
بسم الله الرحمن الرحيم

پروژه درس داده کاوی

خوشه‌بندی

استاد محترم درس

جناب آقای دکتر پاینده

دانشجو

آیدا اعلابیکی

در ابتدا بر خود واجب می‌دانم از زحمات استاد ارجمندم جناب آقای دکتر پاینده، کمال تشکر را به جا آورم.

خوشه‌بندی

4-1- مقدمه

خوشه‌بندی، گروه‌بندی نمونه‌های مشابه با هم در یک مجموعه‌داده می‌باشد. تجزیه و تحلیل خوشه‌ای، روشی برای گروه‌بندی داده‌ها یا مشاهدات باتوجه به شباهت‌ها با درجه نزدیکی آن‌ها است. از طریق تجزیه‌وتحلیل خوشه‌ای، داده‌ها به دو رده‌ی همگن و ناهمگن (متمایز) تقسیم‌بندی می‌شوند. در روش خوشه‌بندی، هیچ رده‌ای از قبل وجود ندارد و در واقع متغیرها به‌صورت مستقل و وابسته تقسیم نمی‌شوند، بلکه به دنبال گروه‌هایی از داده‌ها هستیم که به هم شباهت دارند و با پیدا کردن این شباهت‌ها، توزیع و همبستگی بین داده‌ها را کشف می‌کنیم. در خوشه‌بندی داده‌هایی که در هر گروه قرار می‌گیرند، شباهت بسیار زیادی با یکدیگر دارند در حالیکه با داده‌های سایر گروه‌ها تفاوت چشم‌گیری دارند.

مسئله مهم در خوشه‌بندی عبارت است از: توزیع داده‌ها به k گروه مختلف که داده‌های هر گروه با یکدیگر مشابه بوده و داده‌های گروه‌های مختلف با یکدیگر نامتشابه باشند. در این تقسیم‌بندی، داده‌های گروه‌های مختلف باید با یکدیگر حداکثر تفاوت ممکن را با هم داشته باشند و داده‌های موجود در یک گروه باید بسیار به هم شبیه باشند. این تشابه یا عدم تشابه بر اساس معیارهای اندازه‌گیری فاصله تعریف می‌شود.

برخلاف رده‌بندی در خوشه‌بندی گروه‌ها از قبل مشخص نمی‌باشند و همچنین معلوم نیست که بر حسب کدام خصیصه‌ها گروه‌بندی صورت می‌گیرد. در نتیجه پس از انجام خوشه‌بندی باید یک فرد خبره خوشه‌های ایجاد شده را تفسیر کند و در بعضی مواقع لازم است که پس از بررسی خوشه‌ها بعضی پارامترهایی که در خوشه‌بندی درنظر گرفته‌شده ولی بی‌ربط بوده یا اهمیت چندانی ندارد حذف شده و جریان خوشه‌بندی از اول صورت گیرد. پس از این که داده‌ها به چند گروه منطقی و توجیه‌پذیر تقسیم شدند از این تقسیم‌بندی می‌توان برای کسب اطلاعات در مورد داده‌ها یا تقسیم داده‌های جدید استفاده کنیم. کیفیت نتیجه‌های خوشه‌بندی بستگی به روش اندازه‌گیری شباهت به‌کار رفته و همچنین پیاده سازی آن روش دارد. در روند کلی هر خوشه‌بندی، چهار مرحله اصلی وجود دارد:

1- انتخاب یا استخراج خصیصه‌ها

2- طراحی یا انتخاب الگوریتم خوشه‌بندی
3- سنجش اعتبار خوشه‌بندی
4- تحلیل نتیجه‌ها.

خوشه‌بندی در موارد زیر مورد استفاده قرار می‌گیرد:

- تجزیه و تحلیل شباهت یا عدم شباهت: تجزیه‌وتحلیل این‌که کدام نقاط داده در یک نمونه به یکدیگر نزدیک‌تر می‌باشند.
- کاهش بُعد: اندازه و بُعد داده‌ها را می‌توان به وسیله خوشه‌بندی کاهش داد. این کاربرد بیش‌تر به‌عنوان پیش‌پردازش داده‌ها مورد استفاده قرار می‌گیرد.

خوشه‌بندی در زمینه‌های زیر کاربرد دارد:

- هوش تجاری
- علوم اقتصادی
- شناسایی متن
- تشخیص الگو
- جست‌وجوی وب
- پردازش تصویر
- بازاریابی و بیمه
- مطالعات زمین لرزه
- زیست‌شناسی
- علوم پزشکی و ژنتیک
- روان شناسی و جامعه شناسی

۴-۲- نقاط قوت روش خوشه‌بندی

- قدرت روش خوشه‌بندی به غیر مستقیم بودن آن است بدین معنی که روش را می‌توان حتی هنگامی که هیچ نوع اطلاعات قبلی از ساختار داخلی دادگان‌ها

نداریم استفاده نمود. از این روش می‌توان برای کشف الگوهای پنهان و بهبود عملکرد روش‌های مستقیم نیز استفاده نمود.

- خوشه‌بندی را می‌توان برای داده‌های گوناگون استفاده نمود. با انتخاب درست اندازه، فاصله‌های گوناگون خوشه‌بندی را می‌توان برای بیش‌تر انواع داده‌ها استفاد نمود.
- استفاده از این روش آسان است. در این روش لازم نیست که بعضی از زمینه‌ها را به‌عنوان ورودی و بعضی دیگر را به‌عنوان خروجی درنظر بگیریم.

۴-۳- نقاط ضعف روش خوشه‌بندی

- انتخاب اندازه‌های دقیق فاصله‌ها و وزن‌ها کار آسانی نمی‌باشد. در این روش به پارامترهای اولیه نظیر تعداد خوشه‌ها، کمینه نزدیکی و خوشه‌های اولیه حساس است.
- تفسیر نتیجه‌های این روش می‌تواند مشکل باشد و به‌طور معمول نیاز به تحلیل افراد خبره است.

۴-۴- تعیین تعداد خوشه

استقلال یا وابستگی تمام متغیرها، در تعیین تعداد خوشه‌ها مؤثر است. اگر تمام متغیرها کاملاً مستقل باشند هیچ خوشه‌ای ایجاد نمی‌شود، برعکس اگر تمام متغیرها وابسته باشند آن‌گاه تمام داده‌ها تشکیل یک خوشه می‌دهند. در شرایط بین استقلال و وابستگی کامل، ما نمی‌دانیم که واقعاً چند خوشه وجود دارد. در انتخاب تعداد خوشه‌ها، تحلیل‌گر نقش به سزایی دارد. با توجه به کاربردهای متفاوت، ممکن است به تعداد خوشه‌ی بیش‌تر یا کم‌تر نیاز باشد. در بسیاری از مواقع با یک مقدار k خوشه‌بندی را انجام داده و نتیجه را ارزیابی می‌کنیم و به دنبال k دیگر می‌رویم. بعد از هر تکرار، ارزش نتیجه‌ها را به وسیله اندازه‌گیری میزان متوسط فاصله‌ها در داخل خوشه‌ها و میزان متوسط فاصله‌ها بین مراکز خوشه‌ها و یا روش‌های دیگر

بررسی می‌کنیم. گاهی خوشه‌ها به‌وسیله قضاوت‌های ذهنی تحلیل‌گر مورد ارزیابی قرار می‌گیرند تا در کاربردهای خاصی استفاده شوند.

۴-۵- ارزیابی اعتبار در خوشه‌بندی

با توجه به این‌که نتیجه‌های حاصل از پیاده‌سازی الگوریتم‌های خوشه‌بندی بر داده‌ها با توجه به انتخاب پارامترها (شامل تعداد خوشه‌ها) می‌تواند بسیار متفاوت از یکدیگر باشد، لذا باید با در نظر گرفتن شاخص‌هایی، اعتبار و عملکرد خوشه‌ها مقایسه گردند تا خوشه‌هایی که بهترین تناسب را با داده‌ها دارند انتخاب شوند (آربلیتز، 2013).

دو معیار پایه اندازه‌گیری پیش‌نهاد شده برای ارزیابی و انتخاب خوشه‌های بهینه، شامل تراکم و جدایی است. در تراکم، داده‌های متعلق به یک خوشه باید تا حد ممکن به یکدیگر نزدیک باشند. معیار رایج برای تعیین میزان تراکم داده‌ها واریانس داده‌ها است. همچنین در معیار جدایی، خوشه‌ها باید به اندازه کافی از یکدیگر جدا باشند. در حالت کلی، جهت ارزیابی اعتبار در خوشه‌بندی از شاخص‌های درونی و بیرونی استفاده می‌گردد. شاخص‌های بیرونی از طریق ساختاری از پیش تعیین شده به ارزیابی خوشه‌ها می‌پردازد و برای محاسبه آن‌ها نیاز به داده‌های تاریخی است. در صورتی که شاخص‌های درونی، صرفاً از طریق اطلاعات درونی در داده‌های موجود اعتبار خوشه‌بندی را ارزیابی می‌کند. یکی از شاخص‌های درونی، شاخص سیلهوئت است که برای هر خوشه محاسبه می‌شود و در صورتی که به یک نزدیک باشد نشان می‌دهد که نمونه‌های درون خوشه بسیار شبیه به هم هستند و خیلی خوب خوشه‌بندی شده‌اند و در صورتی که نزدیک به صفر باشد، نشان می‌دهد تعداد نمونه‌های مرزی در خوشه زیاد است و مقدار منفی این معیار نشان می‌دهد که نمونه‌های درون خوشه شبیه به یکدیگر نبوده و عمل خوشه‌بندی نادرست صورت گرفته است.

- **شاخص درونی**

شاخص سیلهوئت: این شاخص به‌صورت فرمول 4-1 می‌باشد (کافمن و روسو، 1990):

$$\text{شاخص سیلهوئت} = \frac{\sum_{i=1}^{n} S(i)}{n}$$

$$S(i) = \frac{b(i) - a(i)}{\max(a(i); b(i))} \qquad (۱-۴)$$

$$a(i) = \frac{\sum_{j \in c_r} d_{ij}}{n_r - ۱}$$

$$b(i) = \min d_{ic_s} \quad s \neq r$$

$$d_{iC_s} = \frac{\sum_{j \in c_s} d_{ij}}{n_s}$$

که $a(i)$ میانگین عدم تشابه i امین مشاهده نسبت به بقیه مشاهده‌ها در خوشه C_S است. C_S , C_r نیز خوشه‌های s , r هستند. n_s, n_r نیز تعداد مشاهده‌های در خوشه‌های C_S , C_r هستند. شاخص مذکور بین دو مقدار ۱ -۱,- + قرار دارد و مقدار بزرگ‌تر برای این معیار نشان‌دهنده تعداد خوشه‌های مطلوب‌تر است.

- **شاخص بیرونی**

شاخص جاکارد: به منظور اندازه‌گیری ضریب شباهت بین دو مجموعه‌ی A و B از داده‌ها، ضریب شباهت جاکارد به صورت فرمول ۴-۲ تعریف می‌شود (جین و دوبس، ۱۹۸۸).

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \qquad (۲-۴)$$

ضریب مذکور بین دو مقدار ۰ و ۱ قرار دارد.

۴-۶- ماتریس تشابه و فاصله

به منظور رده‌بندی اشیا باید ملاکی برای اندازه‌گیری میزان تشابه یا فاصله بین آن‌ها در نظر گرفته شود. به این ترتیب اشیائی که به یکدیگر شبیه هستند می‌توانند در یک رده قرار بگیرند. برای نمایش تشابه چند شی، از ماتریس تشابه استفاده می‌شود. برای اندازه‌گیری عدم تشابه مابین دو سری داده، فاصله آن‌ها را محاسبه می‌کنند. می‌دانیم که ماتریس نمایشی منظم از اعداد و ارقام است که از تعدادی سطر و ستون تشکیل شده است. به این ترتیب ساختاری برای نمایش بهتر رابطه بین مجموعه‌ای از نقطه‌ها فراهم می‌شود. اگر تعداد سطرها و ستون‌های یک ماتریس برابر باشد، آن را ماتریس مربعی می‌گویند. اگر مقدارهای درون ماتریس فقط بر روی قطر اصلی قرار داشته باشند، ماتریس را قطری می‌نامند. همچنین اگر مقدارهای بالای قطر اصلی همگی صفر باشند، ماتریس را پایین مثلثی و به همین ترتیب مقدارهای پایین قطر اصلی صفر باشند، ماتریس را بالا مثلثی می‌گویند. برای معرفی ماتریس تشابه و فاصله ابتدا به معرفی تابع تشابه و فاصله می‌پردازیم.

۴-۶-۱- تابع تشابه

فرض کنید مجموعه‌داده از نقاط یک بعدی که آن‌ها را D می‌نامیم، داریم. اگر تابع S در شرایط زیر صدق کند، آن را تابع تشابه می‌نامیم.

۱) $S(x,y) \geq ۰ \qquad x,y \in D$

۲) $S(x,y) = ۱$

۳) $S(x,y) = S(y,x) \qquad \forall x,y \in D$

تابع تشابه تابعی نامنفی و برد آن فاصله بین ۰ تا ۱ است. همچنین رابطه آخر نشان می‌دهد که این تابع متقارن نیز هست. اگر رابطه آخر برای تابعی که به منظور اندازه‌گیری میزان تشابه به‌کار رفته، وجود نداشته باشد، آن را تابع نیمه تشابه می‌گویند.

۴-۶-۲- تابع فاصله

اگر d تابع با مقدارهای حقیقی و نامنفی باشد، در صورتی که خاصیت‌های زیر را برای آن صدق کند، می‌تواند به‌عنوان یک تابع فاصله به‌کار رود.

۱) $d(x,y) \geq ۰$

۲) $d(x,y) = ۰ \leftrightarrow x = y$

۳) $d(x,y) = d(y,x)$

۴) $d(x,y) < d(x,y) + d(y,z) \quad \forall x,y,z \in D$

۴-۶-۳- ماتریس تشابه

فرض کنید مجموعه D از نقاط $x_۱,\ldots,x_n$ تشکیل شده است. بر اساس تابع تشابه S، می‌توان برای هر زوج از اعضای D، تشابه را اندازه‌گیری کرد و در ساختاری به شکل ماتریس قرار داد. بنا بر این اگر $S(x_i,x_j) = S_{ij}$ به معنی تشابه بین دو نقط x_j, x_j باشد، این ماتریس به‌صورت زیر خواهد بود:

$$\begin{bmatrix} S_{۱۱} & \cdots & S_{۱n} \\ \vdots & \ddots & \vdots \\ S_{n۱} & \cdots & S_{nn} \end{bmatrix}$$

همان‌طور که مشاهده می‌کنید این ماتریس یک ماتریس مربع است. اگر تابع S خاصیت‌های تابع تشابه را داشته باشد می‌توان آن را به فرم ساده‌تر نمایش داد.

$$\begin{bmatrix} ۱ & S_{۱۲} & \cdots & S_{۱n} \\ & \ddots & \vdots & \vdots \\ & & ۱ & S_{۲n} \\ & & & ۱ \end{bmatrix}$$

از آن‌جایی که تابع تشابه متقارن است، در نتیجه ماتریس تشابه را می‌توان به صورت یک ماتریس بالا یا پایین مثلثی نمایش داد. توجه داشته باشید که ماتریس تشابه یک ماتریس بالا یا پایین مثلثی نیست بلکه برای نمایش آن از این حالت استفاده می‌کنیم.

۴-۶-۴- ماتریس فاصله

با توجه به مطالبی که در مورد ماتریس تشابه گفته شد، می‌توان ماتریس فاصله را نیز برای نقاط محاسبه کرد. کافی است که برای هر زوج از نقطه‌ها، تابع فاصله مورد نظر را به‌دست آورد و در یک ماتریس قرارا داد. در ماتریس فاصله لازم نیست که مقدارهای حقیقی غیرمنفی باشند، ممکن است عناصر اعداد صفر، منفی یا اعداد مختلط بگیرند. اگرچه در اکثر مواقع ماتریس فاصله در روی قطر اصلی دارای مقدار صفر است ولی می‌تواند غیر صفر را نیز روی قطر اصلی داشته باشند.

۴-۷- روش‌های اصلی خوشه‌بندی

روش‌های اصلی خوشه‌بندی عبارت‌اند از:

1. روش‌های افرازی
2. روش‌های سلسله مراتبی
3. روش‌های مبتنی بر چگال
4. روش مشبکی‌مبنا

۴-۷-۱- روش‌های افرازی

فرض کنید یک دادگان با n نمونه داریم. یک روش افرازی، k افراز از این داده‌های نمونه درست می‌کند به‌طوری‌که هر افراز یک خوشه را نشان می‌دهد و $k < n$. پس داده‌های نمونه در k گروه خوشه‌بندی شده و دارای دو شرط زیر می‌باشند:

- هر گروه حداقل یک نمونه دارد.

- هر نمونه تنها به یک گروه تعلق دارد. این شرط تنها در روش‌های فازی می‌تواند قابل انعطاف باشد.

در روش افرازی برای k معلوم، یک افراز ابتدایی ایجاد می‌شود. سپس یک روش جابه‌جایی تکراری را به‌کار برده که تلاش به بهبود افرازبندی دارد. به این صورت که نمونه‌ها را از یک گروه به دیگر گروه‌ها می‌برد. یک معیار عمومی برای افرازبندی خوب این است که نمونه‌ها در یک خوشه به هم نزدیک یا به یکدیگر وابسته باشند و در مقابل نمونه‌ها در خوشه‌های مختلف، از یکدیگر دور یا تا حد امکان متفاوت باشند.

برای دستیابی به خوشه‌بندی بهینه در روش افرازی، به شمارش کامل همه افرازهای ممکن نیاز خواهد بود یعنی تمام حالات ممکن باید بررسی شوند که این روش برای دادگان‌های بزرگ ناممکن است، بنا بر این الگوریتم‌های زیر برای بررسی این گونه موارد استفاده می‌شوند:

1. الگوریتم *k-means*
2. الگوریتم *k-medoids*

این روش‌ها برای یافتن خوشه‌هایی به شکل کره در دادگان‌های کوچک تا متوسط به‌خوبی کار می‌کنند اما برای یافتن خوشه‌هایی با اشکال پیچیده و یا دارای مجموعه‌داده‌های بزرگ، باید توسعه داده شوند.

۴-۷-۱-۱- الگوریتم *k- means*

این روش، علی‌رغم سادگی یک روش پایه‌ای برای بسیاری از روش‌های خوشه‌بندی دیگر(مانند خوشه‌بندی فازی) محسوب شده و الگوریتمی بسیار ساده، قابل فهم و به‌طور منطقی قابل مقیاس‌بندی است. خوشه‌بندی *k-means* در سال ۱۹۶۵ توسط فورجی ارائه شد که روش رده‌بندی بدون ناظر است. روش‌ها و الگوریتم‌های متعددی برای تبدیل نمونه‌ها به گروه‌های هم‌شکل یا مشابه وجود دارد. الگوریتم *k-* میانگین یکی از ساده‌ترین و

محبوب‌ترین الگوریتم‌هایی است که در داده‌کاوی به‌خصوص در حوزه یادگیری بدون ناظر به کار می‌رود.

