МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**КУРСОВАЯ РАБОТА**

**АДАПТИВНАЯ ВИЗУАЛИЗАЦИЯ НЕСТРУКТУРИРОВАННЫХ ГЕОКОДИРОВАННЫХ ДАННЫХ**

Работу выполнил А.С. Ролдугин

(подпись, дата) (инициалы, фамилия)

Факультет компьютерных технологий и прикладной математики 3 курс

Направление 02.03.02 – «Фундаментальная информатика и информационные технологии»

Научный руководитель

доц., канд.т.н. Т.А. Приходько

(подпись, дата) (инициалы, фамилия)

Нормоконтролер

ассистент, А.А. Климец

(подпись, дата) (инициалы, фамилия)

Краснодар 2020

СОДЕРЖАНИЕ

Введение…………….…………………………………………………..……...........3

# 1 Адаптивная визуализация неструктурированных геокодированных

# данных– формализация задачи …………………………..……………….5

# 2 Методы решения задач визуализации геоданных..…………………….….6

2.1 Стек технологий подготовки геоданных…..……....……………………...6

2.2 Язык R для анализа и визуализации геоданных….. .…………………….7

2.3 Инструменты анализа геоданных………………………………………...8

3 Методы взаимодействия с информационной системой ……...…………………9

3.1 Выборка, извлечение и обработки потоковых данных….……….............10

3.2 Сбор метаданных социальных медиа …….……..….…………...…….....11

3.3 Поиск и сохранение результатов в базе данных..….…….….……….…..12

3.4 Фильтрация результатов выборки данных…………………………..13

4 Визуализация геокодированных данных..……………..………………………..13

4.1 Временные ряды …….……………………..……...……………..……….14

4.2 Создание макета карты…………..……………………………...................15

Заключение………………………………..………………………….....……….19

Список использованных источников...………..………………….….……….20

Приложение А………………………………………………………………… 21

# 

ВВЕДЕНИЕ

Анализ данных социальных сетей (social network data analysis) – одно из современных направлений всестороннего исследования. Интерес исследователей к этому направлению связан с тем, что оно предоставляет новый набор объяснительных моделей и аналитических инструментальных средств, которые находятся вне рамок обычных количественных и вычислительных методов.

В данной работе реализуется технология адаптивной визуализации геоданных. Эта технология может быть применена в современных геосоциальных сервисах, а также для решения различных задач, связанных с визуализацией и обработкой геоинформации пользователями социальных медиа. К особенностям адаптивной геовизуализации в социальных медиа относятся восприятия визуальной информации, контекст задачи, средства отображения геоданных. Адаптивная визуализация геоданных позволяет упростить визуальный анализ сложных геоизображений для конечного пользователя социальных медиа.

Также применение технологии адаптивной геовизуализации позволяет повысить эффективность, скорость и легкость анализа различных наборов данных. При этом уменьшается общая когнитивная нагрузка на потребителя информации. Исследуемая технология адаптивной визуализации геоданных может быть использована для разработки и реализации новых геосоциальных сервисов комплексного анализа данных в сфере социальных медиа и прочих сервисов со значительным объемом уже готовых данных так и потоковых данных из различных источников.

Целью данной работы была разработка модульной системы сбора и визуализации геоданных социальной сети. Для решения поставленной задачи была разработана комплексная система, взаимодействия с API цифровой информационной системы, извлечение, очистка, обработка и визуализация потоковых геокодированных данных отдельно выбранной социальной сети.

Курсовая работа состоит из четырех глав.

Первая глава работы вводит понятие визуализации геоданных и формализует задачу рассматриваемую в данной работе.

Вторая глава данной работы посвящена методам анализа и визуализации геоданных.

В третьей главе рассматривается стек технологий подготовки геоданных.

В четвертой главе рассмотрим данные в разрезе времнных рядов, создадим макет карты и визуализируем геокодированные данные на конктретном примере.

# 1 Адаптивная визуализация неструктурированных геокодированных данных **– формализация задачи**

Предметом исследования являются методы и технология анализа неструктурированных геоданных социальных медиа. Распространённые и широко известные социальные сети представляют собой совокупности социальных акторов и набора связей между ними. В качестве социальных акторов могут выступать индивиды, социальные группы, организации, города, страны. Под связями понимаются не только коммуникационные взаимодействия между акторами, но и связи по обмену различными ресурсами и деятельностью, включая эмоциональные, экономические, политические и культурные отношения. Полученная сеть взаимодействий может быть проанализирована различными методами теории графов, теории информации, математической статистики и математического моделирования.

Таким образом социальная сеть – это ни что иное, как объединение социальных позиций (акторов) и их связей, с математической точки зрения – это социальный граф. Важнейшими характеристиками социальной сети как абстрактной математической модели являются узлы – собственно, участники коммуникационного процесса – или вершины социального графа и связи между ними – ребра социального графа.

Основные методы анализа данных социальных сетей - методы теории графов, в частности, направленные графы и представляющие их матрицы, применяемые для изучения структурных взаимосвязей актора; методы нахождения локальных свойств субъектов, например, центральности, престижа, положения, принадлежности к некоторым подгруппам; методы определения эквивалентности акторов, включая их структурную эквивалентность; блоковые модели и ролевые алгебры; анализ диад и триад; вероятностные модели, включая модели процессов. Далее мы рассмотрим вышеперечисленные методы и приведем примеры содержательных результатов, полученных с их помощью.

