بنام خدا

پروژه پایانی

داده کاوی

Frequency **T**ags **C**lustering

استاد آقای دکتر نشاطی

آموزشیار آقای رستمی

بهزاد خسروی فر

۹۶۴۴۳۱۱۱

فهرست

۱. انتخاب زبان در google sheet

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.

۲. استخراج ۲۰۰ تا تگ پر رخداد در زبان انتخابی R

برای این کار از زبان JavaScript در پلتفرم Node.js ا ستفاده شد تا بتوان با سرعت 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 تگ های بدســـت آمده از نمودار cloud word[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 "dataframe" , "ggplot2" دارای بالاترین میزان فرآوانی در بین تگ‌های زبان R را دارند.

[1] مجموعه‌ای از برچسب‌های وزن‌دار است (به انگلیسی: Tag Cloud) ابر برچسب

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

ابتدا باید یک الگوریتم خو شه بندی منا سب این کار، انتخاب می شد. 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 سل سله مراتبی ما نیاز به فا صله 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.

الگوریتم خوشه بندی سلسله مراتبی ما به روش Complete linkage انجام می‌شود. در این روش فا صله‌ی بین دو خو شه برابر، بی‌شترین فا صله‌ی بین خو شه‌ها یا تگ‌های داخل آن خو شه با دیگر خو شه می‌باشد. به عبارت ریاضی، فاصله دو خوشه A و B برابر است با: $D(A, B) = \max_{x \in A,\ y \in B} d(x, y)$
که در آن A, B دو خوشه مجزا از هم بوده و d(x,y) فاصله بین اعضای خوشه‌های A با B می‌باشد.

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 می سازیم 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.

	ggplot2	Dataframe	Plot	Shiny	Dplyr	Data.table	Matrix	Loops	Regex
ggplot2	0	0.008	0.00076	0.00398	0.00926	0.04348	0.05556	0.02273	0.5
dataframe	0.008	0	0.01282	0.01667	0.00304	0.00543	0.00529	0.00505	0.01493
plot	0.00076	0.01282	0	0.01124	0.08333	0.2	0.01724	0.02	1
shiny	0.00398	0.01667	0.01124	0	0.02632	0.04762	0.33333	0.07143	0.2
dplyr	0.00926	0.00304	0.08333	0.02632	0	0.00476	0.05	0.03704	0.0303
data.table	0.04348	0.00543	0.2	0.04762	0.00476	0	0.05	0.05	0.05263
matrix	0.05556	0.00529	0.01724	0.33333	0.05	0.05	0	0.00794	0.16667
loops	0.02273	0.00505	0.02	0.07143	0.03704	0.05	0.00794	0	0.11111
regex	0.5	0.01493	1	0.2	0.0303	0.05263	0.16667	0.11111	0
function	0.01754	0.00671	0.01471	0.06667	0.01961	0.03226	0.01515	0.00592	0.16667
for-loop	0.03704	0.00606	0.02857	0.07692	0.03571	0.03448	0.01099	0.00323	0.16667
rstudio	0.01887	0.0625	0.02439	0.00532	0.07692	0.25	0.2	0.25	0.25
list	0.04545	0.00257	0.05556	0.1	0.03125	0.04545	0.00787	0.0101	0.05263
time-series	0.00935	0.01587	0.01	0.2	0.03846	0.03448	0.0625	0.04348	1
statistics	0.01563	0.02381	0.01333	0.2	0.125	0.16667	0.03333	0.05882	1
knitr	0.01818	0.25	0.02857	0.02222	0.09091	0.125	1	0.14286	0.2
csv	0.04	0.00943	0.03226	0.02222	0.125	0.0303	0.02778	0.025	0.05263
subset	0.04762	0.00649	0.05882	0.0625	0.03846	0.02703	0.04348	0.04	0.16667
r-markdown	0.04	0.33333	0.07143	0.01075	0.2	0.2	0.5	0.25	0.5
python	0.02632	0.01786	0.02632	0.11111	0.16667	0.2	0.04762	0.16667	0.05882
date	0.01695	0.01031	0.02632	0.07692	0.02222	0.02778	0.25	0.04167	0.06667