در *k-means* عملاً مجموعه‌داده‌ها به تعداد خوشه‌های از پیش تعیین شده تقسیم می‌شوند. ایده اصلی در این الگوریتم تعریف k مرکز برای هر یک از خوشه‌ها است. بهترین انتخاب برای مراکز خوشه‌ها در الگوریتم *k-means* قرار دادن آن‌ها در فاصله هر چه بیش‌تر از یکدیگر است. پس از آن هر نمونه در مجموعه‌داده به نزدیک‌ترین مرکز خوشه تخصیص می‌یابد. مرحله‌های الگوریتم *k-means* به‌صورت زیر است:

۱. ابتدا k مشاهده‌ی اولیه به‌عنوان هسته انتخاب می‌کنیم.

۲. متوسط مقدار خوشه را محاسبه می‌کنیم.

۳. در صورتی که فاصله مشاهده مورد نظر از میانگین خوشه خود زیاد و به خوشه دیگری نزدیک‌تر باشد این مشاهده به خوشه‌ای که نزدیک‌تر است اختصاص می‌یابد.

۴. این کار تا کمینه‌شدن تابع خطا، که به‌طور معمول مجموع فاصله‌های مشاهدات از مرکز خوشه خودش است، با تغییر نیافتن اعضای خوشه‌ها ادامه می‌یابد.

این روش را هنگامی می‌توان استفاده کرد که مفهوم میانگین در یک مجموعه نمونه قابل تعریف باشد. برای رفع مشکلات الگوریتم *k-means* با تغییراتی روی آن این روش را توسعه داده و به جای استفاده از میانگین به‌عنوان مراکز خوشه از نماهای خوشه به‌عنوان نماینده یک خوشه استفاده می‌شود. این روش *k-modes* نام دارد. اگر به‌جای مرکز یا وسط یک خوشه، از میانه آن خوشه استفاده کنیم، آن‌گاه روش نسبت به داده‌های دور از مرکز حساس نمی‌شود زیرا میانه از مقدارهای بزرگ تأثیر نمی‌پذیرد.

عیب‌های الگوریتم *k-means*:

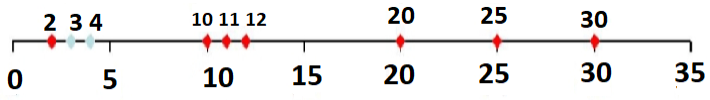
- معیاری برای تعیین تعداد و مراکز اولیه خوشه‌ها وجود ندارد.
- اگر در خوشه‌ای هیچ داده‌ای وجود نداشته باشد راهی برای تغییر و بهبود آن وجود ندارد.
- برای داده‌های دم‌سنگین مناسب نیست.

- جواب نهایی به انتخاب مراکز اولیه خوشه‌ها بستگی دارد.
- این روش برای کشف خوشه‌هایی با شکل‌های پیچیده مناسب نیست.
- امکان تولید خوشه‌های خالی توسط این روش وجود دارد.

مثال ۴-۱- فرض کنید سن ۹ نفر به‌صورت {۲,۴,۱۰,۱۲,۳,۲۰,۳۰,۱۱,۲۵}است. با استفاده از روش k- میانگین، داده‌های به‌دست آمده را به دو خوشه تقسیم کنید؟

گام اول: ابتدا باید دو عدد تصادفی به‌عنوان مراکز خوشه‌ها تولید کنیم. فرض کنید اعداد ۳ و۴ مراکز اولیه خوشه‌ها باشند.

حال هر یک از داده‌ها را با توجه به فاصله‌ی هر یک ازآن‌ها از این مراکز به یکی از خوشه‌ها نسبت می‌دهیم. به‌عنوان مثال عدد ۲ به مرکز ۳ و عدد ۱۰ به مرکز ۴ نزدیک‌تر است.

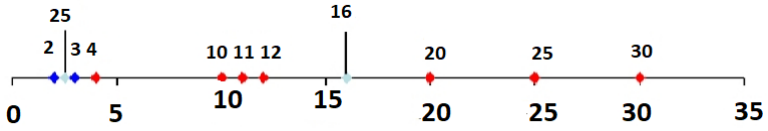


بنا بر این با ادامه‌ی روند بالا، دو خوشه‌ی زیر به‌دست می‌آیند.

$C_1 = \{2,3\}$

$C_2 = \{4,10,12,20,30,11,25\}$

گام دوم: در این گام، مراکز جدید هر خوشه را محاسبه کرده و با توجه به مراکز جدید، دوباره عمل واگذاری داده‌ها را انجام می‌دهیم. مراکز جدید ۲/۵ و ۱۶ خواهند بود.



مشاهده می‌شود عدد 4 که در مرحله قبل به خوشه دوم نسبت داده شد، در این مرحله به مرکز خوشه اول نزدیک‌تر است. در نتیجه باید عدد 4 را در خوشه اول قرار دهیم.

خوشه‌های جدید به شکل زیر خواهند بود:

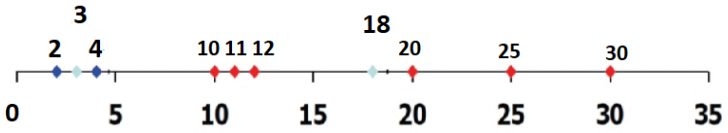
$C_۱ = \{۲,۳,۴\}$

$C۲ = \{۱۰,۱۲,۲۰,۳۰,۱۱,۲۵\}$

روند بالا را تا زمانی که شرط توقف برقرار شود ادامه می‌دهیم.

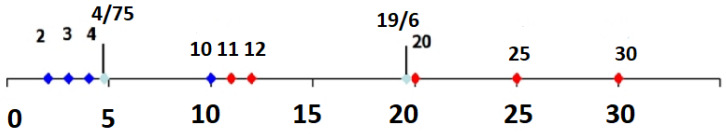
مرحله‌های بعدی به‌صورت زیر خواهند بود:

$$\mu_۱ = ۳, \mu_۲ = ۱۸ \rightarrow C_۱ = \{۲,۳,۴,۱۰\} \quad C_۲ = \{۱۲,۲۰,۳۰,۱۱,۲۵\}$$



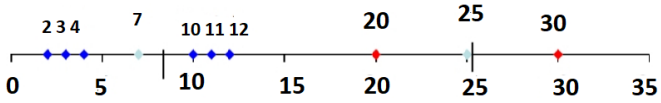
مرحله بعدی به‌صورت زیر خواهد بود:

$$\mu_۱ = ۴/۷۵, \mu_۲ = ۱۹/۶ \rightarrow C_۱ = \{۲,۳,۴,۱۰,۱۱,۱۲\} \quad C_۲ = \{۲۰,۳۰,۲۵\}$$



و مرحله نهایی به‌صورت زیر است که در آن مراکز ثابت شده است و خوشه‌ها تغییر نمی‌کنند.

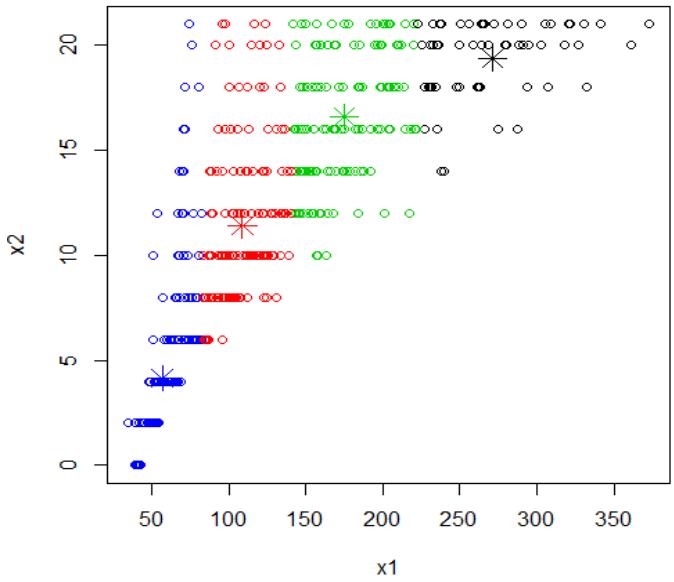
$$\mu_۱ = ۷, \mu_۲ = ۲۵ \rightarrow C_۱ = \{۲,۳,۴,۱۰,۱۱,۱۲\} \quad C_۲ = \{۲۰,۳۰,۲۵\}$$



مثال۴-۲- در مثال۳-۳ به توضیح داده‌های وزن جوجه پرداختیم. همان‌طور که اشاره شد داده‌ها مربوط به ۵۰ جوجه و ۴ رژیم غذایی متفاوت است. حال روش *k-means* را برای داده‌های وزن جوجه با استفاده از بسته نرم‌افزاری *amap* پیاده‌سازی می‌کنیم. (کد۱)

```
> data(ChickWeight)
> ChickWeight
> install.packages("amap")
> library(amap)
> x1=ChickWeight$weight
> x2=ChickWeight$Time
> x=cbind(x1,x2)
> x
> cl<-Kmeans(x,4, iter.max =85, method = "euclide
an")
> plot(x, col = cl$cluster)
> points(cl$centers, col = 1:4, pch = 8, cex=2)
```

خروجی دستورها را در شکل 4-1 مشاهده می کنیم.



شکل 4-1: خروجی الگوریتم *k-means*

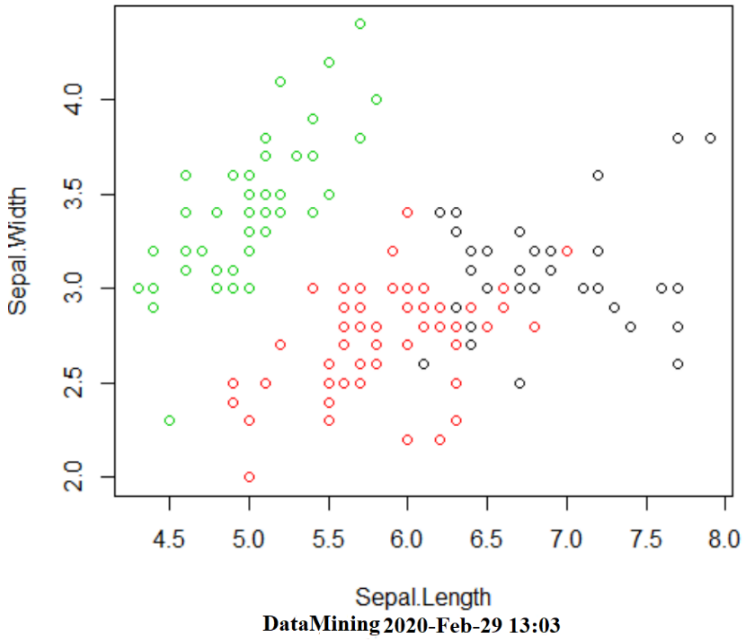
در نمودار بالا تعداد خوشه‌ها را برابر 4 و حداکثر تعداد تکرار تا رسیدن به بهترین خوشه را 85 و معیار فاصله‌ی نقطه تا میانگین، فاصله اقلیدسی در نظر گرفته شده است.

مثال 4-3- داده‌ای گل زنبق را در نظر بگیرید. همان‌طور که می‌دانیم این مجموعه‌داده دارای یک متغیر به‌نام *Species* است و گروه هر یک از گل‌ها را در سه سطح مشخص می‌کند که آن را از مجموعه‌داده‌ها حذف کرده‌ایم. پس تعداد خوشه‌ها را در تابع *kmeans*() برابر سه در نظر خواهیم گرفت. تعداد اعضای هر خوشه همان‌طور که مشاهده می‌کنیم برابر با 96,21,33 است.

میانگین متغیرها در هر خوشه حساب شده و خروجی *Clustering Vector* درحقیقت خوشه‌ی هر نمونه را تعیین می‌کند که این الگوریتم مجموع توان دوم درون خوشه‌ها یعنی *Within cluster sum of square by cluster* را برای هر خوشه کمینه می‌کند و مجموع توان دوم بیرون خوشه‌ای را بیشینه خواهد کرد. این مدل 79٪ از کل واریانس داده‌ها را بیان می‌کند و هر چه‌قدر تعداد خوشه‌ها بیش‌تر باشد این عدد نیز به 100٪ نزدیک خواهد شد. برای مقایسه‌ی خوشه‌های به‌دست آمده از این مدل و گروه‌بندی واقعی که توسط متغیر *Species* تعیین شده است از تابع *table*() استفاده کرده‌ایم. بنا بر این از گونه اول 17 مورد در خوشه دوم و 33 مورد در خوشه سوم قرار گرفته‌اند و از گونه دوم 46 مورد در خوشه اول و 4 مورد در خوشه دوم قرارگرفته و برای گونه سوم تمام موارد در خوشه اول قرار گرفته است. (کد2)

```
> data(iris)
> iris2=iris
> iris2$Species=NULL
> kmeans.result=kmeans(iris2,3)
> kmeans.result
K-means clustering with 3 clusters of sizes 96, 21, 33
Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.314583    2.895833     4.973958   1.7031250
2     4.738095    2.904762     1.790476   0.3523810
3     5.175758    3.624242     1.472727   0.2727273

Clustering vector:
  [1] 3 2 2 2 3 3 3 3 2 2 3 3 2 2 3 3 3 3 3 3 3 3 3 3
2 2  3 3 3 2 2 3 3 3 2 3 3 3 2 3 3
 [42] 2 2 3 3 2 3 2 3 3 1 1 1 1 1 1 1 2 1 1 2 1 1 1 1
1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [83] 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 2 1 1 1 1 1 1 1
1 1  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[124] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1  1
Within cluster sum of squares by cluster:
[1] 118.651875  17.669524   6.432121
 (between_SS / total_SS =  79.0 %)
Available components:
[1] "cluster"      "centers"      "totss"        "with
inss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifau
lt"
> table(iris$Species, kmeans.result$cluster)
              1  2  3
  setosa      0 17 33
  versicolor 46  4  0
  virginica  50  0  0
> plot(iris2[c("Sepal.Length","Sepal.Width")],col=kmea
ns.result$cluster,sub=paste("DataMining",format(Sys.ti
me(),"%Y-%b-%d %H:%S")))
> points(kmeans.result$centers[,c("Sepal.Length", "Sep
al.Width")], col=1:3, pch=8, cex=2)
```

برای رسم نمودار دو بُعدی این خوشه‌بندی، از تابع $plot()$ استفاده کرده و توسط تابع $points()$ می‌توان مراکز هر خوشه را مشخص کرد.



شکل 4-2: نمودار دو بُعدی خوشه‌بندی با داده‌های گل زنبق

۴-۷-۱-۲- الگوریتم k- $medoids$

الگوریتم k- $means$ به داده‌های دور افتاده حساس است و از آن‌جا که فاصله‌ی چنین داده‌هایی با اکثر داده‌ها بسیار زیاد است، هنگام قرار گرفتن در یک خوشه به‌صورت چشم‌گیری باعث انحراف و تغییر میانگین خوشه می‌شود. این موضوع به صورت غیر عمدی روی انتساب نمونه‌ها دیگر به خوشه‌ها تأثیر می‌گذارد. به‌جای در نظر گرفتن مقدار میانگین نمونه‌ها موجود در خوشه به‌عنوان یک نقطه ارتجاعی، می‌توان از نمونه‌های واقعی برای نمایش خوشه‌ها استفاده کرد. مدوئیدها نمونه‌هایی هستند که در مرکزی‌ترین محل یک خوشه قرار دارند. در این الگوریتم می‌توان به‌جای استفاده از مرکز یک خوشه به‌عنوان مرجع از مدوئیدها استفاده کرد. روش افراز به‌دنبال کمینه کردن مجموع اختلاف میان هر نمونه و نماینده نظیر آن است. گروه‌بندی n نمونه در k خوشه با کمینه کردن خطای مطلق، ایده اصلی روش k_مدوئید است.

یکی از الگوریتم‌های رایج در میان روش‌های خوشه‌بندی k ـمدوئید، الگوریتم PAM است. در این الگوریتم همانند الگوریتم k- $means$ در ابتدا برخی از نمونه‌ها به‌صورت اختیاری به‌عنوان نمایندگان اولیه انتخاب می‌شوند. پس از آن بررسی می‌شود که آیا جایگزین نمودن یک نمونه غیر نماینده یا یک نمونه نماینده، کیفیت خوشه‌بندی را بهبود می‌بخشد یا خیر. کلیه‌ی جایگزینی‌های ممکن آزمایش می‌شوند. این جایگزینی نمونه‌ها تا هنگامی که خوشه‌بندی بهبود یابد، ادامه می‌یابد. از آن‌جا که یک نماینده یا به‌عبارتی یک مدوئید نسبت به مقدار میانگین کم‌تر تحت تأثیر داده‌های دور افتاده قرار می‌گیرد، روش k – مدوئید در مواجه با داده‌های دور افتاده و خطا مقاوم‌تر از روش k- $means$ است.

اما در استفاده از هر دو روش لازم است کاربر مقدار k یعنی تعداد خوشه‌ها را مشخص کند. الگوریتم PAM که در رده‌ی خوشه‌بندی مبتنی‌بر افراز می‌گنجد، عملکرد خوبی را در مواجه با مجموعه‌داده‌های کوچک دارد. اما قابلیت اجرایی مقیاس‌پذیر را بر روی داده‌های بزرگ ندارد. جهت کار با داده‌های با اندازه بالا می‌توان از روشی مبتنی‌بر نمونه‌گیری با نام $CLARA$ استفاده کرد. در این روش به‌جای این‌که از همه داده‌ها استفاده شود، تنها بخشی از آن‌ها با روش‌های نمونه‌گیری برای بررسی انتخاب می‌شوند. الگوریتم $CLARA$ عمل خوشه‌بندی را چندین بار بر روی نمونه‌های مختلف اجرا می‌کند و بهترین خوشه‌بندی به‌عنوان خروجی برگردانده می‌شود. این الگوریتم به تعداد نمونه‌ها یا به‌عبارتی دیگر به اندازه‌ی نمونه‌گیری بستگی دارد. الگوریتم PAM در میان یک مجموعه‌داده‌ها به‌دنبال k نماینده است، درحالی‌که الگوریتم $CLARA$ همین کار را بر روی نمونه‌های منتخبی از داده‌ها انجام می‌دهد. در الگوریتم PAM جهت یافتن نمایندگان بهتر، جانشینی و تعویض همه نمونه‌ها با نمایندگان جاری بررسی می‌شود، در حالی‌که در الگوریتم $CLARA$ این بررسی تنها محدود به داده‌های نمونه‌گیری شده به‌صورت تصادفی می‌شود. پیچیدگی محاسبات PAM در هر تکرار متناظر با $O(ks2 + k(n-k))$ می‌باشد که k تعداد خوشه‌ها، S تعداد نمونه انتخاب شده و n کل نمونه‌ها می‌باشد. لذا پیچیدگی الگوریتم کاهش می‌یابد و از مرتبه تعداد نمونه انتخاب شده است.