# 2 Методы решения задач визуализации геоданных

В качестве основного был выбран метод для решения задач визуализации различных геоданных, при котором последовательно решаются следующие задачи:

- сбор данных об объектах визуализации;

- пространственная локализация данных;

- создание дизайна карты;

- выбор и реализация технологии визуализации.

В рамках реализации данной работы были исследованы и отобраны для применения следующие методы:

- извлечение геоданных из цифровых информационных систем, не обладающих средствами экспорта;

- пространственная идентификация объектов визуализации;

- нанесение объектов пространственной визуализации на карту.

Помимо этого, в работе были исследованы и применены методы и соответствующие им технологии динамической кластеризации пространственных объектов. В основу выбора методов и технологий решения задачи визуализации геоданных был положен принцип их доступности. При этом упор был сделан на выборе свободно распространяемых технологических решений.

# 2.1 Стек технологий подготовки геоданных

Ключевой потенциал систем анализа данных кроется в извлечении полезных выводов из данных. Цель данного процесса состоит в том, чтобы ответить на интересующие вопросы, используя технологии анализа данных, и достичь более полного понимания конкретной предметной области.

Инструменты визуализации данных:

Электронные таблицы (Excel, Google Таблицы, LibreOffice и другие программные продукты).

Интеллектуальные карты (Coggle, MindMup, Mind42, XMind, FreeMind). Графы (Gephi, Cytoscape, Graph Online).

Библиотеки JS (D3, Protovis, jChartFX, amMap, AnyMap, Google GeoCharts, CanvasJS, sigmajs).

R (ggplot2, R Graph Gallery, Plotly, moderndata.plot.ly).

Python (Matplotlib, Ggplot, Pygal, Python Graph Gallery, Plotly, graph-tool). Электронные таблицы имеют стандартный дизайн графиков и диаграмм. Действительно красивую визуализацию можно построить при помощи специализированных онлайн-инструментов.

Библиотеки JS, R и Python предоставляют инструменты для анализа и визуализации информации. Возможности работы с ними ограничены лишь знаниями и опытом программирования.

Для проектов, где имеет значение географическое местоположение, полезными будут точечные карты и карты потоков. Задача визуализации данных состоит не в абстрактной красоте, а в том, чтобы один взгляд на визуализацию позволил сделать корректные выводы.

2.2 Язык R для анализа и визуализации геоданных

Язык программирования R – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Данный язык поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и так далее.

Одной из причин, почему R прекрасно подходит для анализа данных, математических задач и прочих научных расчетов, является то, что R – это удобное и выразительное средство связи между читабельным кодом и быстрыми библиотеками, написанными на С\C++.

Другая причина заключается в том, что на данный момент существует много реализаций различных методов анализа данных, соответствующих библиотек и прочих вспомогательных инструментов как для анализа данных и научных вычислений, так и для визуализации геокодированных данных.

Язык R лучше в ряде существенных пунктов: во-первых, он позволяет достаточно легко и быстро сохранять, делиться, транслировать и выкладывать скрипты подготовки и анализа; во-вторых, он позволяет на порядок более гибко и прозрачно готовить, анализировать и визуализировать данные; в-третьих, у него огромное и всегда готовое прийти на помощь комьюнити, регулярно снабжающее потенциальных исследователей новыми удобными библиотеками и идеями по обработке данных.

# 2.3 Инструменты анализа геоданных

В данной главе рассмотрим стек инструментов программной реализации, используемых для решения задачи разработки модульной системы сбора, анализа и визуализации геоданных социальной сети. Пример архитектуры системы анализа геоданных приведен на рисунке 1.

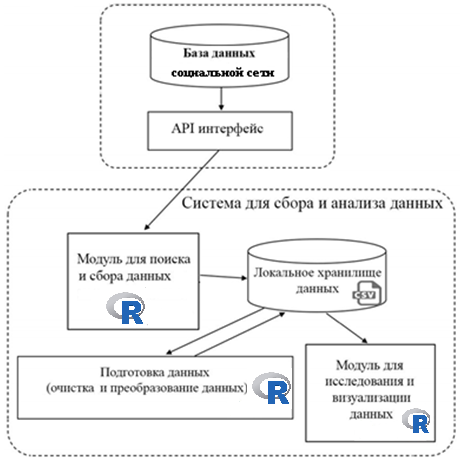


Рисунок 1 – Архитектура системы анализа геоданных

# 3 Методы взаимодействия с информационной системой

Рассмотрим инструменты взаимодействия с API Twitter - одной из популярных социальных сетей. Twitter предоставляет услугу микроблогинга, своего рода электронный журнал, в котором информация следует короткими сообщениями – твитами, размер которых ограничен 140 символами.

Пользователи Twitter обсуждают происходящие события в режиме реального времени, включая торжества, телевизионные сериалы, спортивные состязания, политические и бизнес новости, погодные явления и многое-многое другое.

С учетом разнообразия применений, Twitter представляет собой потенциальный огромный интерес для аналитиков данных. Социальная сеть Twitter предлагает серию прикладных программных интерфейсов (АРI) для доступа к своим данным:

* чтение твитов;
* извлечение информации из профилей пользователей;
* отправку сообщений от имени пользователя.

