پروژه شماره ۵ : خوشه بندی با داده‌های واقعی

زمستان ۱۴۰۰

مقدمه

بورس چیست و چه کاربردی دارد؟ (کار بورس چیست و بازار بورس چگونه است)

برای اینکه دقیقا متوجه شوید معنی بورس چیست، یک تعریف ساده علمی از بورس ارائه می کنیم.

برای تعریف بورس، به زبان ساده باید گفت بورس یک نوع بازار بوده که در آن دارایی های مختلفی مورد معامله قرار می گیرند. بازار بورس همچنین یک نهاد ساختاریافته است که به عنوان یکی از مهمترین ارکان اساسی بازار سرمایه نیز شناخته می شود.

مدیریت انتقال ریسک، شفافیت بازار، کشف قیمت، ایجاد بازار رقابتی و مهمتر از همه جمع آوری سرمایه و پس اندازهای اندک برای تامین سرمایه فعالیت های اقتصادی از مهمترین کاربردهای بورس محسوب می شوند.

نکته : همانطور که در هر بازاری ممکن است دارایی های مالی، فیزیکی، غیر فیزیکی و ... فروخته شود، در بازار بورس هم دارایی های مختلف معامله می شوند. به همین دلیل بورس انواع مختلفی دارد.

بازار بورس ارز چیست ؟

همانطور که مشخص است در این بازار بورس تنها خرید و فروش معامله پول کشورهای دیگر (ارز) انجام می شود. البته بورس ارز در ایران وجود ندارد اما در بسیاری از کشورهای توسعه یافته وجود داشته و پیشرفت قابل توجهی هم دارد.

جفت ارز (Currency pair) چیست؟

ترکیب و جفت‌کردن دو ارز (ارز فیات مانند دلار، یورو، ریال و... و ارزهای دیجیتالی) به منظور انجام برخی معاملات بر پایه ارزش ارزهای جفت شده روشی مرسوم در معاملات بازارهای مختلف بوده است. هرکدام از این جفت‌ها را یک جفت ارز می‌نامند.

نقش اصلی این جفت‌ها در واقع مقایسه ارزهای با یکدیگر و تعیین نرخی برای تبدیل آن‌ها به یکدیگر است. در واقع در بازارهای مالی مانند فارکس (بازار معاملات ارزهای خارجی) ارز جفت‌ارزها برای انجام معاملات از مبدأ کشورهای مختلف به یکدیگر با ارزهای اختصاصی هرکدام استفاده می‌شود.

ارزهای استفاده شده در این پروژه :

نماد جفت ارز	ارز ها
EURCAD	یورو/دلار کانادا
EURAUD	یورو / دلار استرالیا
EURJPY	یورو / ین ژاپن
EURCHF	یورو / فرانک سوئیس
EURGBP	یورو / پوند بریتانیا
USDJPY	دلار آمریکا / ین ژاپن
USDCHF	دلار آمریکا / فرانک سوئیس
USDCAD	دلار آمریکا / دلار کانادا

K-means Cluster Analysis

خوشه بندی مجموعه وسیعی از تکنیک ها برای یافتن زیر گروه های مشاهدات در یک مجموعه داده است.

وقتی مشاهدات را خوشه‌بندی می‌کنیم، می‌خواهیم مشاهدات در یک گروه مشابه و مشاهدات در گروه‌های مختلف غیرمشابه باشند.

از آنجایی که متغیر پاسخی وجود ندارد، این یک روش بدون نظارت است، که به این معنی است که به دنبال یافتن روابط بین **n** مشاهدات بدون آموزش متغیر پاسخ است.

خوشه بندی به ما این امکان را می دهد که تشخیص دهیم کدام مشاهدات مشابه هستند و به طور بالقوه آنها را در آن دسته بندی کنیم.

خوشه‌بندی **K-means** ساده‌ترین و متداول‌ترین روش خوشه‌بندی برای تقسیم یک مجموعه داده به مجموعه‌ای از گروه‌های **k** است.

روش تحقیق

ما در این پروژه تلاش داریم که مجموعه داده خود را به **k** گروه خوشه بندی کنیم.

مجموعه داده ما شامل ۸ جفت ارز است که برای هر جفت ارز تعدادی متغیر مانند اندیکاتورrsi و... محاسبه شده است.

داده‌های فوق مربوط به تاریخ ۲۰۲۱.۱۲.۲۴ هستند.

اجرای پروژه

برای تکرار تجزیه و تحلیل این آموزش، باید بسته های زیر را بارگیری کنید:

```
> library(tidyverse) # data manipulation
> library(cluster) # clustering algorithms
> library(factoextra) # clustering algorithms & visualization
```

آماده سازی داده ها

```
> Data<- read.table("~/4Proj.txt", header = T)
> Data<-Data[,-1]
```

برای انجام تجزیه و تحلیل خوشه ای در R، به طور کلی، داده ها باید به صورت زیر تهیه شوند:

سطرها مشاهدات (افراد) و ستون ها متغیر هستند

هر مقدار از دست رفته در داده ها باید حذف یا تخمین زده شود.