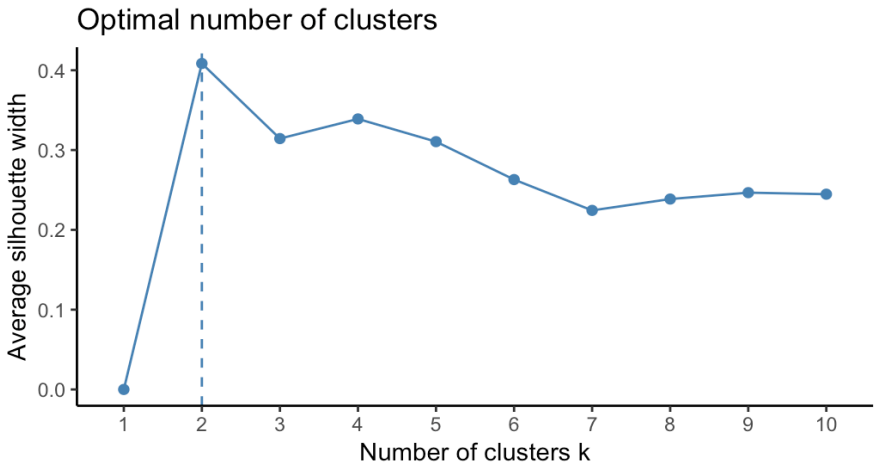
مثال۴-۴- مجموعه‌داده‌های *USArrests* شامل اطلاعاتی مربوط به میزان قتل، اذیت و آزار، جمعیت شهری و میزان تجاوز در ایالات مختلف ایالات متحده‌ی آمریکا را نشان می‌دهد. روی این دادگان الگوریتم *PAM* را پیاده سازی می‌کنیم. (کد۳)

```
> library(datasets)
> data(USArrests)
> df=scale(USArrests)
> head(df,n=3)
```

از طریق دستورهای زیر در نرم‌افزار *R* می‌توان تعداد خوشه‌ها را براورد کرد.

```
> install.packages(c("cluster","factoeextera"))
> library(cluster)
> library(factoextra)
#the optimal number of clusters
> fviz_nbclust(data,pam.method="silhouett")
+geom_vline(xintercep=3)
```

خروجی این دستورها به شکل ۴-۳ نشان داده شده است:



شکل۴-۳: تعداد خوشه‌ها

دستورهای زیر، الگوریتم *PAM* را به ازای $k = ۲$ در نرم‌افزار *R* محاسبه می‌کند:

```
#pam(x,metric="euclidean",stand=FALSE)
> PAM=pam(df,2)
```

```
> Print(PAM)
```

خروجی اجرای این تابع در نرم‌افزار به‌صورت زیر می‌باشد:

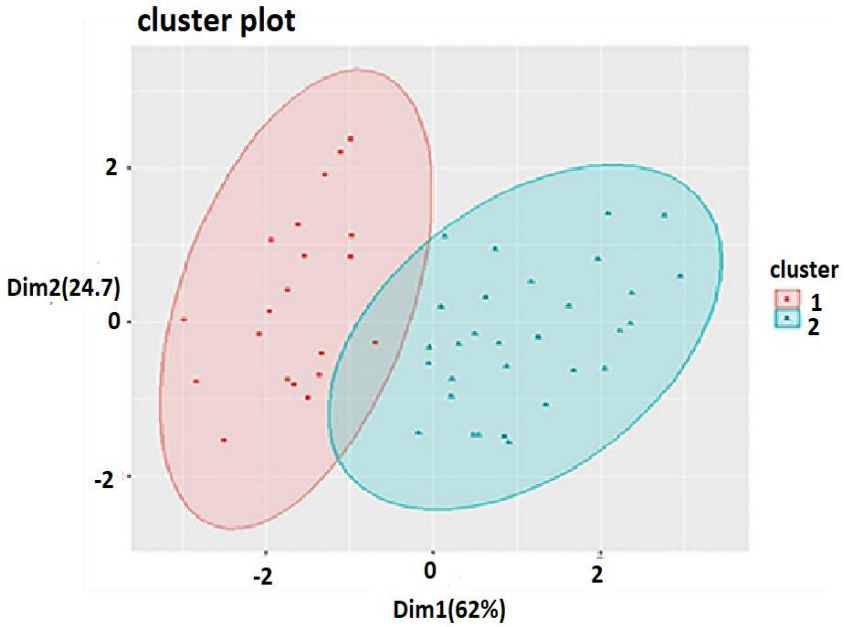
```
Medoids:
           ID     Murder    Assault   UrbanPop       Rape
New Mexico 31  0.8292944  1.3708088  0.3081225  1.1603196
Nebraska   27 -0.8008247 -0.8250772 -0.2445636 -0.5052109
Clustering vector:
       Alabama         Alaska        Arizona       Arkansas     California       Colorado    Connecticut
             1              1              1              2              1              1              2
      Delaware        Florida        Georgia         Hawaii          Idaho       Illinois        Indiana
             2              1              1              2              2              1              2
          Iowa         Kansas       Kentucky      Louisiana          Maine       Maryland  Massachusetts
             2              2              2              1              2              1              2
      Michigan      Minnesota    Mississippi       Missouri        Montana       Nebraska         Nevada
             1              2              1              1              2              2              1
 New Hampshire     New Jersey     New Mexico       New York North Carolina   North Dakota           Ohio
             2              2              1              1              1              2              2
      Oklahoma         Oregon   Pennsylvania   Rhode Island South Carolina   South Dakota      Tennessee
             2              2              2              2              1              2              1
         Texas           Utah        Vermont       Virginia     Washington  West Virginia      Wisconsin
             1              2              2              2              2              2              2
       Wyoming
             2
Objective function:
   build     swap
1.441358 1.368969

Available components:
 [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"  "clusinfo"   "silinfo"    "diss"
 [9] "call"       "data"
```

با توجه به خروجی نرم‌افزار، قسمت اول خروجی یعنی مدوئید نشان دهنده‌ی ماتریسی با سطرهای مدوئید و ستون‌های مربوط به متغیرها است. قسمت دوم خروجی یعنی *Clustering Vector* یک بردار از اعداد حقیقی است که نشان دهنده‌ی اختصاص نقاط به هر یک از دو خوشه می‌باشد که به‌طور خلاصه از طریق دستورهای زیر در نرم‌افزار قابل ملاحظه است:

```
> d=cbind(USArrests,cluster=PAM$cluster)
> head(d,n=3)

        Murder Assault UrbanPop Rape cluster
Alabama   13.2     236       58 21.2       1
Alaska    10.0     263       48 44.5       1
Arizona    8.1     294       80 31.0       1
>
```

خوشه‌های تشکیل شده از این داده‌ها را می‌توان با دستورهای زیر ملاحظه کرد:

```
#visualizing PAM clusters
> library(factoextera)
> fviz_cluster(PAM,geom="point",ellipse.type="norm")
```



شکل ۴-۴ خوشه‌های تشکیل شده

مثال 4–5– داده‌های گل زنبق را در نظر بگیرید. در مثال قبل از تابع $pam()$ برای خوشه‌بندی استفاده نمودیم. حال در این مثال از تابع $pamk()$ استفاده می‌کنیم. ابتدا تعداد خوشه‌های بهینه را در بازه‌ی [۲,۱۰] به‌دست آورده‌ایم که برابر با 2 شده است. (کد4)

```
> library(fpc)
> pamk.result=pamk(iris2,2:100)
> pamk.result$nc
[1] 2
```

سپس به مشخص کردن مراکز خوشه‌ها می‌پردازیم:

```
> pamk.result$pamobject$medoids
     Sepal.Length Sepal.Width Petal.Length Petal.Width
[1,]          5.0         3.4          1.5         0.2
[2,]          6.2         2.8          4.8         1.8
```

حال برای مقایسه‌ی خوشه‌های به‌دست آمده از این مدل و گروه‌بندی واقعی که توسط متغیر *species* تعیین شده است، می‌توانیم از تابع *table*() استفاده کنیم:

```
> table(pamk.result$pamobject$clustering,iris$Species)

    setosa versicolor virginica
  1     50          1         0
  2      0         49        50
```

همان‌طور که مشاهده می‌کنیم گونه‌های *verginica,versicolor* به یکدیگر شباهت دارند به‌طوری که در خوشه 2 قرار گرفته‌اند و گونه‌ی *setosa* با سایر گونه‌ها تفاوت دارد و در خوشه‌ی یک قرار گرفته است.

۴-۷-۲- روش‌های سلسله مراتبی

- خوشه‌بندی سلسله مراتبی، روشی است که در گروه‌بندی یا رده‌بندی داده‌ها به‌کار می‌رود. به عبارتی این روش، یک روش خوشه‌بندی می‌باشد که هدف آن ساخت یک سلسله مراتب از خوشه‌ها است. در روش خوشه‌بندی سلسله مراتبی، به خوشه‌های نهایی ساختاری سلسله مراتبی که به‌طور معمول به‌صورت درختی است،

نسبت داده می‌شود. به این درخت سلسله مراتبی درخت‌واره‌نگار می گویند. نتیجه-های یک خوشه‌بندی سلسله مراتبی عموماً به شکل یک درخت‌واره‌نگار نمایش داده می‌شوند.

روش‌های خوشه‌بندی سلسله مراتبی به دو دسته تقسیم می‌شوند: تجمعی و تجزیه‌ای.

- **تجمعی:** رویکرد این دسته پایین به بالا می‌باشد و با شروع از پایین، در هر مرحله دو خوشه که کم‌ترین تفاوت را دارند با یکدیگر تجمیع شده و یک خوشه جدید را شکل می‌دهند. خوشه‌های جدید در سطوح بالاتر قرار گرفته و این مرحله‌ها تا زمانی که تعداد خوشه‌ها به یک برسد تکرار می‌شود.
- **تجزیه‌ای:** رویکرد این دسته بالا به پایین می‌باشد و با شروع از بالا، در هر مرحله یک خوشه به خوشه‌های کوچک‌تری تجزیه می‌شود که در سطح پایین‌تر قرار می گیرند.

برای این که بدانیم کدام خوشه‌ها باید با هم تجمیع بشوند یا از یکدیگر تقسیم بشوند باید معیاری از تفاوت بین خوشه‌ها تعریف شود. در اکثر روش‌ها، این معیار به کمک تعریف یک تابع امتیاز و یک معیار پیوند حاصل می‌شود. تابع امتیاز فاصله‌ی بین دو تک مشاهده را تعیین کرده و معیار پیوند فاصله‌ی بین دو مجموعه مشاهده را توسط تابعی از فاصله دو به دو بین مشاهدات هر مجموعه تعریف می کند. انتخاب یک تابع امتیاز مناسب شکل خوشه‌ها را تحت تأثیر قرار می‌دهد، زیرا به‌ازای یک تابع امتیاز چند مشاهده می توانند به یکدیگر نزدیک باشند ولی به ازای تابع‌امتیاز دیگری فاصله‌ی آن‌ها از هم افزایش یابد. به‌عنوان مثال در یک فضای دو بُعدی فاصله‌ی بین نقاط (۰,۰) و (۱,۰) بنابر روش‌های معمول یک می‌باشد، اما فاصله‌ی بین همین دو نقطه با در نظر گرفتن فاصله منهتن مقدار 2، با در نظر گرفتن فاصله اقلیدسی جذر ۲ می‌باشد، و با در نظر گرفتن فاصله بیشینه مقدار 1 می‌باشد.

روش‌های خوشه‌بندی مبتنی بر سلسله مراتب به صورت زیر است:

- الگوریتم *Brich*

- الگوریتم *chameleon*
- الگوریتم *DIANA*
- الگوریتم *AGNES*

۴-۷-۲-۱- الگوریتم *Brich*

الگوریتم *Brich* با کنار هم قرار دادن خوشه‌بندی سلسله‌مراتبی و روش‌های دیگر خوشه‌بندی نظیر افراز تکرارشونده با هدف خوشه‌بندی مقدار بزرگی از داده‌های عددی، طراحی شده است.

در الگوریتم *Brich* برای تلخیص یک خوشه، از مفهوم ویژگی خوشه‌بندی و جهت نمایش سلسله‌مراتب خوشه از درختی موسوم به درخت ویژگی خوشه‌بندی استفاده می‌شود. با کمک این ساختارها، این روش سرعت مناسب و قابلیت مقیاس‌پذیری خوبی را در مواجه با دادگان‌های بزرگ و حتی دادگان‌های جریانی از خود نشان می‌دهد. خوشه‌ای از n شئ یا نقطه‌ی d بُعدی را درنظر بگیرید. ویژگی خوشه‌بندی این خوشه یک بردار سه‌بُعدی است که در آن اطلاعات خوشه‌ای نمونه‌ها خلاصه شده است. ویژگی خوشه‌بندی به‌صورت زیر معرفی می‌شود:

$$CF = (n, LS, SS)$$

که در آن LS مجموع خطی n نقطه (به عبارت دیگر $\sum_{i=1}^{n} x_i$) و SS مجموع توان‌دوم نقاط (یعنی $\sum_{i=1}^{n} x_i^2$) را نشان می‌دهند. ویژگی خوشه‌بندی خلاصه‌ای از آماره‌های یک خوشه را نشان می‌دهد. با کمک این ویژگی می‌توان به سادگی آماره‌های سودمندی درمورد یک خوشه را به‌دست آورد که این آماره‌ها و اطلاعات عبارت‌اند از: گرانیگاه خوشه، شعاع خوشه و قطر خوشه. تلخیص یک خوشه با کمک ویژگی خوشه‌بندی باعث می‌شود تا از ذخیره‌ی اطلاعات جزئی در مورد نمونه‌ها به‌صورت جداگانه جلوگیری شود. در عوض تنها به اندازه‌ی ثابتی از حافظه برای ذخیره‌ی این ویژگی‌ها نیاز است. کلید موفقیت الگوریتم *Brich* در مقدار حافظه‌ی مصرفی است. درخت ویژگی خوشه‌بندی یا همان *CF - tree*

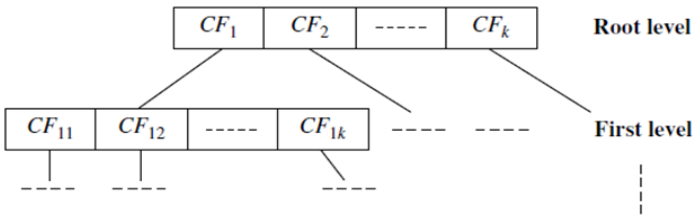
درخت متعادلی است که در خود، ویژگی‌های خوشه‌بندی را برای یک خوشه‌بندی سلسله‌مراتبی نگهداری می‌کند. هر $CF - tree$ دارای دو پارامتر با نام‌های فاکتور شاخه‌بندی B و حدآستانه‌ی T است. حداکثر تعداد فرزندان گره‌های غیرپایانی درخت، با فاکتور شاخه‌بندی B مشخص می‌شود و حدآستانه‌ی T حداکثر قطر (فاصله) خوشه‌های ذخیره شده در گره‌های پایانی یعنی برگ‌ها را تعیین می‌کند. با کمک این پارامترها می‌توان اندازه‌ی درخت حاصل را کنترل کرد.

یکی از مشخصات مهم و قابل توجه در الگوریتم $Brich$ کمینه‌کردن زمان لازم برای ورودی/خروجی است. الگوریتم $Brich$ از یک رویکرد خوشه‌بندی چندمرحله‌ای بهره می‌برد که در آن پس از پیمایش اول داده‌ها و خوشه‌بندی مناسب آن‌ها با پیمایش بعدی درصدد بهبود کیفیت و نتیجه‌های حاصل بر می‌آید. مرحله‌های ابتدایی این الگوریتم عبارت‌اند از:

1. الگوریتم با پیمایش دادگان‌ها، یک $CF - tree$ اولیه در حافظه می‌سازد. این درخت شکلی از فشرده‌سازی چندسطحی داده‌ها است که سعی می‌کند تا ساختار خوشه‌بندی ذاتی موجود در داده‌ها را حفظ کند.
2. الگوریتم با انتخاب یک روش خوشه‌بندی، گره‌های برگ $CF - tree$ را خوشه‌بندی می‌کند. سپس در این مرحله خوشه‌های خلوت شناسایی شده و حذف می‌شوند و گروه‌هایی از خوشه‌ها که به یکدیگر بسیار نزدیک‌تر هستند، با یکدیگر ادغام می‌شوند تا خوشه‌ی بزرگ‌تری ایجاد شود.

در مرحله‌ی اول همان‌طور که نمونه‌ها درج می‌شوند، درخت $CF - tree$ به‌صورت پویا ساخته می‌شود. بنا بر این، این روش از نوع افزایشی است. اندازه‌ی $CF - tree$ را می‌توان با کمک مقدار حد آستانه تغییر داد. اگر اندازه‌ی حافظه‌ی مورد نیاز برای نگهداری $CF - tree$ بیش‌تر از اندازه‌ی حافظه‌ی اصلی باشد، می‌توان مقدار بزرگ‌تری را برای حد آستانه در نظر گرفت و درخت را بازسازی کرد.

