkGetUserGroups(user_id = my_tok$user_id,
                                     access_token = my_tok$access_token)
my_vk_friends <- vkGetUserFriends(user_id=my_tok$user_id,
                                  access_token = my_tok$access_token)
exceptions <- c(139575609, 197310316)
my_vk_friends <- subset(my_vk_friends, (first_name != "DELETED") & !(user_id %in% exceptions))

communities_to_analyze <- my_vk_communities[, c("name", "screen_name")]
communities_to_analyze$count <- 0

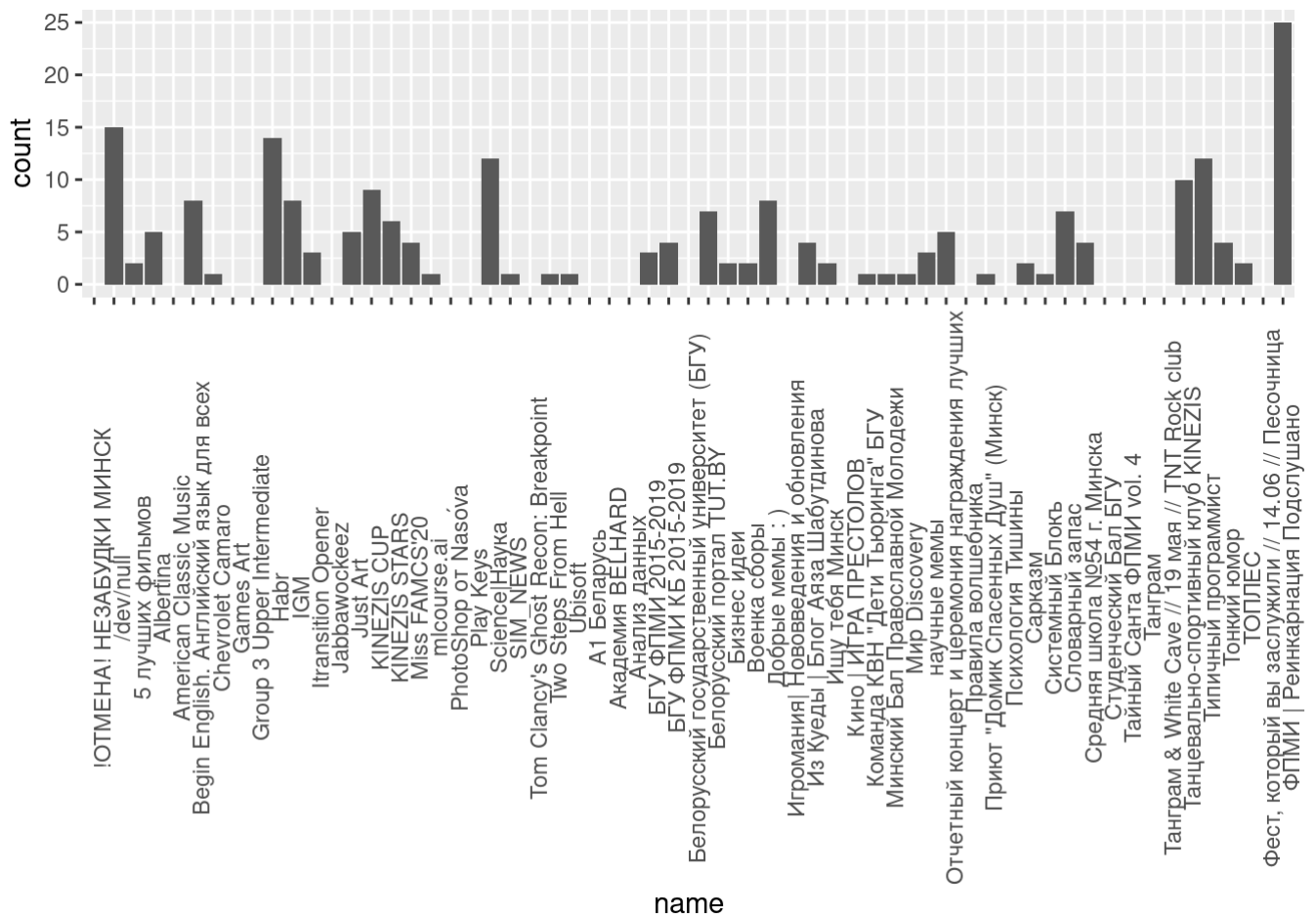
for(i in 1:nrow(my_vk_friends)){
  friend_communities <- vkGetUserGroups(user_id = my_vk_friends[i,"user_id"],
