بنام خدا



پروژه پایانی

داده کاوی

**F**requency **T**ags **C**lustering

استاد آقای دکترنشاطی

آموزشیار آقای رستمی

**بهزاد خسروی فر**

۹۶۴۴۳۱۱۱

فهرست

1. [انتخاب زبان در google sheet 3](#_Toc518909135)
2. [۲۰۰ تا تگ پر رخداد در زبان انتخابی R 4](#_Toc518909136)

# انتخاب زبان در [google sheet](https://docs.google.com/spreadsheets/d/1MeZpZMImfy8COn0iJuPJgiNXy7az9skoA2ILdwfJQDw/edit?usp=sharing)

A) JavaScript

B) Java

C) C#

D) Php

E) Android

F) Python

G) IOS

F) C++

H) **R**

از زبان های فوق زبان R را انتخاب کردم.

# استخراج ۲۰۰ تا تگ پر رخداد در زبان انتخابی R

برای این کار از زبان JavaScript در پلتفرم Node.js استفاده شد تا بتوان بسرعت فایل حجیم ۶۰ گیگابایتی را بصورت استریم خوانده و پست های دارای تگ <R> را استخراج نمود.

کد مربوطه بصورت زیر در فایلی بنام “r-tags-histogram.js” ذخیره شده است:

var bigXml = require('big-xml-streamer');

var fs = require("fs");

var clear = require('clear');

var rTags = {};

var reader = bigXml.createReader('../Posts.xml', /^(row)$/, { gzip: false });

fs.writeFileSync(`${\_\_dirname}/results/r-tags.csv`, ""); // create or clear r-tags.csv file

reader.on('record', function (record) {

// if post is "question" then that have tags attribute

if (record.attrs.PostTypeId == "1") {

var tags = record.attrs.Tags.toLowerCase();

if (tags && tags.indexOf(`<r>`) > -1) { // tags are about "r"

// convert to tags array

var arrTags = tags.replace(/</g, "").split(">").filter(v => v != "" && v != "r");

if (arrTags.length > 0) {

var correlatedTags = arrTags.join(",");

console.warn(correlatedTags);

// write any post tags which have 'R' tag

fs.appendFileSync(`${\_\_dirname}/results/r-tags.csv`, correlatedTags + "\n")

arrTags.forEach(tag => {

var tagCount = rTags[tag];

rTags[tag] = tagCount ? tagCount + 1 : 1;

});

fs.writeFileSync(`${\_\_dirname}/results/r-tags-histogram.json`, JSON.stringify(rTags));

}

}

}

});

در کد فوق فقط هرجایی که <row> با attribute ای بنام PostTypeId و با مقدار ۱ بود خوانده می شد، زیرا این نوع از پست‌ها مربوط به سوالات است و تگ‌ها فقط بر روی پست سوالات زده می‌شوند نه بر روی پست جواب‌ها.

در قسمت دوم تمام تگ‌ها به حروف کوچک تبدیل شدند تا به حداکثر تشابه تگ‌های نسبت به هم برسیم، به عبارت دیگر، تگ R برابر تگ r قرار بگیرد.

در قسمت سوم فقط تگ‌های پست‌هایی ثبت می‌شود که آن پست حتما دارای تگ زبان انتخابی یعنی R باشد.

در نهایت خروجی دو فایل زیر است:

r-tags.csv تگ‌های تمام پست‌های دارای تگ <R>

r-tags-histogram.json هستگرام یا فراوانی تگ‌های بدست آمده در ارتباط با تگ <R>

در قسمت بعدی با یک الگوریتم ساده‌ و با استفاده از Node.js تمام تگ‌ها را بصورت نزولی مرتب کرده و ۲۰۰ تای اول را در فایل “top-freq-tags.json” ذخیره کردیم که بصورت زیر همراه با تعداد فراوانی‌شان ثبت شده اند:

{

"ggplot2": 20782,

"dataframe": 14244,

"plot": 9837,

"shiny": 9754,

"dplyr": 8164,

"data.table": 6871,

"matrix": 5300,

"loops": 3858,

"regex": 3770,

"function": 3725,

"for-loop": 3568,

"rstudio": 3550,

"list": 3514,

"time-series": 3225,

"statistics": 3120,

"knitr": 3013,

"csv": 2838,

"subset": 2799,

"r-markdown": 2743,

"python": 2594,

.