فرایند بازسازی، با ساخت یک درخت جدید از برگ‌های درخت قدیمی انجام می‌شود. بنا بر این برای اجرای این فرایند لازم است اطلاعات مربوط به نمونه‌ها یا نقاط دوباره خوانده شوند. لذا برای ساخت درخت، داده‌ها تنها یک بار خوانده می‌شوند. پیمایش‌های اضافی داده‌ها با استفاده از سلسله‌مراتب‌ها و روش‌های متعدد باعث می‌شود داده‌های دور افتاده شناخته شوند و کیفیت درخت‌های ویژگی خوشه‌بندی یعنی $CF - tree$ افزایش یابد. پیچیدگی الگوریتم برابر با $O(n)$ است که در آن n به تعداد نمونه‌هایی که باید خوشه‌بندی شوند، اشاره می‌کند. چون الگوریتم $Brich$ از مفهوم‌هایی چون شعاع و قطر برای کنترل کران‌های یک خوشه استفاده می‌کند، هنگامی که خوشه‌ها کروی شکل نباشند، این الگوریتم عملکرد مناسبی از خود نشان نمی‌دهد. مثالی از درخت ویژگی خوشه‌بندی یا همان $CF - tree$ در شکل ۴-۵ نشان داده شده است:

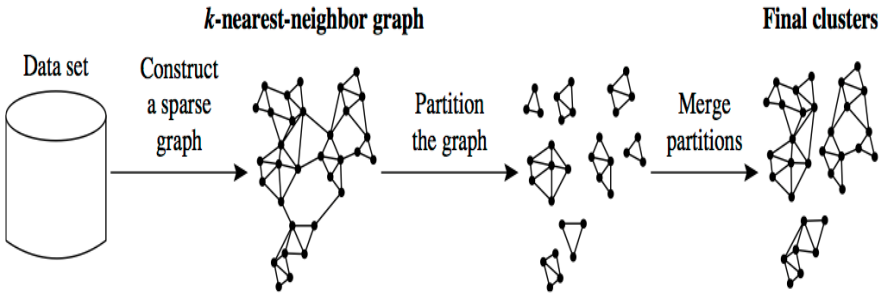


شکل۴-۵ : درخت ویژگی خوشه‌بندی

۴-۷-۲-۲- الگوریتم $chameleon$

الگوریتم $Chameleon$ یک الگوریتم خوشه‌بندی سلسله‌مراتبی است که با کمک مدل‌سازی پویا، شباهت میان زوج خوشه‌ها را تعیین می‌کند. در این الگوریتم شباهت خوشه بر اساس چگونگی اتصال مناسب نمونه‌ها درون خوشه و مجاورت و نزدیکی خوشه‌ها ارزیابی می‌شود. بنا بر این دو خوشه‌ای ادغام خواهند شد که دارای اتصالات قوی در درون خود و به یکدیگر بسیار نزدیک باشند. بنا بر این الگوریتم وابسته به یک مدل ایستا که توسط کاربر تهیه شده‌است، نیست و به‌صورت خودکار و بر اساس خصیصه‌های داخلی خوشه‌های

ادغام‌شونده عمل می‌کند. در شکل ۴-۶ چگونگی کار الگوریتم *Chameleon* نشان داده شده‌است:



شکل۴-۶: چگونگی کارکرد الگوریتم *Chameleon*

این الگوریتم از یک رویکرد خاص برای ساخت یک گراف استفاده می‌کند. رأس‌های این گراف، نمونه‌ها را نشان می‌دهند و وجود یک یال میان دو رأس نشان می‌دهد که یکی از نمونه‌ها نظیر رأس‌ها در میان k شئ مشابه به شئ دیگر قرار دارد. وزن‌های یال‌ها تشابه میان نمونه‌ها را نشان می‌دهند. الگوریتم *Chameleon* با کمک یک الگوریتم افراز گراف، گراف حاصل از مرحله‌ی قبل را به‌گونه‌ای به بخش‌های کوچک‌تر تقسیم می‌کند که برش یال‌ها کمینه باشند. بنا بر این خوشه‌ی C به‌نحوی به زیرخوشه‌های C_i و C_j تقسیم می‌شود که وزن یال‌های انتخابی برای برش کمینه باشند. الگوریتم *Chameleon* به‌هم پیوستگی مطلق میان خوشه‌های C_i و C_j را ارزیابی می‌کند و سپس این الگوریتم از یک روش خوشه‌بندی سلسله‌مراتبی تجمیعی استفاده می‌کند تا به‌صورت تکرار شونده زیرخوشه‌ها را بر اساس تشابه آن‌ها ادغام کند. جهت تعیین دو زیرخوشه‌ای که بیش‌ترین شباهت را با یکدیگر دارند، به‌هم پیوستگی و نزدیکی خوشه‌ها بررسی می‌شوند. الگوریتم *Chameleon* جهت تعیین تشابه میان خوشه‌های C_i و C_j از دو معیار به‌هم پیوستگی نسبی یعنی $RI(C_i,C_j)$ و نزدیکی نسبی یعنی $RC(C_i,C_j)$ استفاده می‌کند. الگوریتم *Chameleon* دارای توانایی بالاتری در کشف خوشه‌هایی با کیفیت مطلوب و شکل

دلخواه نسبت به الگوریتم‌های شناخته شده‌ای مانند *BIRCH* و الگوریتم چگالی مبنا *DBSCAN* می‌باشد.

۴-۷-۲-۳- الگوریتم *AGNES*

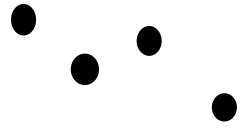
در الگوریتم *AGNES*، ابتدا هر کدام از نمونه‌ها در داخل یک خوشه قرار می گیرند. سپس خوشه‌ها مرحله به مرحله و بر اساس بعضی معیارها با یکدیگر ترکیب می‌شوند. اگر فاصله بین نمونه‌های هر خوشه با نمونه‌ها خوشه‌های دیگر را حساب کنیم و دو نمونه متعلق به خوشه‌ها، کم‌ترین فاصله را داشته باشند آن دو خوشه با یکدیگر ترکیب می‌شوند. ترکیب خوشه‌ها آن‌قدر ادامه می‌یابد تا همه نمونه‌ها درون یک خوشه قرار گیرند. در این روش تحلیل گر از قبل می‌تواند تعداد خوشه‌ها را انتخاب کند و از آن برای پایان شرط الگوریتم استفاده کند. معیارهای متفاوتی برای فاصله بین خوشه‌ها وجود دارد که پیوند تکی، پیوند کامل، پیوند متوسط و پیوند مرکزی از جمله این معیارها هستند. از جمله روش‌های مهم الگوریتم *AGNES* می‌توان به *Singe link* و *Complete link* و *Average link* اشاره کرد که به توضیح هر یک از آن‌ها در ادامه می‌پردازیم.

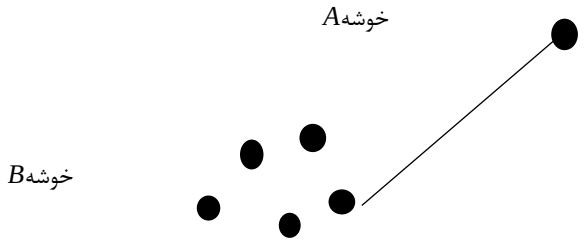
۴-۷-۲-۳-۱- خوشه‌بندی با روش $\sin gel - link$

این روش یکی از ساده‌ترین روش‌های خوشه‌بندی و جزء روش‌های خوشه‌بندی سلسله مراتبی محسوب می‌شود. به این روش خوشه‌بندی نزدیک‌ترین همسایه نیز گفته می‌شود. در این روش برای محاسبه شباهت بین دو خوشه A و B از معیار زیر استفاده می‌شود:

$$d_{AB} = \min d_{ij}, i \in A, \ j \in B$$

که i یک نمونه داده متعلق به خوشه A و j یک نمونه داده متعلق به خوشه B می‌باشد. در واقع در این روش بین دو خوشه، کم‌ترین فاصله بین یک عضو از یکی با یک عضو از دیگری است. در شکل ۴-۷ این مفهوم نشان داده شده‌است.





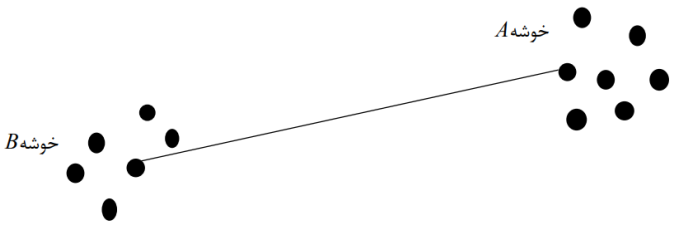
شکل۴-۷: شباهت بین دو خوشه در روش *single-link* برابر است با کم‌ترین فاصله بین داده‌های دو خوشه

۴-۷-۲-۳-۲- خوشه‌بندی با روش *complete- link*

این روش همانند *single-link* جزء روش‌های خوشه‌بندی سلسله مراتبی محسوب می‌شود. به این روش خوشه‌بندی دورترین همسایه نیز گفته می‌شود. در این روش برای محاسبه شباهت بین دو خوشه A و B از معیار زیر استفاده می‌شود:

$$d_{AB} = \max d_{ij},\ i \in A,\ j \in B$$

که i یک نمونه داده متعلق به خوشه A و j یک نمونه داده متعلق به خوشه B می‌باشد. در واقع در این روش شباهت بین دو خوشه بیش‌ترین فاصله بین یک عضو از یکی و یک عضو از دیگری است. در شکل ۴-۸ این مفهوم بهتر نشان داده شده است.



شکل۴-۸ : شباهت بین دو خوشه در روش *complete-link* برابر است با بیش‌ترین فاصله بین دو داده‌های دو خوشه

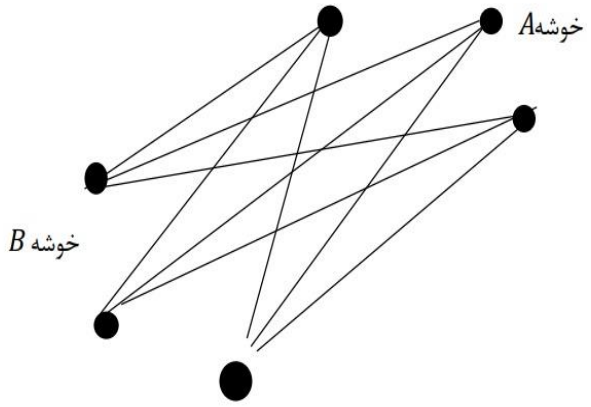
۴-۷-۲-۳-۳- خوشه‌بندی با روش *Average- link*

این روش همانند *single-link* جزء روش‌های خوشه‌بندی سلسله مراتبی است. از آن‌جا که هر دو روش *single-link* و *complete- link* به‌شدت به نوفه حساس می‌باشد، این

روش که محاسبات بیش‌تری دارد، پیش‌نهاد شد. در این روش برای محاسبه شباهت بین دو خوشه A و B از معیار زیر استفاده می‌شود:

$$d_{AB} = \frac{\sum_{i \in A, j \in B} d_{ij}}{N_A N_B}$$

تعداد اعضا خوشه A و N_B تعداد اعضاء خوشه B است. در واقع در این روش، شباهت بین دو خوشه میانگین فاصله بین تمام اعضا یکی با تمام اعضا دیگری است. در شکل ۴-۹ این مفهوم بهتر نشان داده شده است.



شکل۴-۹: شباهت بین دو خوشه در روش *Average-link* برابر است با میانگین فاصله بین داده‌های دو خوشه

مثال 4-6- فرض کنید 6 نمونه داده داشته باشیم و ماتریس فاصله بین آن‌ها در جدول ۴ -۱ محاسبه شده باشد. با روش *single-Link* نحوه اعمال خوشه‌بندی را نشان داده و درخت‌واره‌نگار مربوطه را رسم می‌کنیم.

جدول 4-1: ماتریس فاصله بین 6 نمونه داده

	1	2	3	4	5	6
1	0	4	13	24	12	8
2		0	10	22	11	10
3			0	7	3	9
4				0	6	18
5					0	8.5
6						0

در این روش ابتدا هر داده به‌عنوان یک خوشه در نظر گرفته می‌شود و یافتن نزدیک‌ترین خوشه در واقع یافتن کمینه فاصله بین داده‌های بالا خواهد بود. با توجه به جدول ۴ -۱ مشخص است که داده‌های 3 و 5 کم‌ترین فاصله را دارا هستند که در نتیجه دو خوشه 3 و 5 را با هم ترکیب می‌کنیم و خوشه جدیدی حاصل می‌شود که فاصله آن از سایر خوشه‌ها برابر است با کم‌ترین فاصله بین 3 و 5 از سایر خوشه‌ها. نتیجه در جدول 4-2 نشان داده شده است.

جدول4-2 : ماتریس فاصله بین 5 خوشه حاصل از تکرار اول

	1	2	(3و5)	4	6
1	0	4	12	24	8
2		0	10	22	10
(3و5)			0	6	8.5
4				0	18
6					0

با توجه به جدول 4-2 خوشه‌های 1 و2 کم‌ترین فاصله را دارا هستند پس آن‌ها را با هم ترکیب می‌کنیم و خوشه جدیدی حاصل می‌شود که فاصله آن از سایر خوشه‌ها برابر با کم‌ترین فاصله بین 1 و 2 از سایر خوشه‌ها است. نتیجه در جدول 4-3 نشان داده شده است.

جدول4-3 : ماتریس فاصله بین 4 خوشه حاصل از تکرار دوم

	(1و2)	(3و5)	4	6
(1و2)	0	10	22	8

(3و5)		0	6	8.5
4			0	18
6				0

با توجه به جدول 4-3 مشخص است که خوشه‌های (3و5) و 4 کم‌ترین فاصله را دارا هستند. در نتیجه آن‌ها را با هم ترکیب کرده و خوشه جدیدی حاصل می‌شود که فاصله آن از سایر خوشه‌ها برابر است با کم‌ترین فاصله بین خوشه (3و5) و 4 از سایر خوشه‌ها. نتیجه در جدول 4-4 نشان داده شده است.

جدول 4-4: ماتریس فاصله 3 خوشه حاصل از تکرار سوم

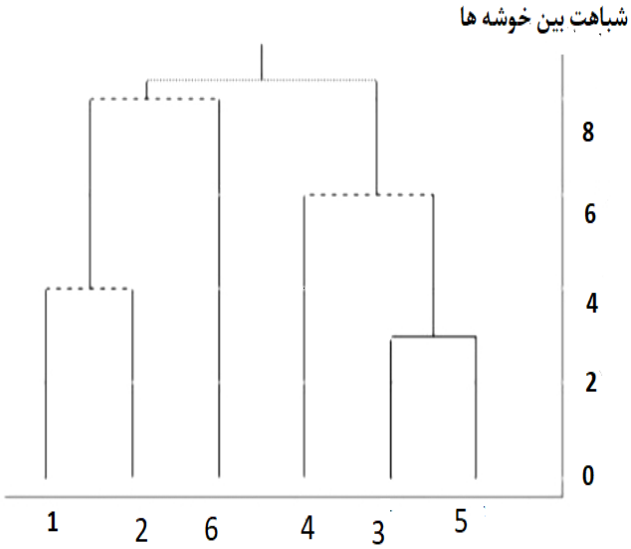
	(1و2)	(3و4و5)	6
(1و2)	0	10	8
(3و4و5)		0	8.5
6			0

با توجه به جدول 4-4 مشخص است که خوشه‌های (1و2) و 6 کم‌ترین فاصله را دارا هستند و در نتیجه آن‌ها را با هم ترکیب کرده و خوشه جدیدی حاصل می‌شود که فاصله آن از سایر خوشه‌ها برابر است با کم‌ترین فاصله بین (1و2) و 6 از سایر خوشه‌ها. نتیجه در جدول 4-5 نشان داده شده است.

جدول4-5: ماتریس فاصله بین 2 خوشه حاصل از تکرار چهارم

	(1و2و6)	(3و4و5)
(1و2و6)	0	8.5
(3و4و5)		0

در نهایت این خوشه‌ها ترکیب می‌شوند و درخت‌واره‌نگار مربوطه به صورت شکل 4-10 حاصل می‌شود.

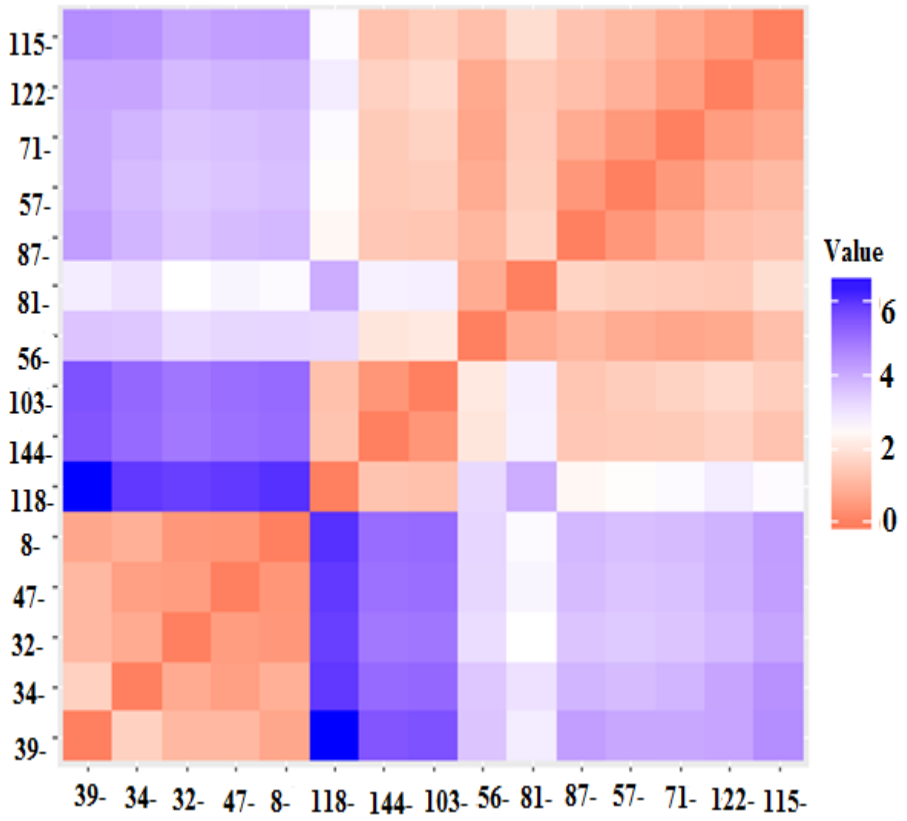


شکل4-10: درخت‌واره‌نگار روش *single-link*

مثال4-7- در این مثال به بررسی خوشه‌بندی سلسله مراتبی با داده‌های گل زنبق می-پردازیم که هدف ایجاد سه خوشه یعنی $k=3$ برای یک نمونه 150 تایی از داده‌ها می‌باشد. می‌دانیم داده‌های گل زنبق شامل سه نوع گل زنبق است پس به همین دلیل از هر دسته 50 تایی یک نمونه 5 تایی یعنی $n=5$ تهیه شده تا مشاهده نمودار و خروجی‌های خوشه‌بندی سلسله مراتبی بهتر دیده شود. در این مثال تابعی که برای اندازه‌گیری فاصله بین مشاهدات و یا خوشه‌ها در نظر گرفته شده، فاصله اقلیدسی است و معیار پیوند خوشه‌ها، روش میانگین است که محاسبات بر اساس آن صورت گرفته شده است. ماتریس فاصله با دستور *dist* تولید شده در متغیر *distance* و خروجی دستور *hclust* خوشه‌بندی سلسله مراتبی را می-دهد، به شکل4-11 و 4-12 توجه کنید: (کد5)

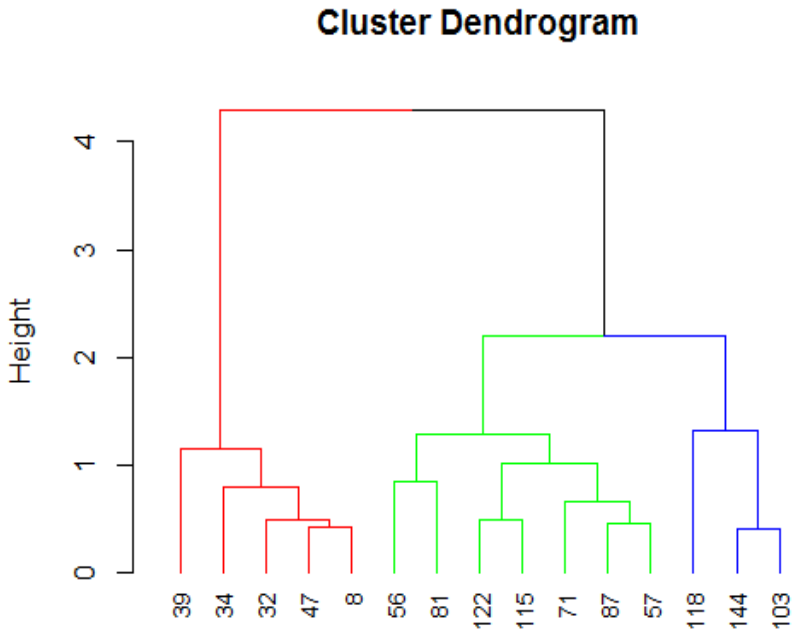
```
> library(stats)
> library(factoextra)
> n=5
> k=3
> data=iris[c(sample(x = 1:50,size = n),sample(x
=51:100,size=n),sample(x = 101:150,size= n)),1:4]
> distmethod=c('euclidean')
> linkagemethod=c("average")
> distance=dist(data,distmethod)
> hc=hclust(d = distance,method=linkagemethod)
> distance
29        41        20         2        35        98
54        94        75

41  0.2645751
20  0.4358899 0.3741657
2   0.5000000 0.5291503 0.8366600
35  0.4358899 0.4690416 0.7348469 0.1414214
98  3.2969683 3.4351128 3.2954514 3.3645208
3.2832910
54  3.0446675 3.1591138 3.1080541 2.9698485 2.9086079
0.9695360
94  2.3452079 2.4351591 2.4474477 2.1794495 2.1283797
1.7000000 0.9110434
75  3.3630343 3.5099858 3.3674916 3.4467376 3.3674916
0.2000000 1.1224972 1.8466185
74  3.6138622 3.7509999 3.6124784 3.6565011 3.5735137
0.4358899 1.0535654 1.8601075 0.5196152
115 4.3874822 4.4698993 4.3428102 4.4022721 4.3243497
1.4212670 1.6613248 2.4677925 1.4899664
119 6.4459289 6.5924199 6.4311741 6.5314623 6.4544558
3.1780497 3.7868192 4.6936127 3.0886890
133 4.8414874 4.9547957 4.8072861 4.8918299 4.8114447
1.5968719 2.1047565 2.9899833 1.5842980
117 4.6065171 4.7318073 4.5661800 4.6829478 4.5967380
1.3379088 1.9974984 2.8670542 1.3076697
142 4.5912961 4.7127487 4.5486262 4.7021272 4.6227697
1.4730920 2.1931712 3.0298515 1.3892444
74       115       119       133       117
41
20
2
35
98
54
94
```

```
75
74
115 1.3000000
119 2.9410882 2.6267851
133 1.3784049 0.8062258 1.8520259
117 1.0954451 1.0246951 1.9519221 0.4690416
142 1.4491377 1.1445523 2.0322401 0.7745967 0.7615773

> fviz_dist(dist.obj = distance)
> hc

Call:
hclust(d = distance, method = linkagemethod)

Cluster method   : average
Distance         : euclidean
Number of objects: 15
> fviz_dend(x = hc,k = k)
```



شکل4-11 : نمایش تصویری ماتریس فاصله

شکل 4-11 نمایشی از ماتریس فاصله را نشان می‌دهد که در آن هر شئ با خودش کم‌ترین فاصله را دارد و با رنگ خاکستری نشان داده شده است و نمونه‌هایی که بیش‌ترین فاصله را با هم دارند با رنگ خاکستری نشان داده شده است.