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 می شود 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=",")
```

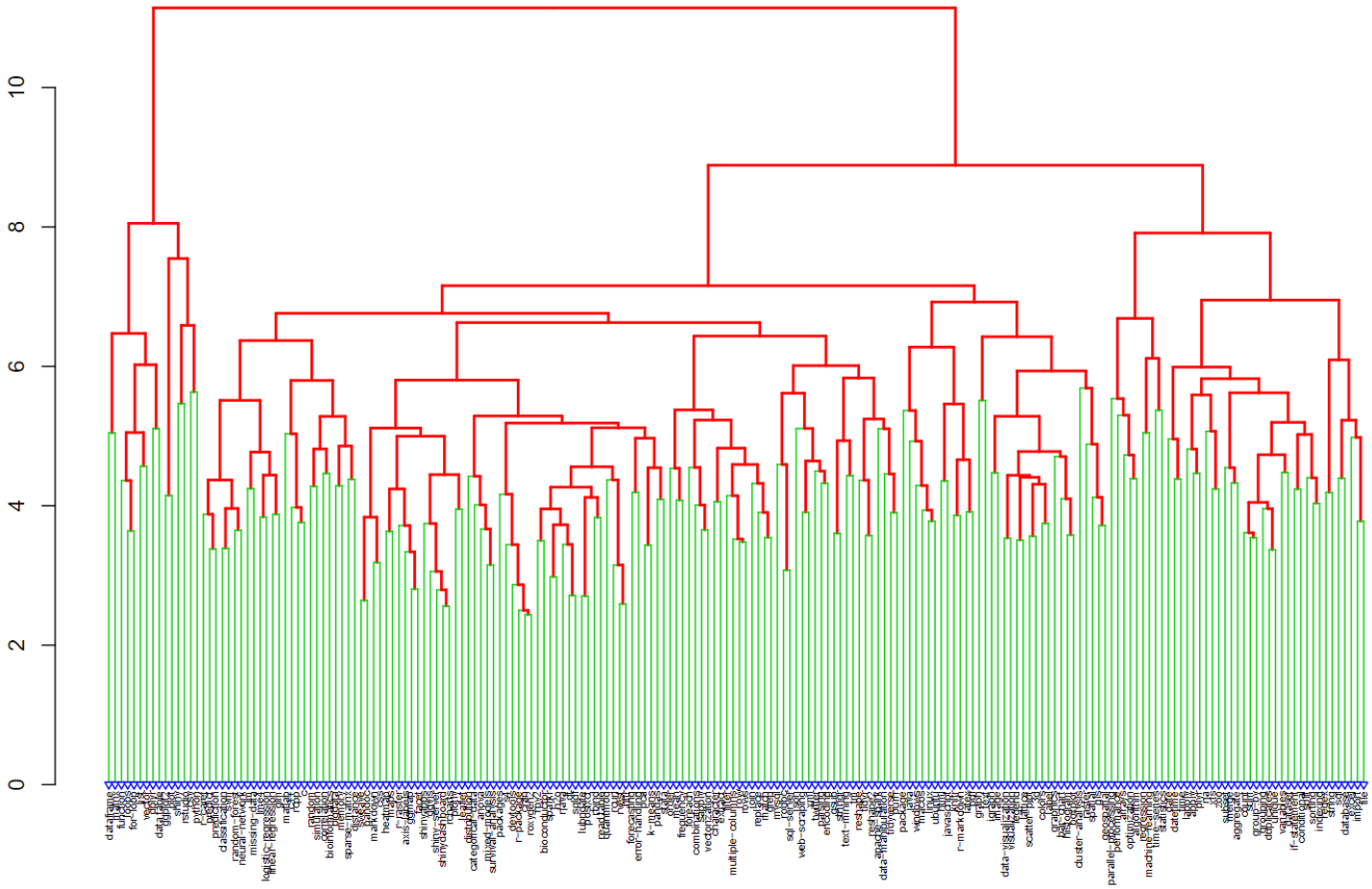
حال نوبت محاسبه ماتریس پایین مثلثی فاصله و خوشه‌بندی مدل سلسله مراتبی است:

```
d <- dist(as.matrix(distanceMat))      # find distance matrix
hc <- hclust(d, method = "complete")    # hierarchical clustering by complete linkage method
```

و در نهایت برای رسم نمودار از تابع زیر استفاده شده:

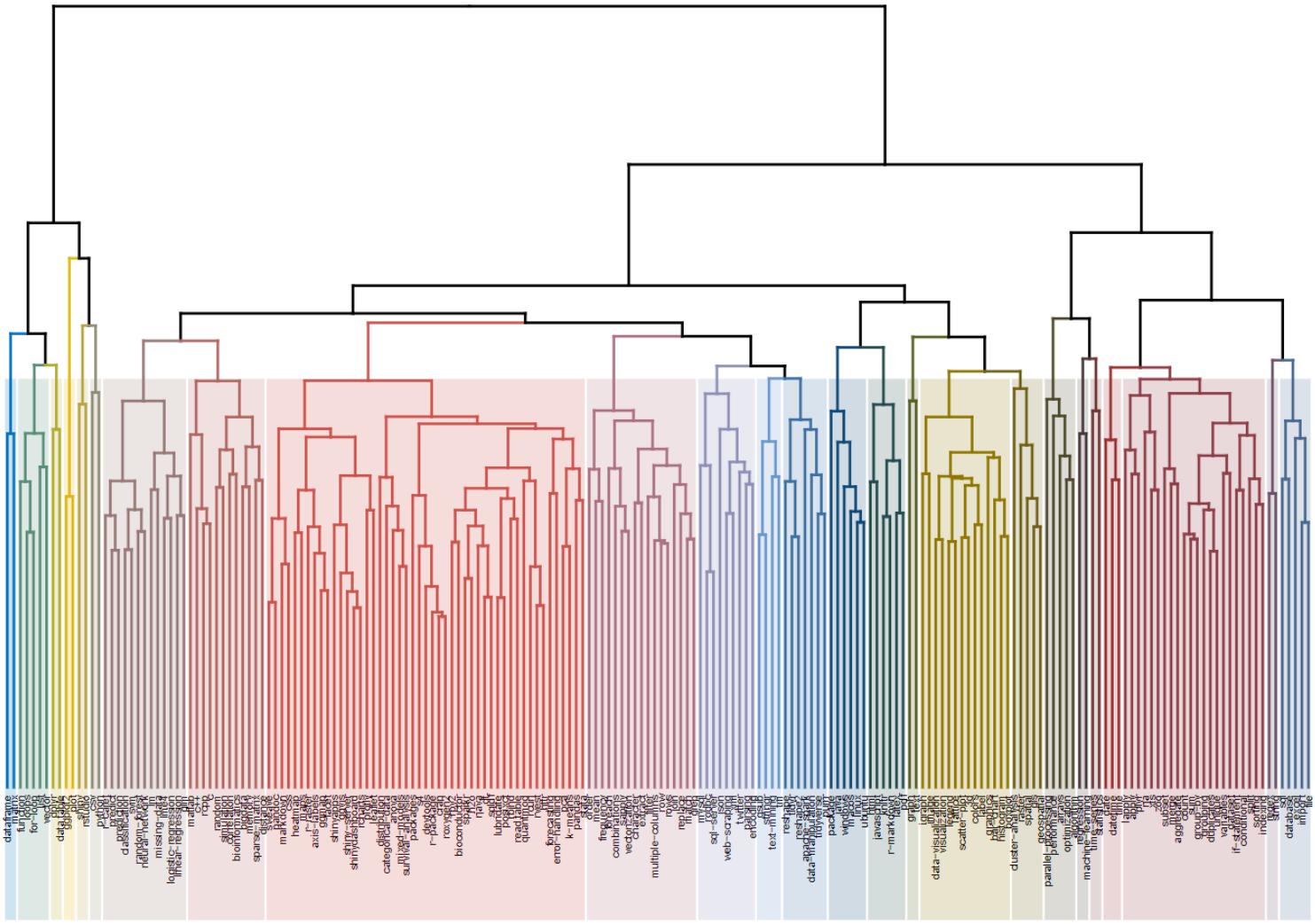
```
plot(hc, …)
```

نمودارهای حاصل از نتایج کدهای R به شکل زیر است: (همگی در پوشه results)



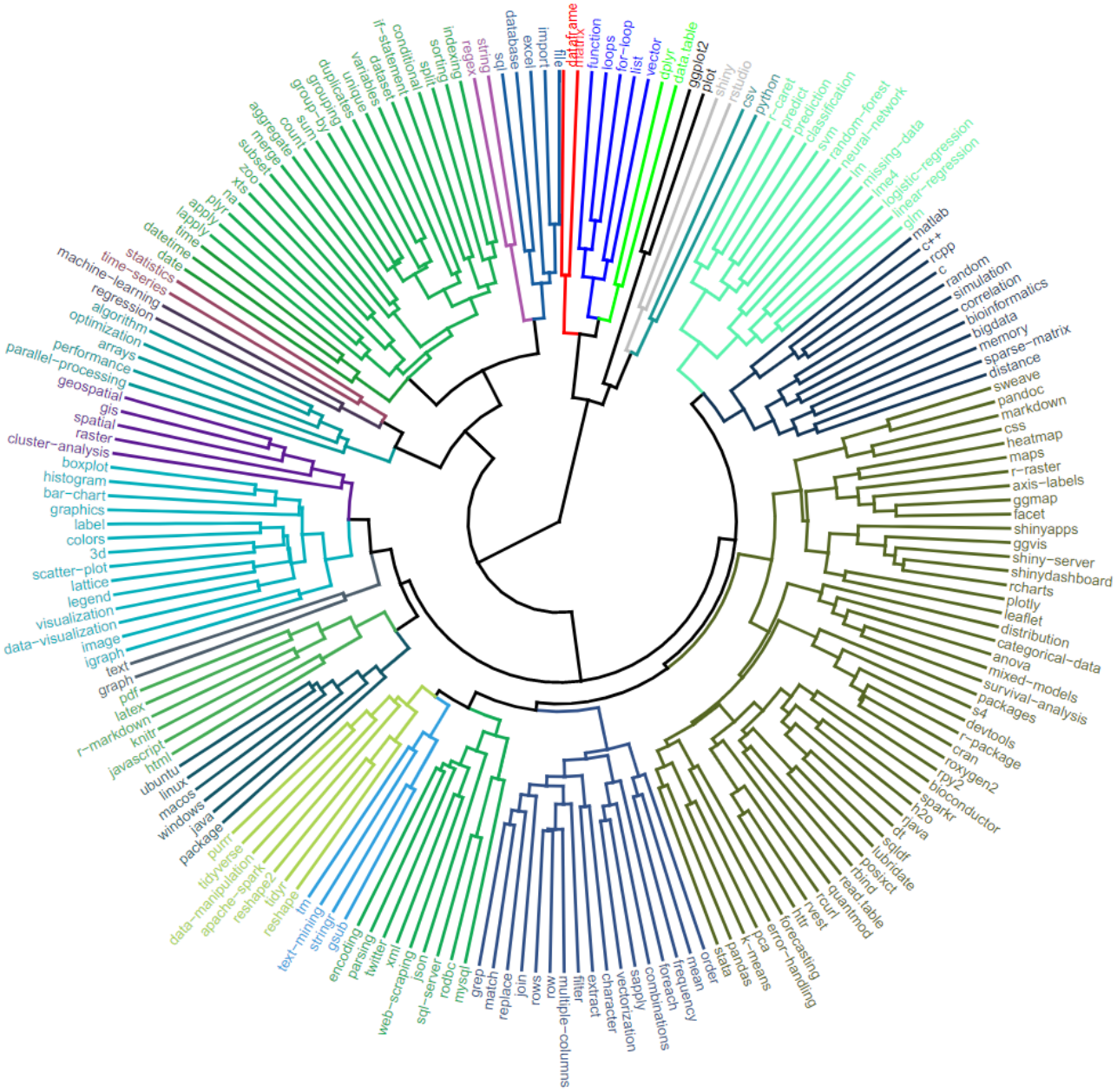
شکل ۴