.

.

}

برای نمایش گرافیکی یا Visualization کردن تگ‌های بدست آمده از نمودار word cloud[[1]](#footnote-1) استفاده کردم که نتیجه بصورت شکل ۱ در آمد.



**شکل ۲**

در شکل فوق کلمات یا تگ‌های دارای فرآوانی بیشتر بصورت درشت‌تر و بزرگ‌تر نمایش داده می‌شوند. لذا، دو کلمه “ggplot2” , “dataframe” دارای بالاترین میزان فرآوانی در بین تگ‌های زبان R را دارند.

# ۳. خوشه بندی تگ‌های مرتبط بهم و ایجاد حوزه‌ها‌ی تخصصی در زبان انتخابی

ابتدا باید یک الگوریتم خوشه بندی مناسب این کار، انتخاب می‌شد. الگوریتم‌های موجود در کتاب که در زیر عنوان شده‌اند:

* روش پارتیشن بندی Partitioning approach   
  k-means , medoids, CLARANS
* روش سلسله مراتبی Hierarchical approach   
  Diana, Agnes, BIRCH, CAMELEON
* روش مبتنی بر چگالی Density based approach   
  DBSACN, OPTICS, DenCLue
* روش مبتنی بر گرید Grid based approach   
  STING, WaveCluster, CLIQUE

الگوریتم انتخابی‌ برای دسته بندی تگ‌ها، الگوریتم Hierarchical Agglomerative Clustering یا همان سلسله مراتبی است که دو به دو تگ‌های بهم نزدیک را پیدا کرده و خوشه‌ها را پیدا می‌کند.

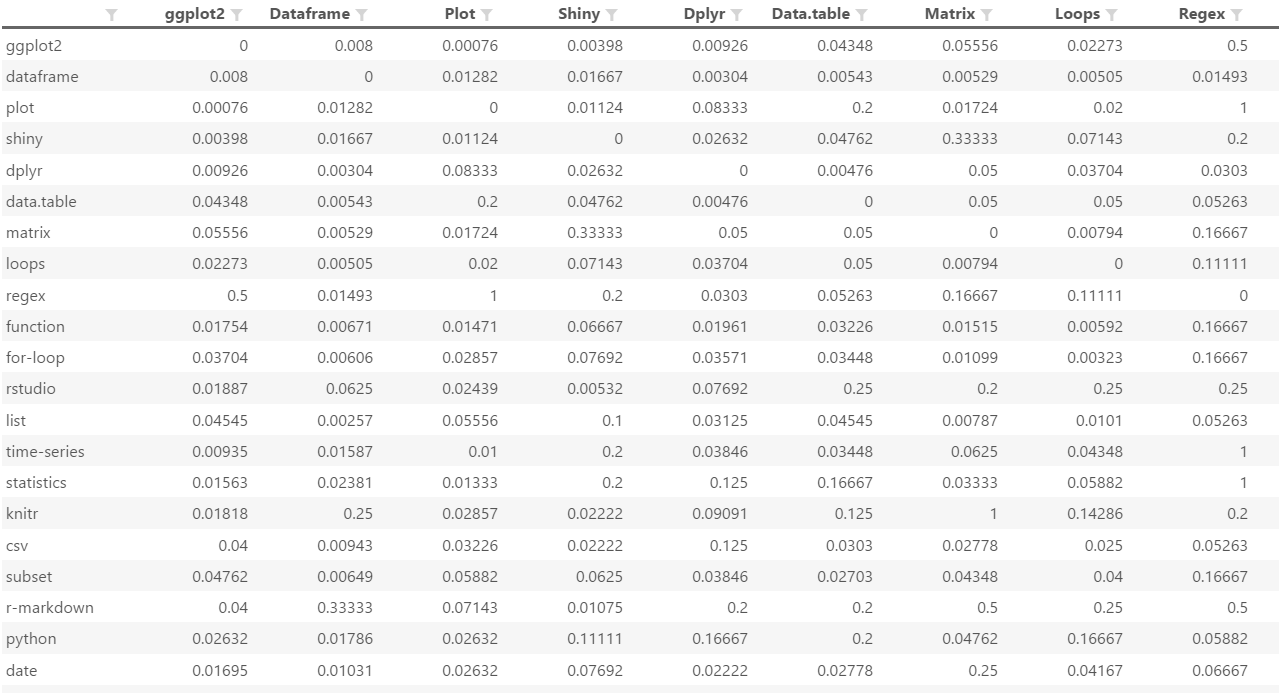
در الگوریتم سلسله مراتبی ما نیاز به فاصله بین تگ‌ها داریم یعنی در واقع به یک ماتریس ۲۰۰×۲۰۰ نیاز داریم که سطرها و ستون‌ها در آن ۲۰۰ تگ بدست آمده است. هر درآیه آن نشان گر فاصله تگ ستونی از تگ سطری است. در مثال شکل ۲ این ماتریس نمایش داده شده است. در این ماتریس، قطر فرعی فاصله‌ی تگ از خودش است که باید ۰ صفر باشد.