شکل 4-12 نمودار سلسله مراتبی داده‌های گل زنبق

۴-۷-۲-۴- مقایسه خوشه‌بندی سلسله مراتبی و غیر سلسله مراتبی

روش‌های خوشه‌بندی غیر سلسله مراتبی نیاز به تصمیم‌گیری‌هایی از سوی کاربر دارد به‌طور مثال انتخاب تعداد خوشه‌ها. در این روش به‌طور معمول خوشه‌های اولیه ایجاد می‌شود و سپس در ادامه بهبود می‌یابد. با توجه به این نکته که تعداد خوشه‌های اولیه و انتخاب آن مهم است و دارای حساسیت می‌باشد، گاهی اوقات به کمک روش سلسله مراتبی، تعداد خوشه‌ها براورد می‌شود. ازجمله ایرادات روش سلسله مراتبی این است که در یک مرحله قابلیت تغییر در مرحله‌های بعد را ندارد که این موجب تصمیمات نادرست می‌شود.

۴-۷-۳- روش‌های مبتنی بر چگال

در تعیین تعداد خوشه‌ها استقلال یا وابستگی کامل داده‌ها نقش مهمی دارد. اگر تمام متغیرها وابسته باشند، آن‌گاه تمام داده‌ها تشکیل یک خوشه می‌دهند، یا به عکس اگر تمام متغیرها به‌طور کامل مستقل باشند، هیچ خوشه‌ای ایجاد نمی‌شود (تمام فضا به‌صورت تصادفی با نقاط داده پر می‌شود). در شرایط استقلال یا وابستگی کامل نمی‌دانیم که واقعاً چند خوشه وجود دارد. به‌طور معمول در انتخاب تعداد خوشه‌ها نقش تحلیل‌گر بیش‌تر از رایانه است، با توجه به کاربردهای متفاوت خوشه‌بندی، تعداد خوشه‌ها ممکن است بیش‌تر یا کم‌تر باشد. در بسیاری از مواقع با یک مقدار k خوشه‌بندی را انجام داده و نتیجه‌ها را بررسی می‌کنند و دوباره یک مقدار k دیگر را امتحان می‌کنند. بعد از هر تکرار، ارزش نتیجه‌ها به‌وسیله میزان متوسط فاصله‌ها در داخل خوشه‌ها و یا میزان متوسط فاصله‌ها بین مراکز خوشه‌ها، اندازه‌گیری و با یکدیگر مقایسه می‌شوند. این موضوع را هم باید مد نظر قرار داد که گاهی خوشه‌ها به‌وسیله قضاوت‌های ذهنی تحلیل‌گر مورد ارزیابی قرار می‌گیرد تا ارزش آن‌ها در کاربردهای خاصی مشخص شوند. معیارهایی برای ارزیابی رده‌های تشکیل شده و همچنین تعیین k، مناسب وجود دارد. اکثر روش‌های افراز بر اساس فاصله‌ی میان نمونه آن‌ها را خوشه‌بندی می‌کنند. این روش‌ها می‌توانند تنها خوشه‌های کروی شکل را پیدا کنند و برای کشف خوشه‌هایی با شکل دلخواه با مشکل مواجه می‌شوند. ایده کلی در روش‌های مبتنی بر چگال به این‌صورت است که مادامی‌که چگالی (تعداد نمونه یا داده‌ها) در همسایگی یک خوشه، از مقدارهایی بیش‌تر باشد، خوشه به رشد خود ادامه می‌دهد. چنین روشی را می‌توان برای یافتن داده‌های دور افتاده یا کشف خوشه‌هایی با شکل دلخواه استفاده کرد. روش‌های چگالی مبنا می‌توانند مجموعه‌ای از نمونه را در چندین خوشه‌ی انحصاری یا یک سلسله‌مراتبی از خوشه‌ها تقسیم کنند. به‌طور معمول در این روش‌ها خوشه‌ها انحصاری هستند، به این معنی که هر شئ تنها به یک خوشه تعلق دارد. همچنین با تعمیم این روش‌ها می‌توان از آن‌ها برای خوشه‌بندی زیرفضا نیز استفاده کرد. یکی از رایج‌ترین الگوریتم‌های

خوشه‌بندی مبتنی بر چگال الگوریتم *DBSCAN* است. همچنین می‌توان الگوریتم *optics* را به‌عنوان یک روش خوشه‌بندی مبتنی بر چگال اشاره کرد.

شکل ۴-۱۳: یک مثال برای خوشه‌های چگالی مبنا

۴-۷-۳-۱- الگوریتم *DBSCAN*

یکی از معروف‌ترین الگوریتم‌ها در حوزه خوشه‌بندی الگوریتم *DBSCAN* است. در این الگوریتم نیاز به تعیین تعداد خوشه‌ها توسط کاربر نیست و خود الگوریتم می‌تواند خوشه‌ها را مبتنی بر غلظت آن‌ها شناسایی کند، یا داده‌ها را بر اساس تراکم آن‌ها خوشه‌بندی کند. الگوریتم *DBSCAN* به دو پارامتر ورودی نیاز دارد: شعاع هر خوشه و حداقل نقطه در درون هر خوشه (جانسون،2014). در ابتدا یک نمونه انتخاب می‌شود و با توجه به شعاع، به‌دنبال همسایه برای این نقطه در فضا (مجموعه‌داده‌ها) می‌گردد. اگر در شعاع مشخص، الگوریتم بتواند حداقل نقاط موجود در داده‌ها را پیدا کند، آن‌گاه همه‌ی آن داده‌ها با هم به یک خوشه تعلق می‌گیرند. سپس به‌دنبال نقاط هم‌جوار نقطه فعلی می‌رود تا با شعاع *Epsilon* در آن نقطه به‌دنبال همسایه دیگر بگردد، و اگر تعداد نقاط همسایه جدید باز هم پیدا نشوند، این الگوریتم دوباره همه‌ی آن نقاط جدید را با نقاط قبلی به یک خوشه متعلق می‌کند و اگر نقطه‌ی جدیدی پیدا نشود این خوشه تمام شده است و برای پیدا کردن خوشه‌های دیگر در نقاط دیگر، به‌صورت تصادفی یک نقطه دیگر را انتخاب می‌کند و شروع به یافتن همسایه و تشکیل خوشه جدید برای آن نقطه می‌کند. این کار آن‌قدر ادامه می‌یابد تا تمامی نقاط بررسی شوند.

حال با توجه به توضیحات بالا می‌توان نتیجه گرفت که یک خوشه چگالی مینا، مجموعه‌ای از نقاط متصل به‌یکدیگر از نظر چگالی است که هر داده را که خارج از این خوشه‌ها باشند، به‌عنوان داده دور افتاده در نظر می‌گیرد. یکی از مشکلات الگوریتم *DBSCAN* مشخص نبودن تعداد *Epsilon* و حداقل نقاط است که باید توسط کاربر مشخص شود.

۴-۷-۳-۱-۱- مزیت‌ها و عیب‌های الگوریتم *DBSCAN*

مزیت‌ها:

- عدم محدودیت به شکل خوشه‌ها
- سریع برای داده‌های با بُعد کم
- یافتن خوشه‌ها برای اشکال نامنظم
- تشخیص نوفه

عیب‌ها:

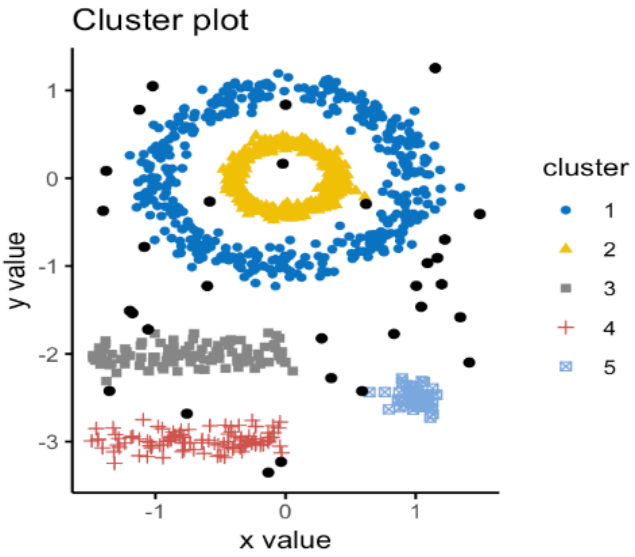
- نقاط مرزی که می‌توانند در دو خوشه نیز باشند، ممکن است به هر یک از خوشه‌ها تعلق گیرند
- مشکل بودن تعیین مقدار دقیق پارامترهای ورودی
- عدم تشخیص خوشه‌های با چگالی متفاوت

مثال۴-۸- مجموعه‌داده‌های *Multishapes* مجموعه‌ای در دادگان‌های نرم‌افزار *R* است که مناسب برای روش‌های خوشه‌بندی چگالی مبنا می‌باشد، این مجموعه‌داده با ۱۱۰۰ مشاهده می‌باشد که دارای سه متغیر *x*، *y* و *shape* (یک بردار عددی مربوط به شماره خوشه‌ها) است. برای پیاده‌سازی الگوریتم *DBSCAN* روی این مجموعه‌داده‌ها به‌صورت زیر عمل می‌شود: (کد۶)

```
> data("multishapes",packege="factoextra")
> df=multishapes[1:2,]
> library("fpc")
> set.seed(123)
> db=fpc::dbscan(df,eps=0.15,minpts=5)
> print(db)
```

```
> library("factoextra")
> fviz_cluster(db,data=df,stand=FALSE,ellipse=FALSE,
show.clust.cent=FALSE
> geom="point",palette="jco",ggtheme=theme_classic())
```

خروجی دستورهای بالا شکل ۴-۱۴ خواهد بود:



شکل۴-۱۴ : نشان می‌دهد که این مجموعه به ۵ خوشه تقسیم شده

در این شکل نقاط دایره‌ای شکل مربوط به داده‌های نوفه می‌باشد. برای تغییر پیکربندی خوشه‌ها نیز می‌توان مقدارهای *eps* و *MinPts* را در تابع *fpc::dbscan()* تغییر داد.

۴-۷-۳-۲- الگوریتم *OPTICS*

اگرچه الگوریتم *DBSCAN* با کمک پارامترهای *Epsilon* (حداکثر شعاع یک همسایه) و حداقل نقاط لازم در همسایگی یک شئ قادر است نمونه را خوشه‌بندی کند، اما کاربرها می‌بایست بهترین مقدارها را برای این پارامترها مشخص کنند تا الگوریتم خوشه‌های قابل قبولی را استخراج کند. این مشکل در بسیاری از الگوریتم‌های خوشه‌بندی دیگر نیز وجود دارد. به‌طور معمول تنظیم چنین پارامترهایی به‌صورت تجربی انجام می‌شود و تعیین آن‌ها

به‌خصوص برای داده‌های دنیای واقعی با تعداد ابعاد بالا، بسیار دشوار است. اکثر الگوریتم‌ها به مقدارهای این پارامترها حساس هستند، به‌نحوی که تغییر ناچیز آن‌ها ممکن است نتیجه‌های بسیار متفاوتی در خوشه‌بندی را به همراه داشته باشد. اغلب در دنیای واقعی، داده‌ها دارای ابعاد بالایی هستند و همچنین توزیع‌های آن‌ها با چولگی زیاد است. بنا بر این ساختار طبیعی خوشه‌های موجود در این گونه داده‌ها را نمی‌توان تنها با کمک چند پارامتر چگالی سراسری توصیف نمود.

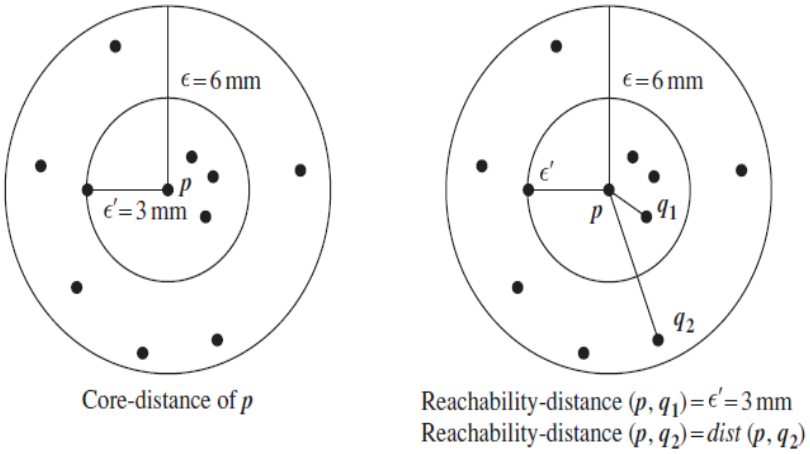
جهت غلبه بر مشکلاتی که استفاده از پارامترهای سراسری در تحلیل خوشه ایجاد می‌کنند، یک روش تحلیل خوشه به نام $OPTICS$ پیش‌نهاد شد. این الگوریتم به‌صورت آشکار عمل خوشه‌بندی داده‌ها را انجام نمی‌دهد و در عوض یک ترتیب خوشه را در خروجی خود قرار می‌دهد. این خروجی یک فهرست خطی از کلیه‌ی نمونه تحت بررسی و تحلیل است و ساختار خوشه‌بندی چگالی مبنا داده‌ها را نمایش می‌دهد. در ترتیب خوشه، نمونه‌هایی که در یک خوشه‌ی متراکم‌تر هستند در این فهرست به یکدیگر نزدیک‌تر خواهند بود. این نظم معادل خوشه‌بندی چگالی مبنا است که با کمک طیف وسیعی از تنظیمات پارامترها به‌دست آمده است. بنا بر این در $OPTICS$ لازم نیست کاربر حد آستانه‌ی چگالی مشخصی را تعیین کند. ترتیب خوشه را می‌توان برای استخراج اطلاعات پایه‌ای خوشه‌بندی (برای مثال مراکز خوشه‌ها یا خوشه‌هایی با شکل‌های دلخواه)، کشف ساختار طبیعی و موجود خوشه‌ها و همچنین مصورسازی خوشه‌بندی استفاده کرد.

برای ساخت خوشه‌بندی‌های مختلف به‌طور هم‌زمان، نمونه به یک نظم خاصی پردازش می‌شوند. در این نظم شیٔ‌ای که در رابطه با کم‌ترین مقدار ε قابل دسترس است، انتخاب می‌شود. بدین ترتیب خوشه‌هایی با تراکم بالاتر (ε کوچک‌تر) زودتر به پایان خواهند رسید. بر اساس همین ایده الگوریتم $OPTICS$ برای هر شیٔ به دو فقره اطلاعات مهم نیاز دارد:

1. فاصله‌ی هسته‌ی یک شیٔ p به کوچک‌ترین مقداری از ε' اطلاق می‌شود که در شعاع همسایگی ε' از شیٔ p بتوان حداقل تعداد $MinPts$ شیٔ پیدا کرد.

2. فاصله‌ی قابل دسترس به شئ p از q به مقدار حداقل شعاعی اطلاق می‌شود که باعث می‌شود p از q قابل دسترس باشد. طبق تعریف قابلیت دسترسی، شئ q باید یک شئ هسته باشد و p نیز باید در همسایگی q قرار گرفته باشد. چنانچه با توجه به مقدارهای ε و $MinPts$ شئ q یک شئ هسته نباشد، فاصله‌ی قابل دسترس شئ p از q نیز تعریف نشده‌است.

ممکن است شئ‌ای مانند p به‌طور مستقیم از چندین شئ هسته قابل دسترس باشد. بدین ترتیب شئ p باتوجه به نمونه هسته‌ی مختلف دارای چندین فاصله‌ی قابل دسترس است. از میان آن‌ها کوچک‌ترین مقدار برای ما اهمیت دارد، چون با کمک این مقدار، کوتاه‌ترین مسیری که شئ p به یک خوشه‌ی متراکم متصل می‌شود، مشخص می‌گردد. شکل۴- ۱۵ دو مفهوم فاصله‌ی هسته و فاصله‌ی قابل دسترس را به‌صورت تصویری نشان می‌دهد:



شکل۴- ۱۵ : فاصله‌ی هسته و فاصله قابل دسترس

الگوریتم *OPTICS* نظم و ترتیب را برای کلیه‌ی نمونه موجود در دادگان‌ها محاسبه می‌کند و مقدارهای فاصله‌ی هسته و یک فاصله‌ی قابل دسترس مناسبی برای هر یک از نمونه را نیز ذخیره می‌کند. در این الگوریتم برای تولید نظم خروجی، از فهرستی به نام

OrderSeeds استفاده می‌شود. نمونه در این فهرست بر اساس فاصله‌ی قابل دسترس از نزدیک‌ترین نمونه هسته‌ی مربوط به خود، به‌صورت صعودی مرتب شده‌اند.
الگوریتم *OPTICS* کار خود را با انتخاب یک شئ مانند p به‌صورت اختیاری از دادگان‌ها آغاز می‌کند. همسایگان p (به شعاع ε) بازیابی می‌شوند، فاصله‌ی هسته نیز برای p تعیین می‌گردد. از آن‌جا که فاصله‌ی قابل دسترس این شئ یعنی p مشخص نیست، مقدار فاصله‌ی قابل دسترس آن با برچسب تعریف نشده علامت‌گذاری می‌شود. شئ p به‌عنوان یک شئ پردازش‌شده در خروجی نوشته می‌شود. اگر شئ p یک شئ هسته نباشد، الگوریتم به سراغ شئ بعدی در فهرست *OrderSeeds* می‌رود و یا اگر فهرست *OrderSeeds* خالی باشد، از دادگان شئ‌ای انتخاب می‌شود. نظم به‌دست آمده از الگوریتم را می‌توان به‌صورت گرافیکی نمایش داد. این کار به مصورسازی و فهم ساختار خوشه‌بندی موجود در مجموعه‌داده‌ها کمک می‌کند.