                                       access_token = my_tok$access_token)
  friend_community_names <- friend_communities$screen_name
  communities_to_analyze[communities_to_analyze$screen_name %in% friend_community_names, c("count")] %+=% 1
}
```

Визуализация результатов анализа сообществ

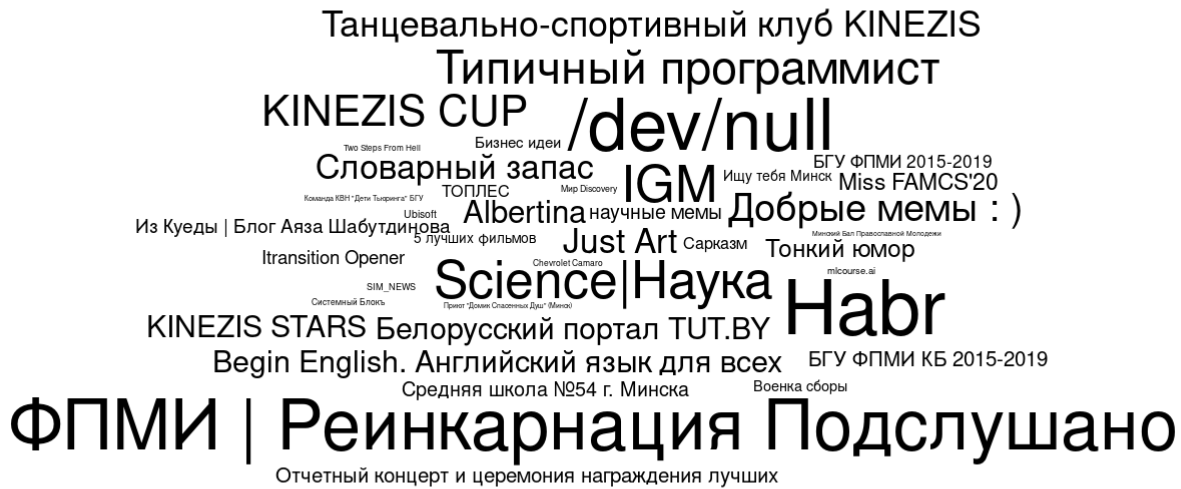
На графиках отображена информация о том, какие мои группы есть у моих друзей.

Первый график показывает сами данные, а на втором графике эта информация предоставлена в более удобном виде. Из него можно сделать вывод, что большинство моих друзей в вк - это знакомые из БГУ и из среды программистов. Вторая большая группа - знакомые из танцевальной среды.