Чтобы настроить проект для доступа к данным в Twitter, необходимо выполнить два предварительных этапа:

* зарегистрировать приложение;
* выбрать клиента Twitter API.

После регистрации получаем для входа в систему Keys и Access Token (Ключи и Маркеры доступа), а также необходимые для дальнейшего доступа к данным разным данным социальной сети Twitter – Ключ потребителя (Consumer Key) и секретный ключ потребителя Consumer Secret, которые называют API Key и API Secret. В свою очередь уровень доступа Access Level определяет, какие действия может выполнять приложением в Twitter от имени пользователя.

# 3.1 Выборка, извлечение и обработки потоковых данных

Социальная сеть Twitter поддерживает несколько API, которые условно можно разделить на два класса:

- REST API – поисковый;

- STREAMING API – потоковый.

Поисковые REST API позволяют обращаться только к прошлому.

Взаимодействуя с Twitter через REST API, можно искать фактически существующие твиты, опубликованные и доступные для поиска. Эта группа API ограничивают количество твитов, которые можно получить, ограничивая не только частоту запросов, но интервалы времени между запросами. Как правило для доступа открыты данные не более чем недельной давности, то есть более ранние твиты извлечь нельзя. Второй важный недостаток REST API, состоит в том, что не гарантируются возврат абсолютно всех сообщений, которые были опубликованы в Twitter, так как некоторые могут быть недоступны для поиска или еще не были проиндексированы [8].

Потоковый STREAMING API, после установки соединения и условии сохранения HTTP соединения открытым позволяет двигаться вперед по линии времени получая все твиты, соответствующие заданным критериям фильтрации, по мере их публикации.

В общем случае потоковый интерфейс STREAMING API предпочтительнее, когда требуется загрузить большое количество твитов, так как взаимодействие с платформой ограничивается лишь поддержкой соединения открытым. Недостаток такого подхода заключается в его продолжительности, потому что приходится ждать публикации твитов, прежде чем получится их собрать и проанализировать.

В свою очередь REST API удобно использовать, когда требуются для анализа твиты, созданные определенным пользователем, или получить доступ по определенной ленте, а STREAMING API требуется, когда необходимо выполнить фильтрацию по конкретному ключевому слову и скачать связанный с ним массивный объем твитов - реакцию на конкретное событие в реальном времени [6].

# 3.2 Сбор метаданных социальных медиа

Для сбора данных социальной сети Twitter и последующей их обработки необходимо приложение – клиент, реализующее различные формы взаимодействия и обращения к Twitter API.

На первом этапе после регистрации и настройки аутентификации приложения имеем: ключ потребителя, секретный ключ потребителя, маркер доступа и секретный маркер доступа. Сохраним учетные данные в переменных окружения, чтобы отделить прикладную логику от конфигурации.

Для подключения к Twitter из R необходимо установить пакет расширения R — twitteR, необходимый для поиска и сбора данных в Twitter.

Для этого понадобятся данные полученные при регистрации Keys and Access Tokens (API Key, API Secret, Access Token и Access Token Secret) нашего Twitter-приложения в разделе Twitter Developer. Их мы передадим функции setup\_twitter\_oauth () из пакета twitteR, которая непосредственно выполняет аутентификацию.

Введем в командном окне R:

- library(twitteR);

- api\_key <- "API KEY";

- api\_secret <- "API SECRET";

- access\_token <- "ACCESS TOKEN";

- access\_token\_secret <- "ACCESS TOKEN SECRET";

- setup\_twitter\_oauth (api\_key, api\_secret, access\_token, access\_token\_secret).

Процедура аутентификация в Twitter API приведена на рисунке 2:

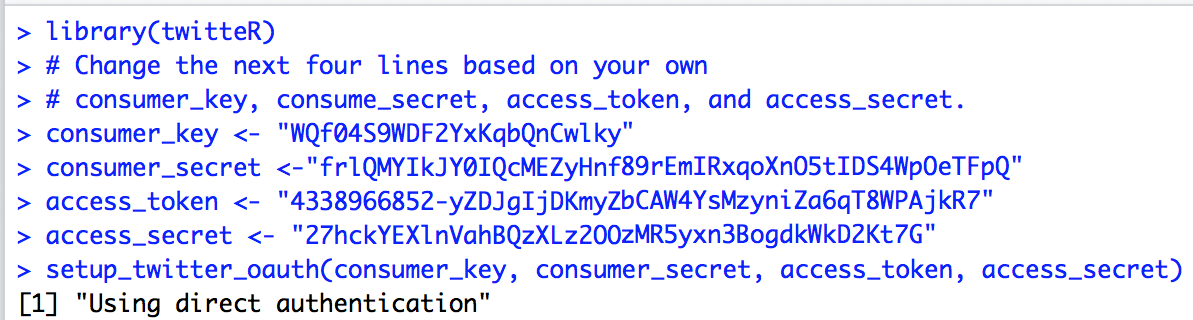


Рисунок 2 – Аутентификация в Twitter API

После того как R сигнализирует об успешном подключении к Twitter API, приступаем к поиску и загрузке данных [7].