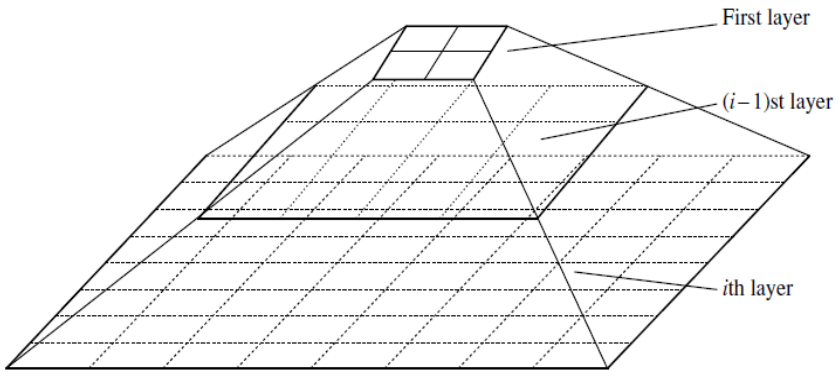
۴-۷-۴- روش مشبکی‌مبنا

روش‌های خوشه‌بندی که تا این‌جا بحث شد، داده محور هستند. آن‌ها با توجه به توزیع داده‌ها مجموعه‌ای از نمونه را بخش‌بندی می‌کنند. یک روش خوشه‌بندی مشبکه‌ای از یک رویکرد فضا محور بهره می‌گیرد و این کار را با افراز فضای تعبیه شده به سلول‌ها و مستقل از توزیع نمونه ورودی انجام می‌دهد. روش مشبکی‌مبنا به سلول‌های مختلف، امکان کار بر روی اطلاعات با درجه وضوح و شفافیت‌های متفاوت را فراهم می‌کند. در این رویکرد ابتدا فضا به سلول‌هایی تقسیم شده و سپس کلیه عملیات خوشه‌بندی بر روی این فضا انجام می‌گیرد. یکی از مزیت‌های این روش سرعت بالای پردازش آن است، زیرا پیچیدگی وابسته به تعداد سلول‌ها است. از دیگر مزیت‌های این روش کارا بودن است. ابتدایی‌ترین روش در این دسته روش شبکه اطلاعات آماری یا *STING* است.

۴-۷-۴-۱- الگوریتم *STING*

روش *STING* یکی از روش‌های مشبکی‌مبنا است که در این روش فضا به سلول‌های ابتدایی تقسیم می‌شود، این تقسیم‌بندی می‌تواند به روش سلسله مراتبی یا بازگشتی باشد. با افراز هر سلول در سطوح بالا، تعداد سلول در سطوح پایین شکل می‌گیرد. به‌طور مثال از ترکیب هر 4 سلول یک سلول در لایه‌ای بالاتر با درجه تفکیک کم‌تر شکل می‌گیرد. سپس برای هر سلول پارامترهایی مانند میانگین، میانه، کمینه و بیشینه محاسبه و به‌عنوان پارامترهای آماری ذخیره می‌شوند.

شکل 4-16 یک ساختار سلسله‌مراتبی را برای خوشه‌بندی *STING* نشان می‌دهد. پارامترهای آماری سلول‌های سطوح بالاتر را می‌توان با کمک پارامترهای سلول‌های سطوح پایین‌تر محاسبه کرد. این پارامترها می‌توانند گونه‌های مختلفی باشند: پارامتر مستقل از صفت‌خاصه مانند شمارش مقدارها، پارامترهای وابسته به صفت‌خاصه نظیر میانگین، انحراف استاندارد، بیشینه و کمینه و گونه‌ی سوم نوع توزیعی است که مقدارهای صفت‌خاصه‌ی موجود در سلول از آن پیروی می‌کنند مانند توزیع نرمال، یکنواخت، نمایی و یا حتی توزیع ناشناخته (هان و کامبر،2006).



شکل 4-16: خوشه‌بندی به روش *STING*

در این‌جا پارامتر محاسبه‌شده، یک معیار منتخب برای تحلیل است و هنگام بارگذاری داده‌ها درون دادگان‌ها، پارامترهای شمارش، میانگین، انحراف استاندارد، بیشینه و کمینه سلول‌های

سطح پایین را می‌توان به‌طور مستقیم از روی داده‌ها محاسبه نمود. چنانچه از قبل نوع توزیع مشخص باشد، کاربر آن‌را برای سلول معین می‌کند و یا درصورت عدم آگاهی کاربر از توزیع، می‌توان آن‌را با کمک آزمون‌هایی نظیر آزمون خی‌دو به‌دست آورد. نوع توزیع سلول‌های سطوح بالاتر با رأی اکثریت نوع توزیع در سلول‌های متناظر سطوح پایین‌تر محاسبه می‌شوند. برای این کار می‌توان از حدآستانه‌ای نیز استفاده کرد. چنانچه توزیع‌های سلول‌های سطح پایین‌تر بر اساس انتخاب یک توزیع مناسب برای سلول سطح بالاتربه توافق نرسند و آزمون حدآستانه نیز این مشکل را برطرف نسازد، آن‌گاه نوع توزیع سطح بالا با برچسب ناشناخته تنظیم و علامت‌گذاری می‌شود.

چگونه این اطلاعات آماری برای پاسخ به پرسش سودمند است؟ از پارامترهای آماری می‌توان در یک روش مشبکی‌مبنا و یک روش بالا به پایین استفاده کرد. در ابتدا لایه‌ای از ساختار سلسله‌مراتب تعیین می‌شود، لایه‌ای که فرایند پاسخ به پرسش از آن آغاز می‌شود. به‌طور معمول این لایه شامل تعداد کمی از سلول‌ها است و برای هریک از سلول‌های سطح جاری، فاصله اطمینان محاسبه می‌شود تا ارتباط هر یک از آن‌ها با پرسش مورد نظر مشخص شود. بر روی سلول‌های بی‌ربط بررسی‌های بیش‌تری انجام نمی‌شود و آن‌ها از این فرایند حذف می‌شوند. جهت پردازش لایه‌ی بعدی تنها سلول‌های مرتبط و باقی‌مانده در نظر گرفته می‌شوند. این فرایند تا سطح پایین ساختار ادامه می‌یابد. چنانچه در این زمان مشخصات پرسش دیده شود، مناطق از سلول‌های مرتبط که پرسش را در بر می‌گیرد، برگردانده می‌شوند. در غیر این‌صورت داده‌هایی که درون سلول‌های مرتبط هستند، بازیابی می‌شوند و در ادامه تا رسیدن به نیازمندی‌های پرسش پردازش‌های بیش‌تر انجام می‌شود.

یکی از خصیصه‌های جالب توجه روش *STING* این است که چنانچه خوشه‌بندی به صفر نزدیک شود (به عبارت دیگر به سمت داده‌های سطح پایین حرکت کند)، نتیجه‌های خوشه‌بندی در این الگوریتم به نتیجه‌های حاصل از الگوریتم *DBSCAN* نزدیک می‌شود. به عبارت دیگر با کمک الگوریتم *STING* و همچنین استفاده از اطلاعات مربوط به

اندازه‌ی سلول و شمارش آن‌ها می‌توان خوشه‌های متراکم را شناسایی نمود. بدین ترتیب می‌توان الگوریتم *STING* را یک روش خوشه‌بندی چگالی مبنا نیز تصور نمود.

از آن‌جا که الگوریتم *STING* از یک رویکرد با چندین سطح وضوح و درستی جهت تحلیل خوشه استفاده می‌کند، کیفیت خوشه‌بندی در آن وابسته به دانه‌بندی پایین‌ترین سطح از ساختار شبکه می‌باشد. چنانچه دانه‌بندی بسیار ریز انتخاب شود، هزینه‌ی پردازش به‌صورت قابل ملاحظه‌ای افزایش خواهد یافت و از طرفی اگر سطح پایین ساختار شبکه از دانه‌بندی درشت استفاده کند، کیفیت تحلیل خوشه کاهش خواهد یافت.

مثال4-9: در این مثال به بررسی تعداد بهینه‌ی خوشه‌ها برای دادگان گل زنبق می‌پردازیم. از تابع *fviz_nbclust()* در بسته نرم‌افزاری *factoextra* و تابع *NbClust()* در بسته نرم‌افزاری *NbClust* برای تعیین تعداد بهینه‌ی خوشه‌ها می‌توان استفاده کرد. در این مثال به بررسی تعداد بهینه‌ی خوشه‌ها با استفاده از تابع *NbClust()* می‌پردازیم. (کد7)

تابع *NbClust()* در بسته‌ی نرم‌افزاری *NbClust*، 26 شاخص متفاوت برای تعیین بهترین تعداد خوشه ارائه می‌کند که همگی آن‌ها لزوماً با یکدیگر سازگار نمی‌باشند، اما می‌توان از نتایج این تابع به‌عنوان راهنمایی برای انتخاب مقدارهای ممکن k (تعداد خوشه) استفاده کرد.

تابع *fviz_nbclust()* در بسته نرم‌افزاری *factoextra* : از این تابع می‌توان در محاسبه‌ی متدهای *elbow* و *silhouette* استفاده کرد. همچنین این تابع قابلیت به‌کارگیری در انواع مختلف تابع‌های خوشه‌بندی مانند *k-means*، *k-medoids* و *CLARA* را دارد.

تابع *NbClust()* در بسته نرم‌افزاری *NbClust* : ورودی‌های تابع *NbClust()*، دادگان، معیار فاصله، روش خوشه‌بندی، کم‌ترین تعداد خوشه و بیش‌ترین تعداد خوشه است. این تابع تعداد شاخص‌های هر خوشه همراه با بهترین تعداد خوشه‌ی پیش‌نهادی را به‌عنوان خروجی نمایش می‌دهد.

کاربر با تغییر متغیرهایی چون تعداد خوشه‌ها، معیار محاسبه فاصله بین داده‌ها و روش خوشه-بندی مورد استفاده، بهترین الگوی خوشه‌بندی را به‌عنوان خروجی تابع خواهد داد. تنها با

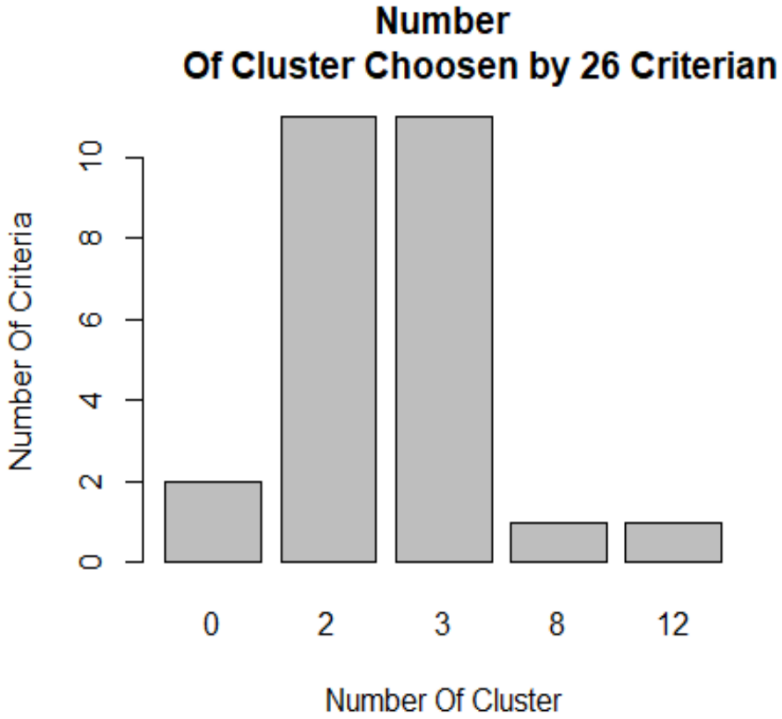
یکبار اجرای این تابع، مقدارهای بهینه‌ی این شاخص‌ها محاسبه شده و تعداد بهینه خوشه‌ها تعیین می‌شود.

```
> data(iris)
> iris2=iris
> iris2$Species=NULL
> library(NbClust)
> nc=NbClust(iris2,min.nc=2,max.nc=15,method=
"kmeans")

*** : The Hubert index is a graphical method
of determining the number of clusters.
                In the plot of Hubert index,
we seek a significant knee that corresponds t
o a
                significant increase of the v
alue of the measure i.e the significant peak
in Hubert
        index second differences plot.
*** : The D index is a graphical method of de
termining the number of clusters.
In the plot of D index, we seek a significant
knee (the significant peak in Dindex second d
ifferences plot) that corresponds to a signif
icant increase of the value of the measure.
**********************************************
*********
* Among all indices:
* 11 proposed 2 as the best number of cluster
s
* 11 proposed 3 as the best number of cluster
s
* 1 proposed 8 as the best number of clusters
* 1 proposed 12 as the best number of cluster
s
            ***** Conclusion *****
* According to the majority rule,the best num
ber of clusters is 2
**********************************************
*********
> BAR=table(nc$Best.nc[1,])
> BAR
```

```
 0  2  3  8 12
 2 11 11  1  1
> barplot(BAR,xlab="Number Of Cluster",ylab="
Number Of Criteria", main = "Number
+Of Cluster Choosen by 26 Criterian")
```

که همان‌طور که پیدا است تعداد خوشه بهینه 2 و 3 در نظر گرفته شده است. یعنی اگر $k = ۲$ در نظر گرفته شود تعداد 11 شاخص تأیید می‌شود و اگر $k = ۸$ در نظر گرفته شود تعداد 1 شاخص تأیید می‌شود.

نمودار میله‌ای تعداد خوشه‌ها در مقابل تعداد شاخص‌ها به‌صورت شکل است.



شکل4-17: نمودار میله‌ای تعداد خوشه‌ها در مقابل تعداد شاخص‌ها

۶-۳-۵- الگوریتم $K2$

الگوریتم $K2$ یکی از روش‌های مبتنی‌بر جست‌وجو و رتبه‌بندی است که از جست‌وجو حریصانه استفاده می‌کند. ایده اصلی الگوریتم $K2$ بیشینه کردن احتمال ساختار بر اساس داده های موجود است. الگوریتم $K2$ احتمالاً یکی از بهترین الگوریتم‌هایی است که تاکنون برای یادگیری ساختار شبکه‌های بیزی ابداع شده است. این الگوریتم از یک معیار امتیازدهی بیزی استفاده می‌کند. هدف اصلی این الگوریتم جست‌وجوی پایگاه‌داده D و بیشینه کردن $P(B_s, D)$ برای دست‌یابی به بهینهٔ B_S^* (شبکهٔ بیزی) است.

الگوریتم کار خود را با فرض این‌که تمام گره‌ها فاقد والدین هستند، شروع می‌کند؛ سپس در هر مرحله به‌تدریج برای هر گره، متغیرهایی که منجر به بیش‌ترین افزایش در احتمال ساختار نهایی می‌شوند، به‌عنوان والدین گره انتخاب می‌کند (اسکندری و همکاران، 2018).

- **ورودی و خروجی الگوریتم**

ورودی: مجموعه‌ای از n گره یا متغیر تصادفی گسسته، یک ترتیب روی گره‌ها، یک حد بالای u روی تعداد والدین هر گره و یک دادگان شامل m نمونهٔ مستقل .

خروجی: مجموعهٔ والدین مربوط به هر گره.

به‌طور کلی مرحله‌های اجرای الگوریتم به‌صورت زیر است:

شروع الگوریتم K2

برای i از یک تا n انجام شود:

مجموعهٔ والدین متغیر i را صفر در نظر بگیرید، یعنی: $Pa(x_i) = \cdot$

$P_{old} = g(x_i, Pa(x_i))$ (تابع g تابع امتیاز K۲ شبکهٔ بیزی است)

تا زمانی که متغیری برای ادامه‌دادن وجود دارد و تعداد والدین گره x_i از u کمتر است، ادامه بدهید.

شروع

تا زمانی که گره z از مجموعه متغیرهایی که x_i می‌تواند به‌عنوان والدین خود انتخاب نماید و هنوز انتخاب نشده است، وجود دارد، آن را طوری انتخاب کنید که عبارت $g(x_i, Pa(x_i) \cup \{z\})$ را ماکسیمم کند.

$$P_{New} = g(x_i, Pa(x_i) \cup \{z\})$$

اگر $P_{New} > P_{old}$

شروع

$$P_{old} := P_{New}$$
$$Pa(x_i) = Pa(x_i) \cup \{z\}$$

پایان

اگر متغیری که به‌عنوان والدین متغیر x_i انتخاب شود، در ترتیب وجود نداشت،

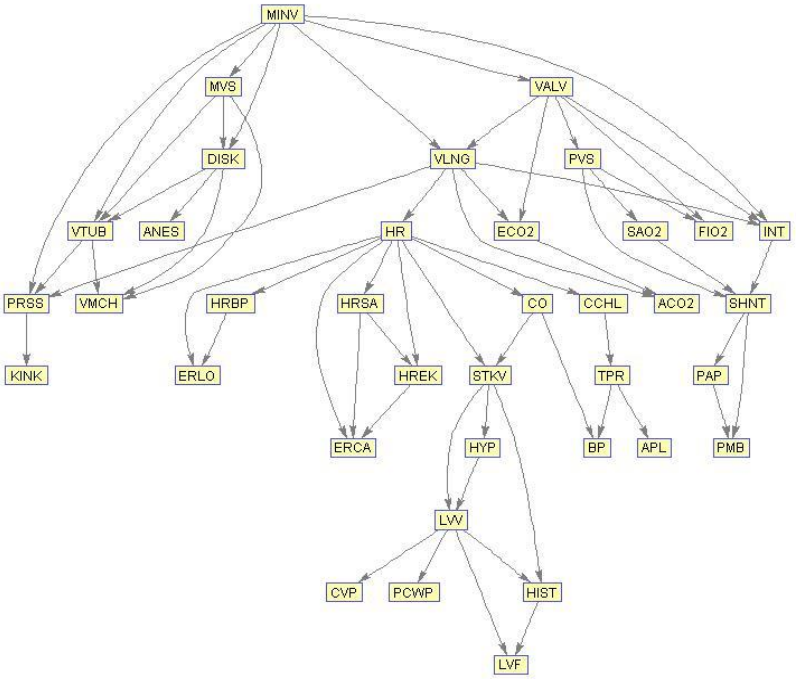
پایان

والدین مربوط به متغیر x_i را چاپ کنید.