در شکل ۴ نمودار دندوگرام سلسله مراتبی را می‌بینید که از یک کلاستر یا خوشه بزرگ به نام ریشه شروع شده و هرچه پایین تر بروید خوشه‌ها کوچکتر شده و در نهایت به خود تگ‌ها می‌رسیم. این نمودار در یک نگاه نتیجه الگوریتم خوشه بندی را در نهایت ظرفیت خوشه‌ها، به نمایش درآورده است.



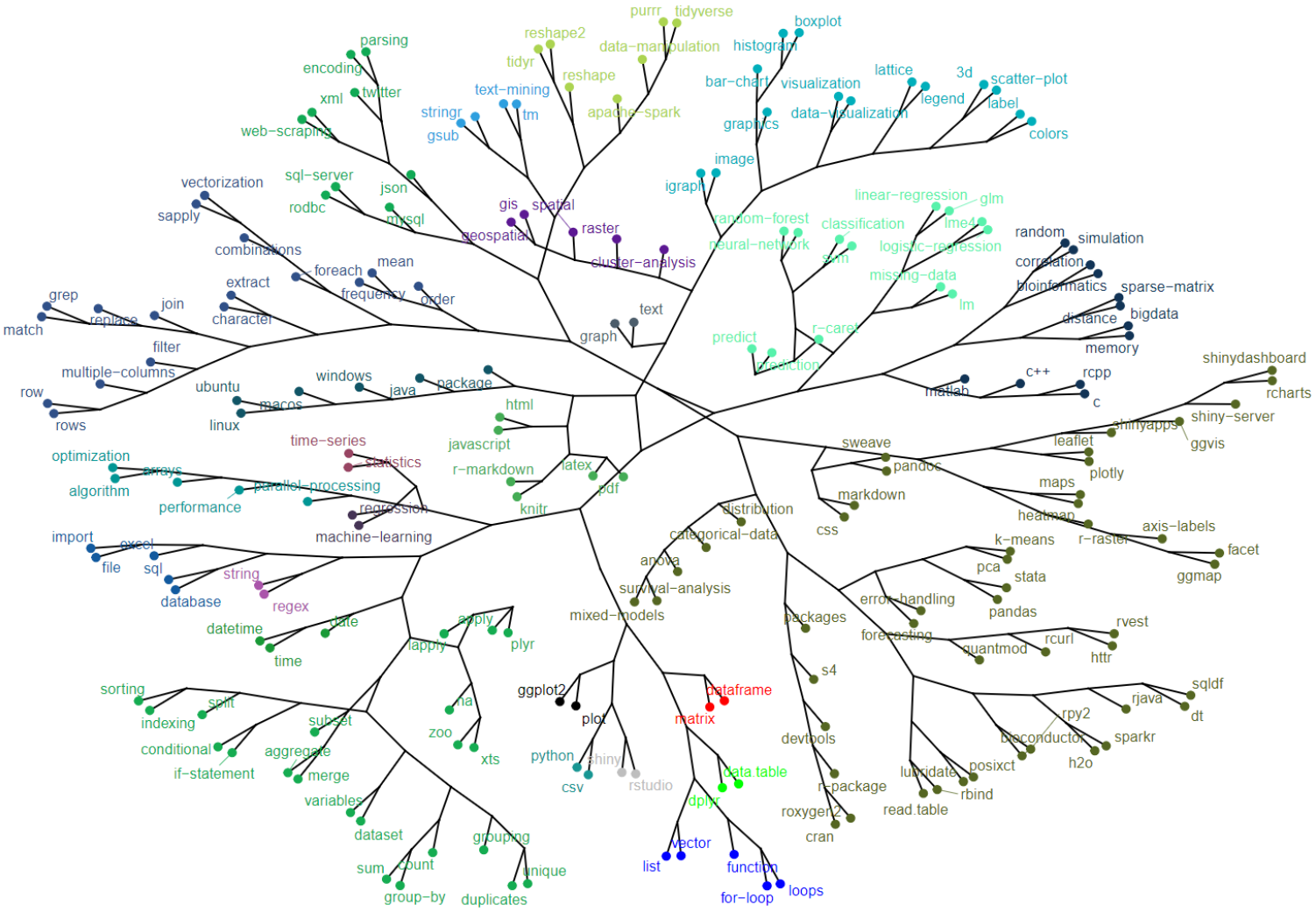
شکل ۵

در نمودار شکل ۵ می‌توان ۲۵ خوشه بندی را مشاهده کرد. در این نمودار هرچه به سمت پایین نگاه کنیم تعداد خوشه‌ها بیشتر و بیشتر می‌شود. نتایج در قالب فایل pdf در پوشه results ذخیره شده است.



شکل ۶

نمودار شـکل ۶ یک نمودار دایره‌ای اسـت که به آن Fan نیز می‌گویند. خوشـه‌ها با رنگ‌های متفاوت از هم متمایز شده‌اند و شما می‌توانید ۲۵ خوشه‌بندی را در اینجا مشاهده کنید.



شکل ۷

در نهایت نمودار شکل ۷ بصورت درختی یا مدل گرافی تگ‌ها را به نمایش درآورده که در آن تگ‌های با فاصله کمتر در یک شاخه قرار گرفته‌اند که براحتی با نگاه کردن می‌توان شاخه ها یا خوشه‌ها را مشاهده کرد که باز با رنگ‌های متفاوت ۲۵ خوشه از آنها متمایز شده‌اند.