3.3 Поиск и сохранение результатов в базе данных

Поиск в Twitter выполняется функцией searchTwitter () с единственным обязательным аргументом – строкой ключевых слов.

Далее необходимо создать базу в текущем каталоге.

register\_sqlite\_backend("data\_mining.sqlite")

По заданным ключевым словам, осуществим поиск необходимых для исследования твитов: tweets = searchTwitter("XXXXX")

Сохраним твиты в базе: store\_tweets\_db(tweets)

Произведем загрузку твитов из базы данных в таблицу R:

from\_db = load\_tweets\_db(as.data.frame = TRUE)

Функция searchTwitter () позволяет задать количество возвращаемых твитов (n), их язык (lang), временной интервал (since – “от” и until – “до”), расположение автора твита (geocode) и другие параметры. Кроме того, аргумент retryOnRateLimit блокирует запуски на заданное время по достижении лимита, установленного разработчиками API. Заметим, что количество возвращаемых твитов может быть меньше заданного n значения также из-за ограничений API [8].

3.4 Фильтрация результатов выборки данных

Составим строку поиска из запросов, разделённых знаками “+”:

search\_string <- paste0(c("XXXXX","#XXXXX"),collapse = "+")

Если поиск ведётся на русском языке, это желательно указывать в параметре lang

tweets <- searchTwitter (search\_string, n=15000, lang="ru")

иначе есть риск получить в результатах постороннюю “нелатиницу”.

Объект tweets относится к классу status и имеет множество методов get\*(), возвращающих ту или иную часть твита, так функция getText() возвращает текст

text <- sapply(tweets, function(x) x$getText()).

Отфильтруем из полученных твитов те, что содержат геоданные о расположении их авторов (непустые значения долготы или широты!) и сохраним тексты твитов, имена и расположение авторов, а также время создания твита в таблице df:

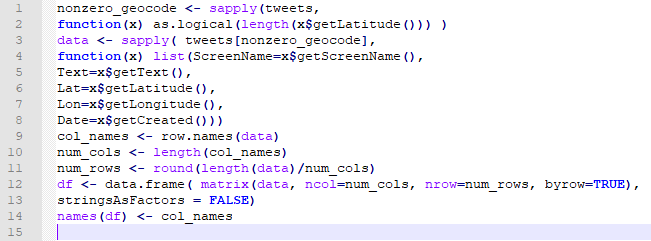


Рисунок 3 – Фильтрация геоданных

4 Визуализация геокодированных данных

Визуализация данных — это область информатики, связанная с визуальным представлением данных. Визуальное представление – мощный инструмент, позволяющий исследовать сложные данные, и эффективный способ представления и передачи результатов анализа в целом. Посредством визуализации люди получают возможность увидеть аспекты, которые не заметны с первого взгляда.

В конце концов, изображение стоит тысячи слов - удачная визуализация позволяет читателю впитывать сложные понятия при помощи простого изображения.

Например, визуализация данных может использоваться исследователями-аналитиками во время первичного анализа данных, чтобы понять их суть. Кроме того, исследователи-аналитики могут также использовать визуализацию для общения с неспециалистами и объяснения, что есть интересного в данных.

Язык программирования R предлагает большое количество инструментов и библиотек для визуализации данных, средствами которых представляется возможным выводить изображения типографского качества в многочисленных форматах и видах [9].

# 4.1 Временные ряды

Распределение твитов во времени один из важнейших и интересных аспектов исследования.

Временной ряд — это последовательность точек данных, которая состоит из наблюдений за тот или иной интервал времени. Поскольку твиты имеют поле created\_at с точным временем создания, мы можем распределить твиты по временным дискретным интервалам, чтобы выяснить, как пользователи реагируют на те или иные события во времени. Для такого анализа лучше всего подходят данные, собираемые через потоковый программный интерфейс Streaming API.

Преобразуем временные метки в соответствующий формат, округлим Latitude и Longitude (широту и долготу) — отобразим активность твит-авторов по сетке в одну десятую градуса. Последнее преобразование долготы необходимо для дальнейшей визуализации.

Активируем библиотеку ggplot2 и можем вывести распределение по времени:

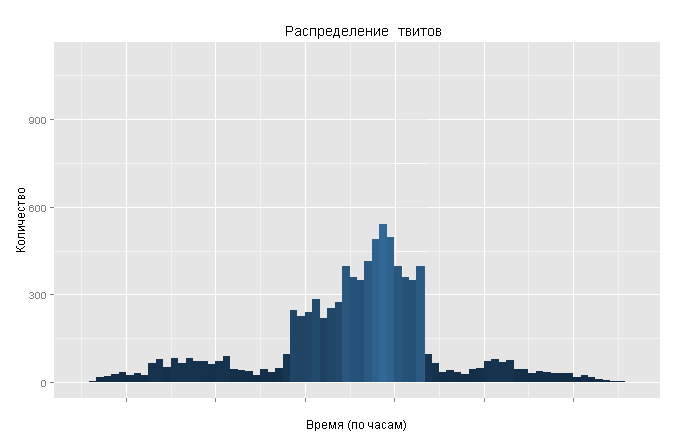


Рисунок 3 — Распределение твитов

Этот пример наглядно демонстрирует, какова реакция на те или иные события. Таким образом данный анализ временных рядов крайне необходим многим компаниям. Посредством такого мониторинга мнений пользователей управленческие решения будут более оптимизированными и взвешенными во всех отношениях. А динамическая природа социальных сетей идеально подходит для выяснения реакции на любые события, новости, явления.