پایان

پایان الگوریتم K۲

مثال۶-۸- شبکهٔ بیزی ساخته‌شده حاصل از الگوریتم $K۲$ روی مجموعه‌دادهٔ آلارم در شکل ۶ -۱۴ نشان داده شده است.



شکل شماره ۶-۱۴: ساختار شبکهٔ بیزی مجموعه‌دادهٔ آلارم با استفاده از الگوریتم $K2$

لازم به ذکر است که ترتیب انتخاب متغیرها بر اساس مقایسه بی‌نظمی (آنتروپی) بین متغیرها تعیین شده است.

یک پروژه نمونه

مقدمه

هدف این پروژه، یافتن گروه هایی از مشتریان عمده فروشی بر اساس کالای خریداری شده است. مجموعه داده استفاده شده، مجموعه داده مشتریان عمده فروشی از دامنه عمومی خواهد بود.

۶۰ خوشه بندی

این مجموعه داده به عنوان خوشه بندی بخش مشتری انتخاب شده است که یکی از موضوعات رایج است. خوشه‌بندی متغیرها/ویژگی‌ها با استفاده از یادگیری بدون نظارت اغلب امکان کشف شباهت زیاد بین طبقاتی و عدم تشابه زیاد درون کلاسی را فراهم می‌کند. این خوشه بندی به راحتی در زمینه های دیگر در کنار بخش مشتری قابل استفاده است. حوزه های دیگری مانند بیمه خودرو، کاربران کارت اعتباری، ریزش مشتریان مخابراتی و خریداران تجارت الکترونیک از حوزه هایی هستند که خوشه بندی برای آن ها قابل استفاده است.

از آنجایی که این مدل‌سازی‌ها براساس داده‌های تاریخی با فرضیات معین است. این پروژه به سایر عوامل خارجی مانند نحوه خرید (کارت اعتباری / نقدی)، تحویل مستقیم یا خود جمع آوری و زمان تراکنش خرید توجهی نکرده است.

داده ها

FRESH: هزینه سالانه (m.u.) برای محصولات تازه (پیوسته)

MILK: هزینه سالانه (m.u.) برای محصولات شیر (پیوسته)

GROCERY: هزینه سالانه (m.u.) برای محصولات خواربارفروشی (پیوسته)

FROZEN: هزینه سالانه (m.u.) برای محصولات منجمد (پیوسته)

DETERGENTS_PAPER: هزینه سالانه (m.u.) برای مواد شوینده و محصولات کاغذی (پیوسته)

DELICATESSEN: هزینه سالانه (m.u.) برای محصولات و اغذیه فروشی ها (پیوسته)

CHANNEL: کانال مشتریان - Horeca (هتل/رستوران/کافه) یا کانال خرده فروشی (اسمی)

REGION: منطقه مشتریان - Lisnon، Oporto یا دیگر (اسمی)

تحلیل اکتشافی داده ها

```
library(stats)
library(ggplot2)
library(plyr)
library(dplyr)
```

```
library(data.table)
library(corrplot)
library(factoextra)
```

نگاهی اجمالی به داده ها به شرح زیر است.

```
library(NbClust)
library (cluster)
df.original<-read.csv('Wholesale customers data.csv')
set.seed(1) #Ensure reproducable code
df<- sample_frac(df.original,0.7) #split into test and t
rain data by 7:3 ratio
df.index<- as.numeric(rownames(df))
df.test<- df.original[-df.index,]
head(df)
```

```
##     Channel Region Fresh  Milk Grocery Frozen Deterge
nts_Paper Delicassen
## 117       1      3 11173  2521    3355   1517
310        222
## 164       2      3  5531 15726   26870   2367
13726        446
## 251       1      1  3191  1993    1799   1730
234        710
## 397       2      3  4515 11991    9345   2644
3378       2213
## 88        1      3 43265  5025    8117   6312
1579      14351
## 391       1      3  3352  1181    1328   5502
311       1000
```

برای مقایسه با توضیحات ارایه شده درباره داده ها، به رکوردهای نمونه در مجموعه داده نگاه کنید.

۶ دسته محصول با هزینه سالانه توسط گروه مشتریان عمده فروشی توسط کانال و منطقه وجود دارد.

```
summary(df)
```

```
##     Channel          Region          Fresh             Milk
##  Min.   :1.000   Min.   :1.000   Min.   :    18   Min.   :   55
##  1st Qu.:1.000   1st Qu.:2.000   1st Qu.:  3094   1st Qu.: 1610
##  Median :1.000   Median :3.000   Median :  8130   Median : 3684
##  Mean   :1.331   Mean   :2.513   Mean   : 12059   Mean   : 5716
##  3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.: 16851   3rd Qu.: 7119
##  Max.   :2.000   Max.   :3.000   Max.   :112151   Max.   :73498
##     Grocery          Frozen        Detergents_Paper  Delicassen
##  Min.   :    3   Min.   :   33.0   Min.   :    3.0   Min.   :    3.0
##  1st Qu.: 2156   1st Qu.:  805.8   1st Qu.:  282.0   1st Qu.:  395.0
##  Median : 4904   Median : 1619.0   Median :  824.5   Median :  944.5
##  Mean   : 7729   Mean   : 3089.2   Mean   : 2764.6   Mean   : 1558.2
##  3rd Qu.:10550   3rd Qu.: 3532.5   3rd Qu.: 4003.2   3rd Qu.: 1820.2
##  Max.   :59598   Max.   :60869.0   Max.   :26701.0   Max.   :47943.0
```

یک اطلاعات خلاصه سریع در مورد متغیرها را می توان به شرح زیر نتیجه گرفت:

فقط دسته های غذایی از نظر آماره های توصیفی معنی دارند.

نیازی به مقیاس نیست زیرا واحدهای اندازه گیری یکسان است. ما از دسته بندی محصولات برای خوشه بندی استفاده خواهیم کرد.

```
summary(is.na(df))
```

```
##   Channel          Region          Fresh             M
ilk
##  Mode :logical   Mode :logical   Mode :logical   Mode
:logical
##  FALSE:308       FALSE:308       FALSE:308       FALS
E:308
##   Grocery          Frozen        Detergents_Paper Del
icassen
##  Mode :logical   Mode :logical   Mode :logical    Mod
e :logical
##  FALSE:308       FALSE:308       FALSE:308        FAL
SE:308
```

در این بخش به بررسی متغیرهای Null پرداخته شده است. می توان دید که مجموعا ۴۴۰ مشاهده وجود دارد و هیچ کدام از متغیرها دارای مقدار گمشده نیستند.

```
table(df$Channel)
```

```
##
##   1   2
## 206 102
```

در این بخش فراوانی متغیرهای رسته ای مشاهده می شود. می توان دید که ۲۹۸ مشاهده وجود دارد که از دسته بندی Horeca (هتل، رستوران، کافه) خرید کرده اند و ۱۴۲ مشاهده از خرده فروشی خرید کرده اند.

```
table(df$Region)
```

```
##
##   1   2   3
##  57  36 215
```

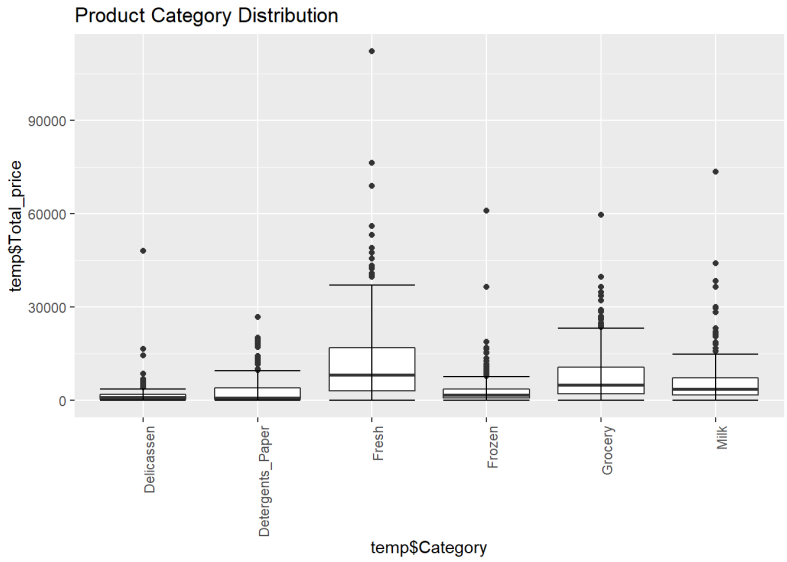
در رابطه با متغیر منطقه می توان ترکیب بالا را مشاهده کرد.

```
df %>%

  group_by(Channel,Region) %>%
# multiple group columns

  summarise(total_fresh = sum(df$Fresh), total_Milk = su
m(df$Milk), total_Grocery= sum(df$Grocery),total_Frozen=
sum(df$Frozen),total_Detergents_Paper=sum(df$Detergents_
Paper), total_Delicassen= sum(df$Delicassen))  # multipl
e summary columns
```

```
## # A tibble: 6 x 8

## # Groups:   Channel [?]

##   Channel Region total_fresh total_Milk total_Grocery
total_Frozen

##     <int>  <int>       <int>      <int>         <int>
<int>

## 1       1      1     3714211    1760547       2380616
951463

## 2       1      2     3714211    1760547       2380616
951463

## 3       1      3     3714211    1760547       2380616
951463

## 4       2      1     3714211    1760547       2380616
951463

## 5       2      2     3714211    1760547       2380616
951463

## 6       2      3     3714211    1760547       2380616
951463

## # ... with 2 more variables: total_Detergents_Paper <
int>,
```

در بخش بالا مجموع هزینه برای دسته های مختلف محصول را نشان می دهد. دسته های تازه و مواد غذایی پرفروش ترین ها هستند.

```
temp <- reshape(df, direction="long", varying=c("Fresh",
"Milk","Grocery","Frozen","Detergents_Paper", "Delicasse
n"), v.names= "Total_price", timevar="Category", time=c(
"Fresh", "Milk","Grocery","Frozen","Detergents_Paper", "
Delicassen"))


ggplot(temp, aes(x=temp$Category, y =temp$Total_price))
+geom_boxplot() +stat_boxplot(geom ='errorbar') + theme(
axis.text.x= element_text(angle=90,hjust=1))+ ggtitle("P
roduct Category Distribution")
```



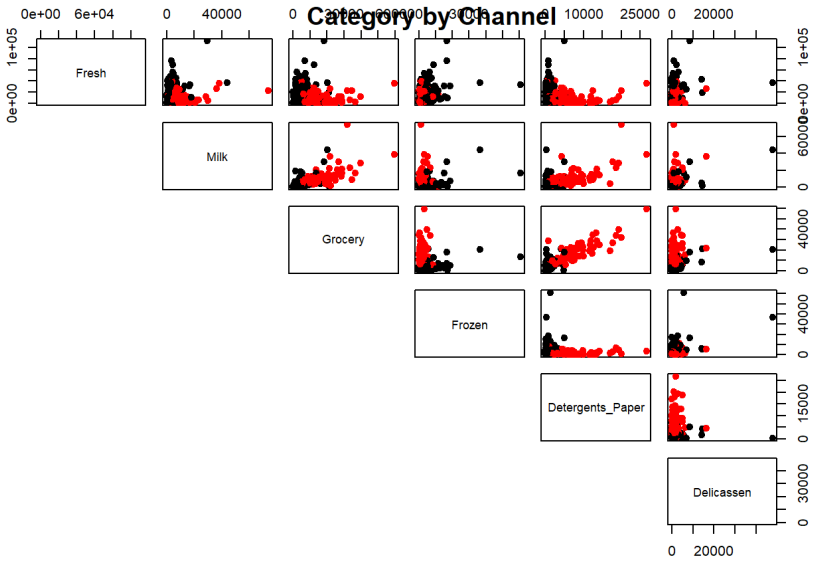
نمودار فوق،

یافته های زیر را ارایه می کند:

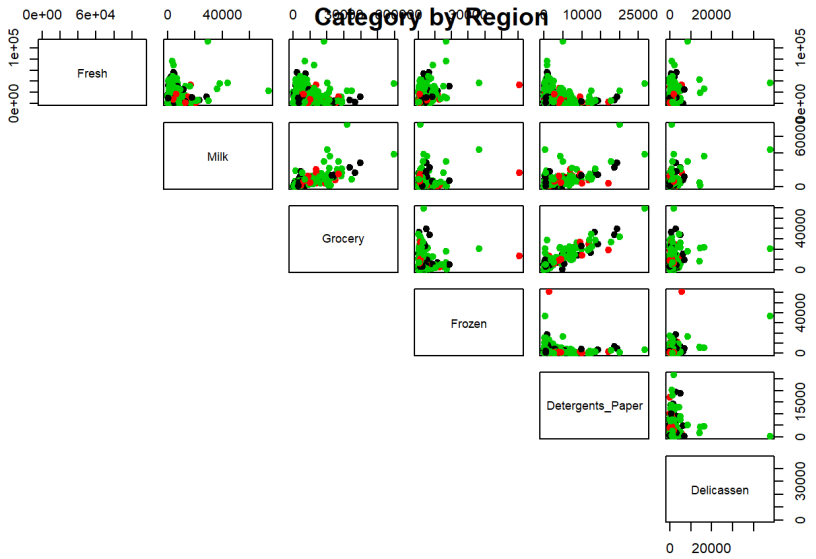
1. تعداد نقاط دور افتاده کمی برای برخی از گروه ها وجود دارد. در نتیجه دو راه حل برای خوشه بندی می توان ارایه داد.
   1.1. نقاط دورافتاده را حذف کنید و سپس الگوریتم K mean را اجرا کنید.
   1.2. بدون حذف موارد پرت، الگوریتم k medoids (PAM) را با فاصله منهتن اجرا کنید.
2. واریانس بالایی برای کالاهای دسته مواد غذایی تازه و غلات وجود دارد.

در ادامه، ما می‌خواهیم نموداری را به صورت سالیانه ارایه کنیم تا دیدی جامع ازفروش داشته باشیم.

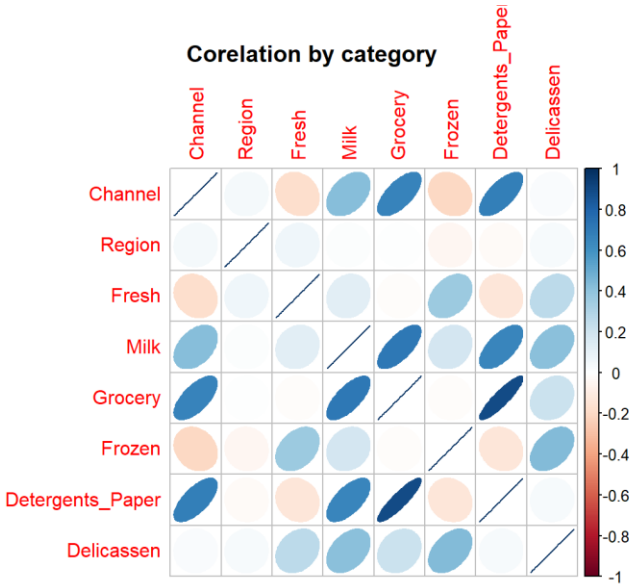
```
cor.result<- cor(df)

pairs(df[,-c(1:2)], col=df$Channel, pch=19, lower.panel
= NULL) +title(main = "Category by Channel")
```




```
plot(df[,-c(1:2)], col=df$Region, pch=19, lower.panel =
NULL)  +title(main = "Category by Region")
```




```
corrplot(cor.result, method="ellipse") +title(main = "Co
relation by category")
```



نمودارهای فوق، یافته های زیر را ارایه می کنند:

مواد شوینده، کاغذ و مواد غذایی بیشترین همبستگی را دارند.

در ادامه بررسی می کنیم که کدام متغیرها دارای نقاط پرت هستند.

```
        #Remove Region & Channel Columns as they are not
sensible

apply(X= df[,-c(1:2)],MARGIN=2, FUN = function(x)length(
boxplot.stats(x)$out))
```

```
##            Fresh             Milk          Grocery
Frozen

##               14               20               17
29

## Detergents_Paper        Delicassen

##               21               18
```

در بخش بالا، عدد نشان داده شده، تعداد نقاط پرت در هر دسته است. برای حذف نقاط پرت، از تکنیک Winsorizing استفاده خواهد شد. به طور خلاصه، نقاط پرت با صدک خاصی از داده ها، معمولاً ۹۰ یا ۹۵ جایگزین می شوند. ابتدا مقدار هر دسته را مرتب می کنیم.

```
        sort(boxplot.stats(df$Grocery)$out)
```

```
##  [1] 23596 23998 24708 24773 25957 26839 26866 26870
28540 28921 28986

## [12] 32114 33586 34792 36486 39694 59598
```

صدک ها به شکل زیر حاصل می شود.

```
quantile(df$Grocery, probs=seq(from =0.9, to=1,by=0.025)
)
```

```
##       90%     92.5%       95%     97.5%      100%

## 19258.40 21198.98 23857.30 28663.83 59598.00
```

از بخش بالا، صدک ۹۵٪ انتخاب می شود. در مرحله بعد، صدک ۹۵ جایگزین نقطه پرت باقی مانده خواهد شد.

```
grocery.max <- as.numeric(quantile(df$Grocery,probs=0.95
))

df$Grocery[df$Grocery > grocery.max] <- grocery.max
```

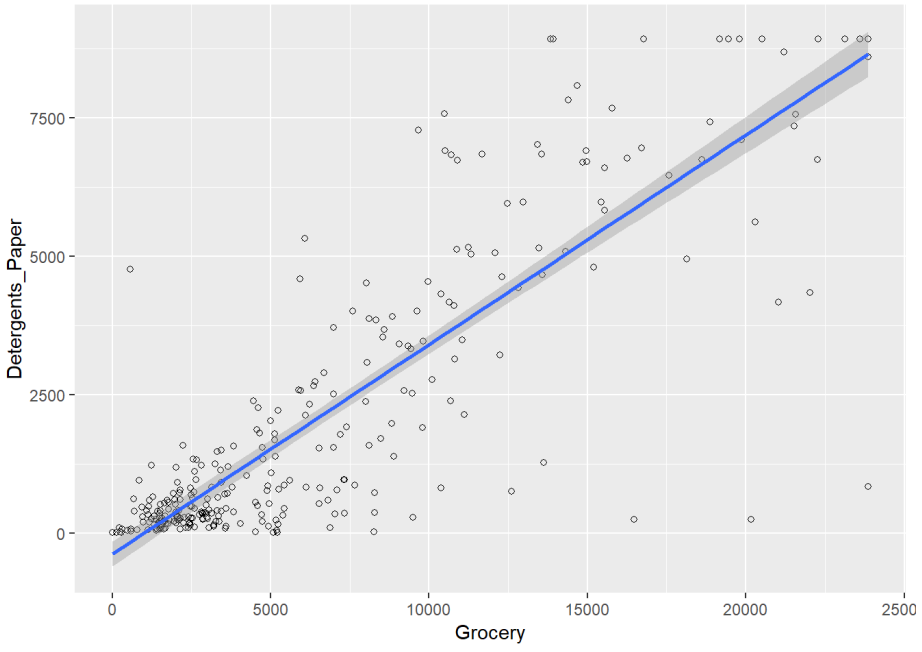
```
sort(boxplot.stats(df$Detergents_Paper)$out)

quantile(df$Detergents_Paper, probs=seq(from =0.9, to=1,
by=0.025))
```

```
##        90%      92.5%        95%      97.5%       100%

##  7464.900  8926.725 11710.900 13629.475 26701.000
```

```
grocery.max <- as.numeric(quantile(df$Detergents_Paper,p
robs=0.925))

df$Detergents_Paper[df$Detergents_Paper > grocery.max] <
- grocery.max
```