```
ggplot(communities_to_analyze, aes(x = name, y = count)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 90))
```



```
set.seed(42)
ggplot(communities_to_analyze, aes(label = name, size = count, fill=count)) +
  geom_text_wordcloud_area(rm_outside = TRUE) +
  scale_size_area(max_size = 10) +
  theme_minimal()
```



Анализ графа связей среди друзей.

Получение данных для анализа связей друзей.

```
full_df <- NA
for(i in 1:nrow(my_vk_friends)){
  cur_friend_friends <- vkGetUserFriends(user_id=my_vk_friends[i, "user_id"],
                                         access_token = my_tok$access_token)

  cur_friend_friends <- subset(cur_friend_friends, (user_id %in% my_vk_friends$user_id) & (first_name != "DELETED") & !(user_id %in% exceptions))
  if(nrow(cur_friend_friends) > 0){
    cur_friend_friends$from <- my_vk_friends[i, "last_name"]
    cur_friend_friends$to <- cur_friend_friends$last_name
    cur_df <- cur_friend_friends[, c("from", "to")]
    if(is.na(full_df)){
      full_df <- cur_df
    }else{
      full_df <- rbind(full_df, cur_df)
    }
  }
}

full_graph <- graph_from_data_frame(full_df)
```

Визуализация результатов анализа связей друзей

На графиках отображена информация о том, какие мои друзья также знают друг друга.

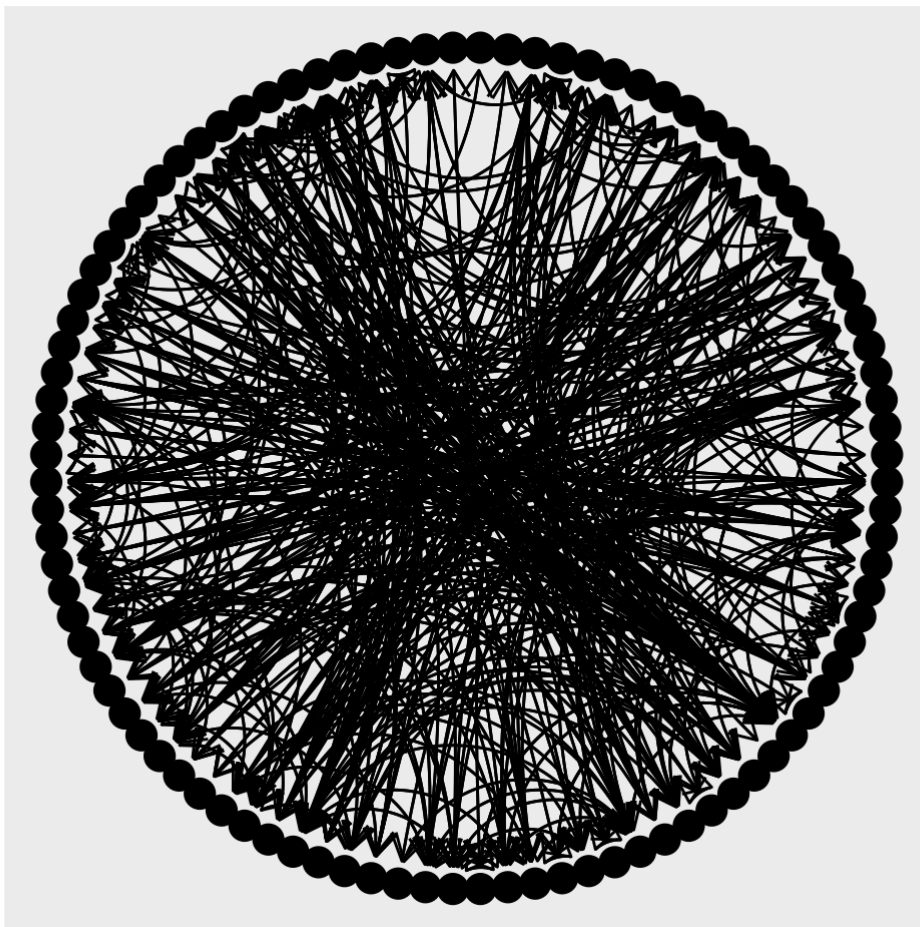
Первый график был попыткой отображения графа в форме круга, когда каждый человек находится на окружности. Я считал, что этот вид должен в наиболее удобном виде показать все связи, но я недооценил количество связей. График оказался нечитабельным.

Второй график уже построен в свободной форме и в связи с этим имеет следующие преимущества и недостатки:

- преимуществом является то, что можно выделить социальные группы людей, к которым относятся мои друзья
- недостатком является то, что большинство имен нечитабельно. Хотя много имен находятся в куче, это не мешает мне как человеку, который знает всех этих людей, определить основные социальные группы.

На графике видны следующие группы: друзья из университета, друзья из танцевальной среды, друзья со школы, а также двое знакомых, с которыми я пересекся в Америке.

```
ggraph(full_graph, layout = 'linear', circular = TRUE) +  
  geom_edge_arc(arrow = arrow(length = unit(4, 'mm')),  
               start_cap = circle(3, 'mm'),  
               end_cap = circle(3, 'mm')) +  
  geom_node_point(size = 5) +  
  coord_fixed()
```



```
ggraph(full_graph, layout = 'graphopt') +  
  geom_edge_link() +  
  geom_node_text(aes(label = name))
```