4.2 Создание макета карты

В экосистеме языка R для визуализации геоданных реализованы широчашие возможности по взаимодействию со всевозможными картографическими агрегаторами и огромным количеством картографических форматов карт.

На рисунке 4 приведен пример реализации анализа геоданных с применением картографического формата карт Geography Markup Language (GML):

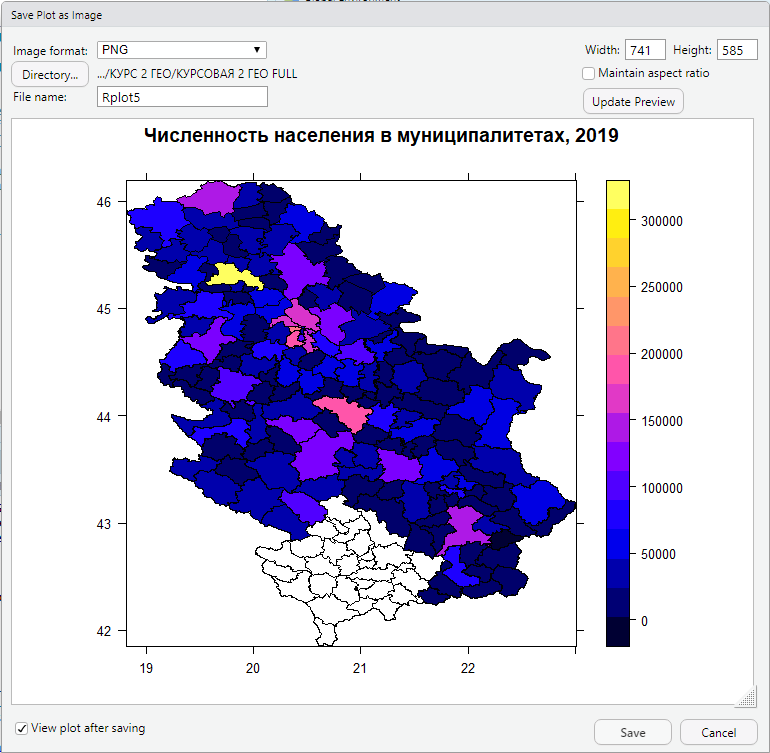


Рисунок 4 – Картографический формат Geography Markup Language (GML)

Для создания интерактивных веб-карт широко используется JavaScript-библиотека Leaflet. Пакет leaflet является интерфейсом к этой библиотеке.

Использование пакета начинается с того, что создается виджет (элемент графического интерфейса) "карта" с помощью функции leaflet(). Затем на карту добавляются слои данных, такие как листы карты (с помощью addTiles()) или маркеры объектов (addMarkers()).

leaflet поддерживает оператор последовательного выполнения функций %>% из пакета magrittr:

m <- leaflet() %>%

addTiles() %>% # добавим листы карт OpenStreetMap

addMarkers(lng=45.768, lat=38.852,

popup="KUBSU")

m # Вывод карты на экран на рисунке 5:

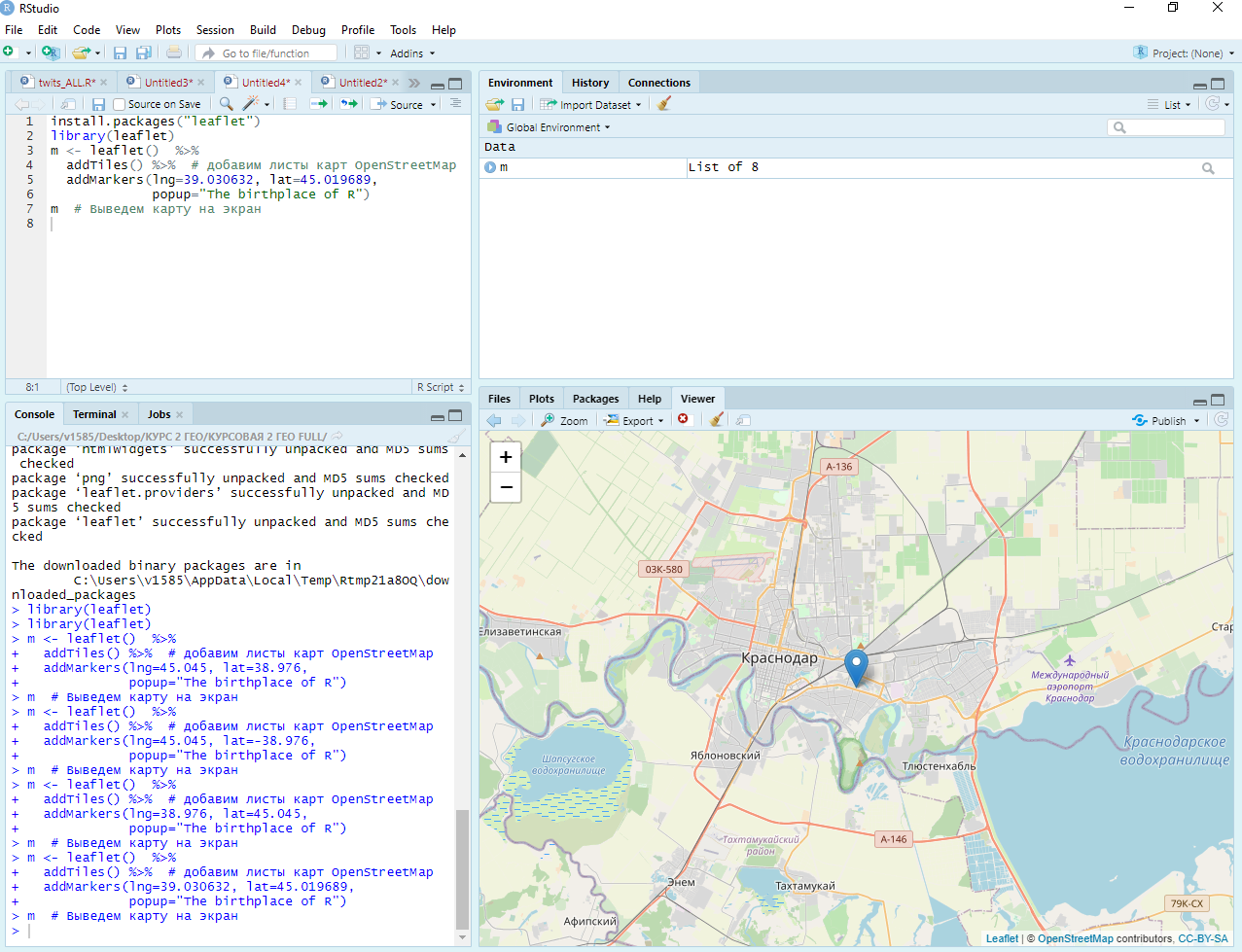


Рисунок 5 – Интерактивная Web карта из пакета Leaflet

По умолчанию функция addTiles() использует карты OpenStreetMap. Выбор карт осуществляется функцией addProviderTiles(). Если, например, нужно использовать карты Esri.WorldTopoMap, то сделать это можно так:

m %>% addProviderTiles("Esri.WorldTopoMap")

Библиотека ggplot2 также предлагает внушительный набор инструментов для быстрого построения информативных графиков и картографических основ.

Функцию geom\_polygon используем для отображения карты, она позволяет отрисовывать полигоны по заданным координатам. Контуры стран возмем из стандартной библиотеки maps.

Исходные данные — таблица сопряженности с координатами и временем публикации твита. Разворачиваем таблицу сопряженности в dataframe из 3 колонок: широта, долгота, частота (количество твитов в данной точке). Удаляем строки, в которых частота равна нулю или координаты выходят за установленные границы.

frame.twits <- melt(table(frame.twits$Latitude, frame.twits$Longitude))

colnames(frame.twits) <- c('Lat', 'Long', 'Volume')

frame.twits$Lat <- as.numeric(as.character(frame.twits$Lat))

frame.twits$Long <- as.numeric(as.character(frame.twits$Long))

Преобразование широты и долготы в числовые переменные необходимо потому, что после «сворачивания-разворачивания» данных они преобразуются в категориальные (factor в терминологии R).

На выходе получаем карту с нанесенными твитами на рисунке 6:



Рисунок 6 – Карта с нанесенными твитами.

# ЗАКЛЮЧЕНИЕ

Геоданные — важнейшая часть данных цифровых информационных систем являясь частным свойством геопространства.  Без геопространственной привязки данные не имеют существенной ценности. Однако геоданные в непосредственном их виде (как массив координат точек в геопространстве) сложны для восприятия их человеком, в связи с чем в пользовательском интерфейсе необходимо использовать различные методы визуализации геоданных — их визуального наложения на изображение геопространства.

Наиболее распространенный способ визуализации геоданных — использование веб - интерфейса открытых картографических сервисов (Google Maps, Яндекс.Карты и др.). Однако данный способ имеет целый ряд недостатков. Прежде всего, работа внешнего веб-интерфейса зависит от сторонних сервисов и их ограничений. И что более важно визуализация геоданных связана с передачей их третьим лицам (стороннему сервису), что также не всегда приемлемо. В связи с этим в данной курсовой работе были реализованы модули для взаимодействия с программным интерфейсом Streaming API, извлечения, сбора, подготовки и визуализации геоданных на стороне клиента обеспечивая тем самым конфиденциальность исследований.