در ادامه دو متغیر Detergents_Paper و Grocery برای خوشه بندی انتخاب می شوند.

```
ggplot(data=df, aes(x=Grocery, y =Detergents_Paper)) + g
eom_point(shape=1) +geom_smooth(method="lm")
```




```
df.subset1<-as.data.frame(df[,c("Grocery","Detergents_Pa
per")])

summary(df.subset1)
```

```
##     Grocery      Detergents_Paper
##  Min.   :    3   Min.   :   3.0
##  1st Qu.: 2156   1st Qu.: 282.0
##  Median : 4904   Median : 824.5
##  Mean   : 7336   Mean   :2398.7
##  3rd Qu.:10550   3rd Qu.:4003.2
##  Max.   :23857   Max.   :8926.7
```

برای متغیر های انتخاب شده نیاز به نرمال سازی وجود دارد زیرا تفاوت زیادی میان آماره های نمایش داده شده وجود دارد.

```
df.subset1<- as.data.frame(scale(df.subset1))
summary(df.subset1)
```

```
##     Grocery        Detergents_Paper
##  Min.   :-1.0922   Min.   :-0.8308
##  1st Qu.:-0.7715   1st Qu.:-0.7341
##  Median :-0.3624   Median :-0.5459
##  Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.4786   3rd Qu.: 0.5565
##  Max.   : 2.4606   Max.   : 2.2640
```

مدل سازی

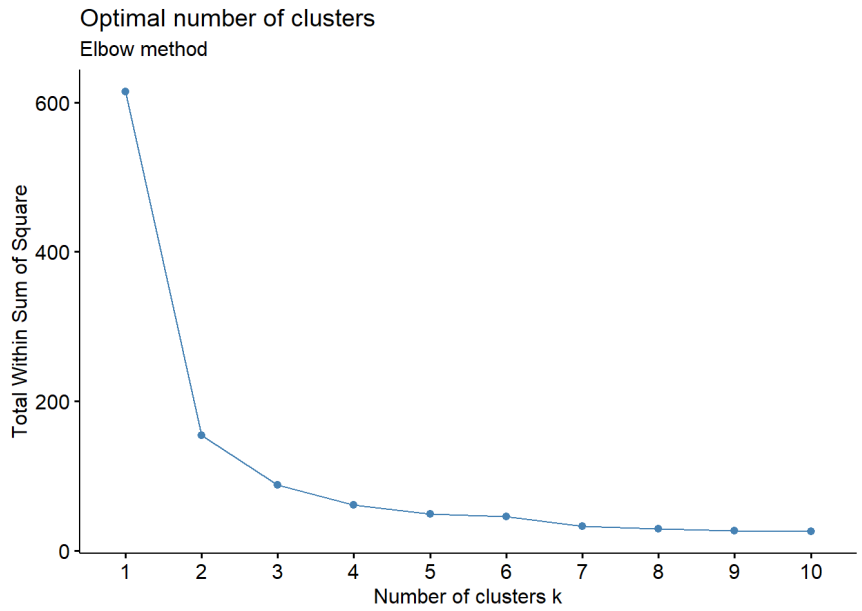
یافتن تعداد بهینه خوشه ها

پس از انجام مقیاس بندی، روش های زیر برای یافتن تعداد بهینه خوشه ها اجرا می شود:

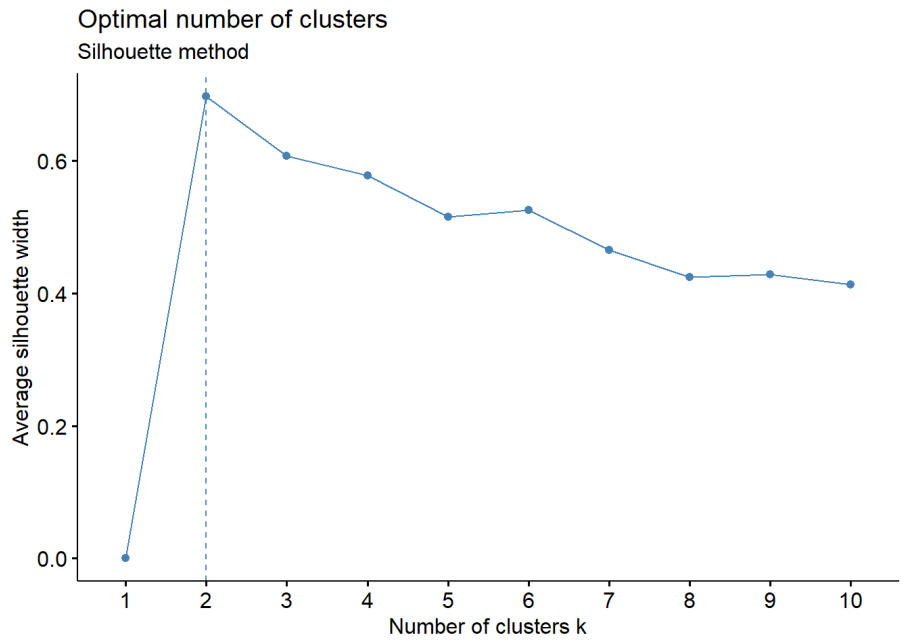
1. روش زانویی
2. . روش میانگین سیلوئت
3. . روش Gap Statisit

برای نمایش بهتر نتایج، ۳ تست اجرا خواهد شد.

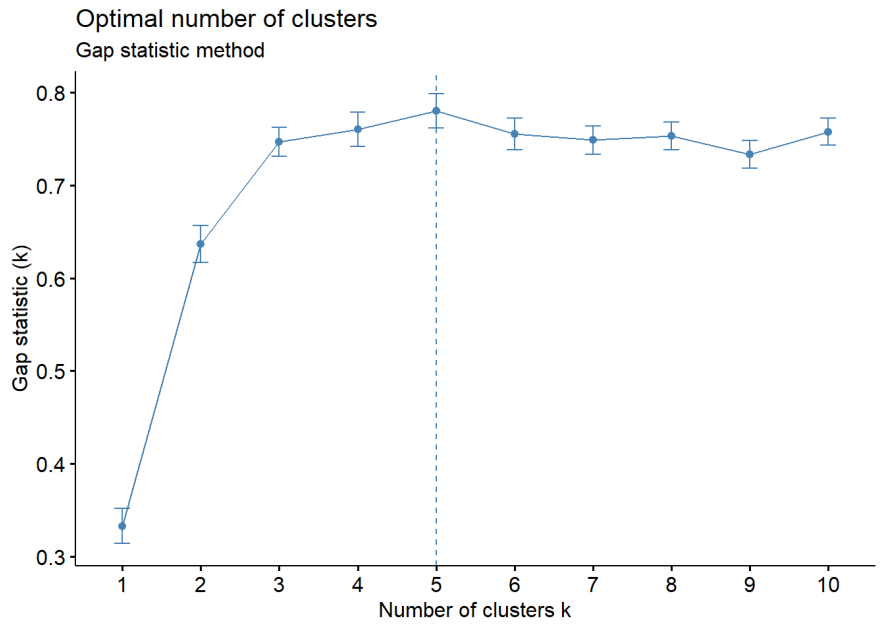
```
        set.seed(102)
# Elbow method
fviz_nbclust(df.subset1, kmeans, method = "wss") +
  labs(subtitle = "Elbow method")
```




```
# Silhouette method
fviz_nbclust(df.subset1, kmeans, method = "silhouette")+
  labs(subtitle = "Silhouette method")
```




```
# Gap statistic
# nboot = 50 to keep the function speedy.
# recommended value: nboot= 500 for your analysis.
# Use verbose = FALSE to hide computing progression.
set.seed(123)
fviz_nbclust(df.subset1, kmeans, nstart = 25,  method =
"gap_stat", nboot = 50)+
  labs(subtitle = "Gap statistic method")
```



۲ خوشه بر اساس تست های (۲ از ۳ تست) بالا انتخاب شده است. در ادامه، مرکز خوشه ها را با علامت مثلث به عنوان مرکز ترسیم می کنیم.

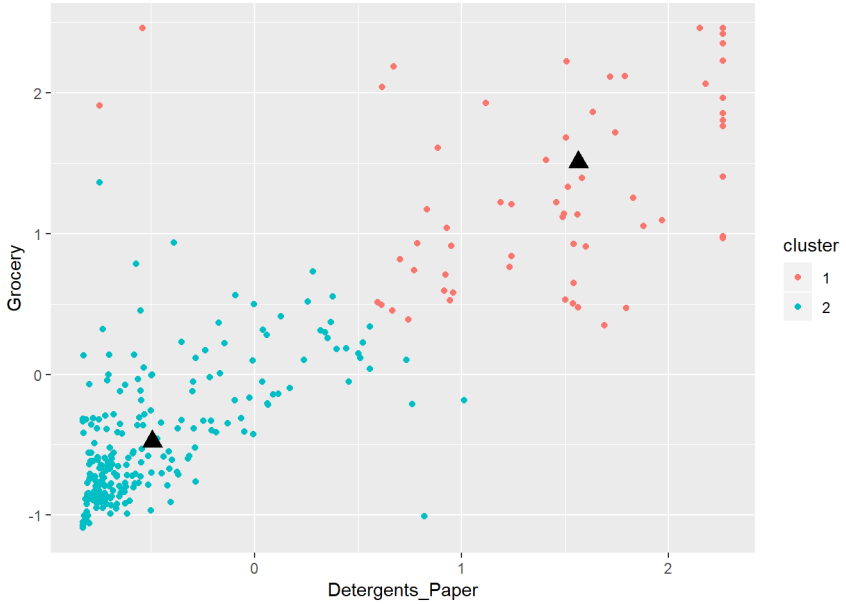
K means

```
set.seed(111)

kmean2.simple <- kmeans(df.subset1,centers=2, iter.max = 25, nstart=100)

df.subset1$cluster <- factor(kmean2.simple$cluster)

summary(df.subset1)
```

```
##     Grocery         Detergents_Paper  cluster
##  Min.   :-1.0922   Min.   :-0.8308   1: 74
##  1st Qu.:-0.7715   1st Qu.:-0.7341   2:234
##  Median :-0.3624   Median :-0.5459
##  Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.4786   3rd Qu.: 0.5565
```

```
##  Max.   : 2.4606   Max.   : 2.2640
```

```
ggplot(data=df.subset1, aes(x=Detergents_Paper, y=Grocer
y, colour=cluster))+geom_point()+geom_point(data=as.data
.frame(kmean2.simple$centers), color ="black", size=4, s
hape =17)
```



نمودار بالا ۲ خوشه را بر اساس ۲ متغیر، Grocery و Detergents_paper بر اساس الگوریتم k میانگین نشان می دهد. می توان دید که این دو متغیر داده های ما را به خوبی از یکدیگر جدا کرده است و خوشه های مطلوبی ایجاد کرده است.

نتیجه گیری

همان طور که مشاهده کردیم نقاط دورافتاده تاثیر بسزایی در عملیات خوشه بندی دارند. در این مطالعه از دو متغیر Grocery و Detergents_paper برای خوشه بندی داده ها استفاده کردیم. این دو متغیر هم بستگی بالایی دارند. به عبارتی افزایش میزان Grocery می تواند نشان دهنده ی افزایش میزان Detergents_paper باشد. در خوشه بندی انجام شده دو خوشه یافت شد. یک خوشه که به صورت میانگین هزینه پایینی برای مواد شوینده، کاغذ و خواربار میپردازند و خوشه ای که هزینه بالایی برای این موارد پرداخت می کنند.

۷۵ خوشه بندی

یک تیم مارکتینگ قدرتمند می تواند از خروجی این خوشه بندی برای برنامه ریزی کمپین های مارکتینگ خود و ارایه پیشنهادات جذاب به مشتریان با توجه به خوشه ای که در آن قرار میگیرد

پیوست
کدهای R

```
#kmeans
data(ChickWeight)
ChickWeight
install.packages("amap")
library(amap)
x1=ChickWeight$weight
x2=ChickWeight$Time
x=cbind(x1,x2)
x
cl<-Kmeans(x,4, iter.max =85, method = "euclidean")
plot(x, col = cl$cluster)
points(cl$centers, col = 1:4, pch = 8, cex=2)
data(iris)
iris2=iris
iris2$Species=NULL
kmeans.result=kmeans(iris2,3)
kmeans.result
table(iris$Species, kmeans.result$cluster)
plot(iris2[c("Sepal.Length","Sepal.Width")],

col=kmeans.result$cluster,sub=paste("DataMining",format(Sys.time()
,"%Y-%b-%d %H:%S")))
points(kmeans.result$centers[,c("Sepal.Length", "Sepal.Width")],
col=1:3, pch=8, cex=2)
#########################
#pam1
library(datasets)
data(USArrests)
df=scale(USArrests)
head(df,n=3)
install.packages(c("cluster","factoextera"))
library(cluster)
```

```
library(factoextra)
#the optimal number of clusters
library(cluster)
fviz_nbclust(df ,cluster::pam,method="silhouette")
pam.res=pam(df,k=2,metric="euclidean" )
pam.res
d=cbind(USArrests,cluster=pam.res$cluster)
head(d,n=3)
library(factoextra)
fviz_cluster(pam.res,geom="point",ellipse.type="norm")
#pam2
library(fpc)
data(iris)
iris2=iris
iris2$Species=NULL
pamk.result=pamk(iris2,2:100)
pamk.result$nc
pamk.result$pamobject$medoids
table(pamk.result$pamobject$clustering,iris$Species)
################################
#سلسله مراتبی
library(stats)
library(factoextra)
n=5
k=3
data=iris[c(sample(x = 1:50,size = n),sample(x
=51:100,size=n),sample(x = 101:150,size= n)),1:4]
distmethod=c('euclidean')
linkagemethod=c("average")
distance=dist(data,distmethod)
hc=hclust(d = distance,method=linkagemethod)
distance
fviz_dist(dist.obj = distance)
hc
fviz_dend(x = hc,k = k)
###################
```

```
#DBSCAN
data("multishapes")
df=multishapes[1:2,]
library("fpc")
set.seed(123)
db=fpc::dbscan(df,eps=0.15,minpts=5)
print(db)
library("factoextra")
fviz_cluster(db,data=df,stand=FALSE,ellipse=FALSE,
show.clust.cent=FALSE
+geom="point",palette="jco",ggtheme=theme_classic())
#OPTICS
data(iris)
iris2=iris
iris2$Species=NULL
library(NbClust)
nc=NbClust(iris2,min.nc=2,max.nc=15,method="kmeans")
BAR=table(nc$Best.nc[1,])
BAR
barplot(BAR,xlab="Number Of Cluster",ylab="Number Of Criteria",
main = "Number
+Of Cluster Choosen by 26 Criterian")
##############################3
library(stats)
library(ggplot2)
library(plyr)
library(dplyr)
library(data.table)
library(corrplot)
library(NbClust)
library (cluster)
df.original<-read.csv("C:/Users/ASUS/Desktop/Wholesale customers
data.csv")
df<- sample_frac(df.original,0.7)
df.index<- as.numeric(rownames(df))
df.test<- df.original[-df.index,]
```

```
head(df)
summary(df)
summary(is.na(df))
table(df$Channel)
table(df$Region)
df %>% group_by(Channel,Region) %>%
  summarise(total_fresh = sum(df$Fresh), total_Milk = sum(df$Milk),
total_Grocery= sum(df$Grocery),

total_Frozen=sum(df$Frozen),total_Detergents_Paper=sum(df$Deter
gents_Paper),
          total_Delicassen= sum(df$Delicassen))

temp <- reshape(df, direction="long",
varying=c("Fresh","Milk","Grocery","Frozen","Detergents_Paper",
"Delicassen"), v.names= "Total_price", timevar="Category",
time=c("Fresh", "Milk","Grocery","Frozen","Detergents_Paper",
"Delicassen"))

ggplot(temp, aes(x=temp$Category, y =temp$Total_price))
+geom_boxplot() +stat_boxplot(geom ='errorbar') +
theme(axis.text.x= element_text(angle=90,hjust=1))+ ggtitle("Product
Category Distribution")
cor.result<- cor(df)
pairs(df[,-c(1:2)], col=df$Channel, pch=19, lower.panel = NULL)
+title(main = "Category by Channel")
plot(df[,-c(1:2)], col=df$Region, pch=19, lower.panel = NULL)
+title(main = "Category by Region")

corrplot(cor.result, method="ellipse") +title(main = "Corelation by
category")
apply(X= df[,-c(1:2)],MARGIN=2, FUN =
function(x)length(boxplot.stats(x)$out))
sort(boxplot.stats(df$Grocery)$out)
quantile(df$Grocery, probs=seq(from =0.9, to=1,by=0.025))
grocery.max <- as.numeric(quantile(df$Grocery,probs=0.95))
```

```
df$Grocery[df$Grocery > grocery.max] <- grocery.max
sort(boxplot.stats(df$Detergents_Paper)$out)
quantile(df$Detergents_Paper, probs=seq(from =0.9, to=1,by=0.025))
grocery.max <-
as.numeric(quantile(df$Detergents_Paper,probs=0.925))
df$Detergents_Paper[df$Detergents_Paper > grocery.max] <-
grocery.max

ggplot(data=df, aes(x=Grocery, y =Detergents_Paper)) +
  geom_point(shape=1) +geom_smooth(method="lm")

df.subset1<-as.data.frame(df[,c("Grocery","Detergents_Paper")])
summary(df.subset1)

df.subset1<- as.data.frame(scale(df.subset1))
summary(df.subset1)

set.seed(102)
# Elbow method
fviz_nbclust(df.subset1, kmeans, method = "wss") +
  labs(subtitle = "Elbow method")

fviz_nbclust(df.subset1, kmeans, method = "silhouette")+
  labs(subtitle = "Silhouette method")

set.seed(123)
fviz_nbclust(df.subset1, kmeans, nstart = 25,  method = "gap_stat",
nboot = 50)+
  labs(subtitle = "Gap statistic method")

set.seed(111)
kmean2.simple <- kmeans(df.subset1,centers=2, iter.max = 25,
nstart=100)
df.subset1$cluster <- factor(kmean2.simple$cluster)
summary(df.subset1)
```

```
ggplot(data=df.subset1, aes(x=Detergents_Paper, y=Grocery,
colour=cluster))+geom_point()+geom_point(data=as.data.frame(kme
an2.simple$centers), color ="black", size=4, shape =17)
```