در اینجا فاصله‌ی بین دو تگ را میزان Support آن دو در نظر گرفته‌ایم. بعبارت دیگر میزان دفعاتی که این تگ‌ها با هم تکرار شده‌اند. پس ابتدا باید این ماتریس را پردازش کرده و در حافظه نگه داریم تا در دفعات متعدد از آن استفاده کنیم.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Dist.** | **r** | **tag** | **file** | **plot** | **reg** |
| **r** | 0 | 184 | 222 | 177 | 216 |
| **tag** | 184 | 0 | 45 | 123 | 128 |
| **file** | 222 | 45 | 0 | 129 | 121 |
| **plot** | 177 | 123 | 129 | 0 | 46 |
| **reg** | 216 | 128 | 121 | 46 | 0 |

**شکل ۲**

با زبان JavaScript یک فایل csv از ماتریس را می‌سازیم تا فقط در R آن را هر بار بخوانیم. این فایل حاوی اطلاعات ماتریس فاصله، در فایل “distance-matrix.csv” بصورت شکل ۳ ذخیره شده است.

 **شکل ۳**

همانطور که در شکل ۳ مشاهده می‌کنید، اعداد فاصله بین دو عدد ۰ و ۱ می‌باشد، که در آن ۰ فاصله‌ی تگی از خودش است و عدد ۱ نشان دهنده‌ی دورترین فاصله است که برای تگ‌هایی استفاده می‌شود که هیچ وقت باهم تکرار نشده‌اند.

بعد از اینکه ماتریس فاصله بدست آمد به سراغ زبان R برای محاسبه ی خوشه‌بندی و همچنین رسم نمودار سلسله مراتبی، می‌رویم.

با دستور زیر در محیط RStudio و زبان R می‌توان فایل .csv ماتریس فاصله را بارگذاری کرد:

distanceMat <- read.delim("../results/distance-matrix.csv", header=TRUE, sep=",")

حال نوبت محاسبه ماتریس پایین مثلثی فاصله و خوشه‌بندی مدل سلسله مراتبی است:

d <- dist(as.matrix(distanceMat)) # find distance matrix

hc <- hclust(d, method = "complete") # hierarchical clustering by complete linkage method

الگوریتم خوشه بندی ما با روش Complete linkage انجام می‌شود.

1. [ابر برچسب (به انگلیسی: Tag Cloud) مجموعه‌ای از برچسب‌های وزن‌دار است](https://timdream.org/wordcloud/#wikipedia:Cloud) [↑](#footnote-ref-1)