Анализ данных социальных сетей – инструмент, мощность которого трудно переоценить. На примере геометрик в работе было продемонстрировано, насколько такой анализ может быть полезен для принятия важных решений и на сколько способен предоставить достаточно полную и подробную картину взаимодействий и реакций сети на те или иные события.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Вассерман С., Фауст K. Анализ социальных сетей. Методы и приложения. New York: Cambridge University Press, 1994.
2. Шипунов А. Б. и др. Наглядная статистика. Используем R!. — 2014. — 296 с.
3. Зарядов И. С. Введение в статистический пакет R: типы переменных, структуры данных, чтение и запись информации, графика. – М.: Изд-во РУДН, 2010. – 207 с.
4. Мастицкий С. Э., Шитиков В. К. Статистический анализ и визуализация данных с помощью R. — М.: ДМК Пресс 2014. — 401 с.
5. Мастицкий С. Э. Визуализация данных с помощью ggplot2. - М.: ДМК Пресс, 2016.-222 с.
6. Храмов Д. А. Сбор данных в Интернете на языке R. - М.: ДМК Пресс, 2017.- 283 с.
7. Кабаков Р. И. R в действии. Анализ и визуализация данных в программе R / пер. с англ. Полины А. Волковой. – М.: ДМК Пресс, 2014. – 588 с.
8. Язык R в задачах науки о данных. Импорт, подготовка, обработка, визуализация и моделирование данных | У. Хэдли,Г. Гарретт, 2017 – 592с.
9. Вицентий А.В. Адаптивная визуализация геоданных в соц. медиа // Интернет-журнал «НАУКОВЕДЕНИЕ» Том 8, №4 (2016) http://naukovedenie.ru Яз. рус., англ.
10. Искусство программирования на R. Погружение в большие данные. – М.: Изд-во РУДН, 2019. – 207 с.

Приложение А

Модули R - скриптов автоматической авторизации, загрузка и визуализации геоданных твитов:

1. # Авторизация в Twitter
2. library(twitteR)
3. api\_key <- "API KEY"
4. api\_secret <- "API SECRET"
5. access\_token <- "ACCESS TOKEN"
6. access\_token\_secret <- "ACCESS TOKEN SECRET"
7. setup\_twitter\_oauth (api\_key, api\_secret, access\_token, access\_token\_secret)
9. # Подключение Библиотек
10. **library**('ggplot2')
11. **library**('maps')
12. **library**('reshape2')
13. **library**('RColorBrewer')
15. # Конфигурация данных исследуемого диапазона времени
16. **setwd**('C/Rtemp/geo/')
17. start.date <- '2020-05-01 00:00:00'
18. finish.date <- '2020-05-01 24:00:00'
19. seconds.in.frame <- 30
20. ma.period <- 60 \* 30 # Период для движущихся твитов в кадре
21. texttime <- 3 \* 24 # Базовое время для демонстрации в кадре
23. # Формат данных старта и финиша
24. start.date <- **strptime**(start.date, **format**='%Y-%m-%d %H:%M:%S')
25. finish.date <- **strptime**(finish.date, **format**='%Y-%m-%d %H:%M:%S')
26. frames <- **as.numeric**(**difftime**(finish.date, start.date, **units**='secs'))/seconds.in.frame
28. # Палитра для ggplot и настрока геометрии
29. twits.colors=brewer.pal(9, 'YlOrRd')
30. roundcoef <- .48
31. textcolors <- **c**('coral5')
33. # Загружаем твиты с координатами и конвертируем значения в нужный формат
34. twits <- **read.csv2**('ny\_tweets.csv', header=**F**)
35. **colnames**(twits) <- **c**('Link', 'Longitude', 'Latitude', 'Timestamp')
36. twits$Timestamp <- **strptime**(twits$Timestamp, **format**='%Y-%m-%d %H:%M:%S')
37. twits$Latitude <- **round**(**as.numeric**(**as.character**(twits$Latitude)), digits=1)
38. twits$Longitude <- **round**(**as.numeric**(**as.character**(twits$Longitude)), digits=1)
39. twits$Longitude <- **sapply**(twits$Longitude, **function**(x){
40. **if**(x < (-169)){
41. x<-360+x
42. }
43. **else**{x}
44. })
46. # Загружаем твиты с текстами и конвертируем дату и время
47. twit.texts <- **read.csv**('tweets.csv', header=**F**)
48. **colnames**(twit.texts) <- **c**('Link', 'Latitude', 'Longitude', 'Timestamp', 'Text')
49. twit.texts$Timestamp <- **strptime**(twit.texts$Timestamp, **format**='%Y-%m-%d %H:%M:%S')
50. twit.texts <- twit.texts[, **c**(4,5)]
51. twit.texts$t.delta <- **rnorm**(**nrow**(twit.texts), **mean** = 1, **sd** = .1) \* texttime
52. twit.texts$t.start <- twit.texts$Timestamp - twit.texts$t.delta \* seconds.in.frame / 2
53. twit.texts$t.end <- twit.texts$Timestamp + twit.texts$t.delta \* seconds.in.frame / 2
54. twit.texts$color <- textcolors[**round**(**runif**(**nrow**(twit.texts), 1, **length**(textcolors)))]
55. twit.texts$x <- **rnorm**(**nrow**(twit.texts), **mean** = 100, **sd** = 30)
56. twit.texts$y <- **rnorm**(**nrow**(twit.texts), **mean** = 56, **sd** = 15)
57. twit.texts$size <- **rnorm**(**nrow**(twit.texts), **mean** = 10, **sd** = 2)
58. twit.texts$opacity <- 0
60. # Страны, чтобы показать на карте
61. countries<**c**('USSR','Albania','Andorra','Austria','Belgium','Bulgaria','Denmark',
62. 'Czechoslovakia','Finland','France','Germany','Hungary','Ireland',
63. 'Italy','Liechtenstein','Luxembourg','Malta','Monaco',
64. 'Netherlands','Norway','Poland','Portugal','Romania','San Marino',
65. 'Sicily','Spain','Sweden','Switzerland','UK','Wales','Yugoslavia')
67. # Загружаем города (столицы) для отметки на карте
68. cities <- **read.csv2**('cities.csv', header=**F**)
69. **colnames**(cities) <- **c**('name', 'Lat', 'Long')
70. cities$Lat <- **as.numeric**(**as.character**(cities$Lat))
71. cities$Long <- **as.numeric**(**as.character**(cities$Long))
72. cities$Long <- **sapply**(cities$Long, **function**(x){
73. **if**(x < (-169)){
74. x<-360+x
75. }
76. **else**{x}
77. })
79. # Загружаем данные базовой карты мира ограничив координатами (долготой и широтой)
80. full\_map <- map\_data('world')
81. **table**(full\_map$region)
82. need.map <- **subset**(full\_map, region %in% countries & long>-25 & long<190 & lat>25)
84. #   Строим гистограмму
86. p <- ggplot()
87. p <- p + geom\_histogram(aes(x=twits$Timestamp, fill = ..count..), binwidth = 3600)
88. p <- p + ylab('Количество') + xlab('Время (по часам)')
89. p <- p + theme(legend.position = 'none')
90. p <- p + ggtitle(**expression**('Частоты твитов'))
91. p
93. max.color <- 0
94. **for**(i **in** 1:frames){
95. # выбор твитов
96. frame.time <- start.date + i\*seconds.in.frame
97. frame.twits <- **subset**(twits,Timestamp <= frame.time & Timestamp > frame.time - ma.period)
99. # создаем таблицу со значениями
100. frame.twits <- melt(**table**(frame.twits$Latitude, frame.twits$Longitude))
101. **colnames**(frame.twits) <- **c**('Lat', 'Long', 'Volume')
102. frame.twits$Lat <- **as.numeric**(**as.character**(frame.twits$Lat))
103. frame.twits$Long <- **as.numeric**(**as.character**(frame.twits$Long))
104. frame.twits <- frame.twits[frame.twits$Volume>0 &
105. frame.twits$Long>=-25 & frame.twits$Long<=190 &
106. frame.twits$Lat>=25 & frame.twits$Lat<=85,]
108. **if**(max.color < **max**(**log**(frame.twits$Volume))){
109. max.color <- **max**(**log**(frame.twits$Volume))
110. }
111. }
113. **for**(i **in** 1:frames){
115. # выбираем только нужные твиты
116. frame.time <- start.date + i\*seconds.in.frame
117. frame.twits <- **subset**(twits,Timestamp <= frame.time & Timestamp > frame.time - ma.period)
119. # Создаем ьаблицу со значениями
120. frame.twits <- melt(**table**(frame.twits$Latitude, frame.twits$Longitude))
122. # Отфильтруем пустые ячейки
123. **colnames**(frame.twits) <- **c**('Lat', 'Long', 'Volume')
124. frame.twits$Lat <- **as.numeric**(**as.character**(frame.twits$Lat))
125. frame.twits$Long <- **as.numeric**(**as.character**(frame.twits$Long))
126. frame.twits <- frame.twits[frame.twits$Volume>0 &
127. frame.twits$Long>=-25 & frame.twits$Long<=190 &
128. frame.twits$Lat>=25 & frame.twits$Lat<=85,]
130. frame.colors <- **round**(1 + (8\***log**(frame.twits$Volume)/max.color), digits=0)