داده ها باید استاندارد شوند (یعنی مقیاس شده) تا متغیرها قابل مقایسه باشند. به یاد بیاورید که استانداردسازی شامل تبدیل متغیرها به گونه ای است که میانگین صفر و انحراف معیار یک داشته باشند.

برای حذف هر مقدار گمشده ای که ممکن است در داده ها وجود داشته باشد، این را تایپ کنید:

```
> Data <- na.omit(Data)
```

از آنجایی که نمی‌خواهیم الگوریتم خوشه‌بندی به یک واحد متغیر دلخواه وابسته باشد، با مقیاس‌بندی/استاندارد کردن داده‌ها با استفاده از دستور شروع می‌کنیم:

```
> Data <- scale(Data)
```

K-Means Clustering

خوشه‌بندی **K-means** متداول‌ترین الگوریتم یادگیری ماشینی بدون نظارت است که برای تقسیم‌بندی یک مجموعه داده به مجموعه‌ای از گروه‌های **k** (یعنی **k** خوشه‌ها) استفاده می‌شود، جایی که **k** نشان‌دهنده تعداد گروه‌هایی است که از قبل توسط تحلیلگر مشخص شده است. این اشیاء را در گروه های متعدد (به عنوان مثال، خوشه ها) طبقه بندی می کند، به طوری که اشیاء درون یک خوشه تا حد ممکن مشابه هستند (یعنی شباهت درون کلاسی بالا)، در حالی که اشیاء از خوشه های مختلف تا حد ممکن متفاوت هستند (یعنی کم بین شباهت طبقاتی). در خوشه بندی **k-means**، هر خوشه با مرکز خود (یعنی مرکز) نشان داده می شود که با میانگین نقاط اختصاص داده شده به خوشه مطابقت دارد.

الگوریتم K-means

اولین گام هنگام استفاده از خوشه بندی **k-means**، نشان دادن تعداد خوشه های (**k**) است که در راه حل نهایی ایجاد می شود. الگوریتم با انتخاب تصادفی **k** شی از مجموعه داده ها شروع می شود تا به عنوان مراکز اولیه برای خوشه ها خدمت کنند. اشیاء انتخاب شده همچنین به عنوان میانگین خوشه یا مرکز شناخته می شوند. در مرحله بعد، هر یک از اشیاء باقی مانده به نزدیکترین مرکز خود اختصاص داده می شود، جایی که نزدیکترین با استفاده از فاصله اقلیدسی بین شی و میانگین خوشه تعریف می شود. این مرحله "مرحله انتساب خوشه" نامیده می شود. پس از مرحله انتساب، الگوریتم مقدار میانگین جدید هر خوشه را محاسبه می کند. برای طراحی این مرحله از اصطلاح **Cluster Centroid update** استفاده می شود. اکنون که مراکز مجدداً محاسبه شده‌اند، هر مشاهده‌ای دوباره بررسی می‌شود تا ببینیم آیا ممکن است به خوشه دیگری نزدیک‌تر باشد یا خیر. همه اشیا مجدداً با استفاده از ابزار خوشه به روز شده تخصیص داده می شوند. مراحل تخصیص خوشه و به‌روزرسانی مرکز به‌طور مکرر تکرار می‌شوند تا زمانی که تخصیص‌های خوشه تغییر نکنند (یعنی تا زمانی که همگرایی حاصل شود). یعنی خوشه های تشکیل شده در تکرار فعلی همان خوشه های به دست آمده در تکرار قبلی هستند.

الگوریتم **K-means** را می توان به صورت زیر خلاصه کرد:

تعیین تعداد خوشه‌ها (**K**) برای ایجاد (توسط تحلیل‌گر) به‌طور تصادفی **k** شیء را از مجموعه داده‌ها به‌عنوان مراکز یا میانگین‌های اولیه خوشه انتخاب کنید.

هر مشاهده را بر اساس فاصله اقلیدسی بین جسم و مرکز به نزدیکترین مرکز خود اختصاص می دهد. برای هر یک از **k** خوشه ها مرکز خوشه را با محاسبه میانگین مقادیر جدید همه نقاط داده در خوشه به روز می کند. مرکز یک خوشه **Kth** یک بردار به طول **p** است که شامل میانگین همه متغیرها برای مشاهدات در خوشه **k**ام است. **p** تعداد متغیرها است.

به طور مکرر مجموع مجموع مربع را به حداقل برسانید. یعنی مراحل ۳ و ۴ را تکرار کنید تا زمانی که تخصیص های خوشه تغییر نکند یا به حداکثر تعداد تکرار برسد. به طور پیش فرض، نرم افزار **R** از ۱۰ به عنوان مقدار پیش فرض برای حداکثر تعداد تکرار استفاده می کند.