133. # Визуализация геоданных
134. # Отрисовка контура карты
135. p <- ggplot()
136. p <- p + geom\_polygon(aes(x=need.map$long, y=need.map$lat, group = need.map$group),
137. colour='white', fill='grey20', alpha=.5)
139. # Нанесение на карту Столиц
140. p <- p + geom\_point(aes(cities$Long, cities$Lat), colour='skyblue', size=1.5)
142. # Нанесение на карту твитов
143. **if**(**nrow**(frame.twits)>0){
144. p <- p + geom\_point(aes(frame.twits$Long,frame.twits$Lat,
145. size=frame.twits$Volume \* 5),
146. colour=twits.colors[frame.colors], alpha = .75)
147. }
149. # Отображение твитов на карте
150. p<-p+geom\_text(aes(x=twit.texts$x,y=twit.texts$y,label=**iconv**(twit.texts$Text,
151. to='UTF-8')),
152. colour=twit.texts$color, size=twit.texts$size, alpha = twit.texts$opacity)
154. # Отрисовка осей, легенды, заголовка
155. p <- p + theme(axis.line=element\_blank(),axis.text.x=element\_blank(),
156. axis.text.y=element\_blank(),axis.ticks=element\_blank(),
157. axis.title.x=element\_blank(),
158. axis.title.y=element\_blank(),
159. legend.position = 'none',
160. **text**=element\_text(**family**='mono', size=20, face='bold', colour='dodgerblue3')
161. )
162. p <- p + scale\_x\_continuous(limits = **c**(-15, 190))
163. p <- p + scale\_y\_continuous(limits = **c**(30, 82))
164. p <- p + ggtitle(**expression**('# МИР ТРУД МАЙ'))
166. # Отрисовка осей легенды заголовка
167. f.name <- **as.character**(i)
168. **repeat**{
169. **if**(**nchar**(f.name) < **nchar**(**as.character**(frames))){
170. f.name <- **paste**('0', f.name, sep='')
171. } **else** {
172. **break**
173. }
174. }
175. }