محاسبه **K-means** خوشه بندی در **R**

ما می توانیم **k-means** را در **R** با تابع **kmeans** محاسبه کنیم. در اینجا داده ها به دو خوشه (**centers = 2**) گروه بندی می شوند. تابع **kmeans** همچنین دارای یک گزینه **nstart** است که چندین پیکربندی اولیه را انجام می دهد و بهترین را گزارش می دهد. به عنوان مثال، افزودن **nstart =11** ، ۱۱ پیکربندی اولیه را ایجاد می کند. این رویکرد اغلب توصیه می شود.

```
> k2 <- kmeans(Data, centers = 2, nstart = 11)
> str(k2)
List of 9
 $ cluster     : Named int [1:11] 2 2 2 2 1 2 2 1 2 1 ...
  ..- attr(*, "names")= chr [1:11] "1" "2" "3" "4" ...
 $ centers     : num [1:2, 1:6] 1.211 -0.692 -0.0306 0.0175 1.2139 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:2] "1" "2"
  .. ..$ : chr [1:6] "Open" "Momentum" "iMA10" "iMA100" ...
 $ totss       : num 60
 $ withinss    : num [1:2] 10.2 13.6
 $ tot.withinss: num 23.8
 $ betweenss   : num 36.2
 $ size        : int [1:2] 4 7
 $ iter        : int 1
 $ ifault      : int 0
 - attr(*, "class")= chr "kmeans"
```

خروجی **kmeans** یک لیست با چند اطلاعات است :

cluster: بردار اعداد صحیح (از ۱ تا **k**) که نشان دهنده خوشه ای است که هر نقطه به آن اختصاص داده شده است.

centers: ماتریسی از مراکز خوشه ای.

totss: مجموع مجموع مربع ها.

insidess: بردار مجموع مربع های درون خوشه ای، یک جزء در هر خوشه.

tot.withinss: مجموع مجموع مجذورات درون خوشه ای، یعنی مجموع (**inthinss**).

betweenss: مجموع مربع های بین خوشه ای، یعنی **$totss-tot.withinss$**.

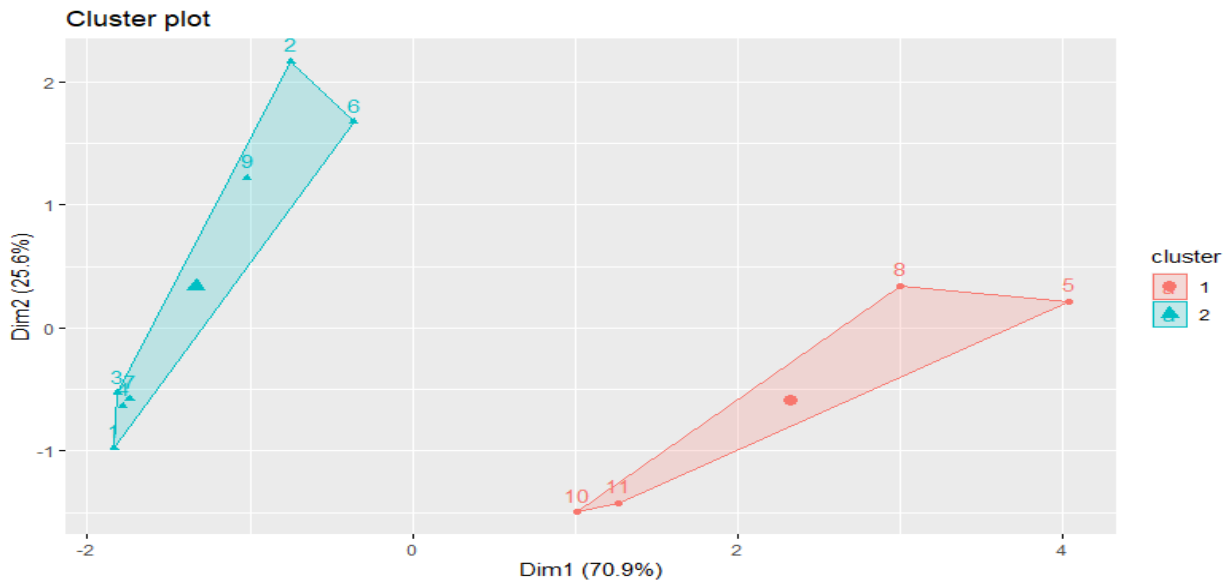
اندازه: تعداد نقاط در هر خوشه.

اگر نتایج را چاپ کنیم، می‌بینیم که گروه‌بندی‌های ما به ۲ اندازه خوشه **4** و **7** منجر شده است. ما مراکز خوشه (میانگین) دو گروه را در شش متغیر می‌بینیم. ما همچنین تخصیص خوشه را برای هر مشاهده دریافت می کنیم (یعنی **open** به خوشه ۲، **stdDev** به خوشه ۱ و غیره اختصاص داده شد).

```
> k2
K-means clustering with 2 clusters of sizes 4, 7

Cluster means:
        Open    Momentum     iMA10     iMA100     StdDev        RSI
1  1.2110313 -0.03061591  1.213861  1.2126131  1.0868986  0.4143065
2 -0.6920179  0.01749481 -0.693635 -0.6929218 -0.6210849 -0.2367466

Clustering vector:
 1  2  3  4  5  6  7  8  9 10 11
 2  2  2  2  1  2  2  1  2  1  1

Within cluster sum of squares by cluster:
[1] 10.19099 13.57550
 (between_SS / total_SS =  60.4 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betwe
enss"    "size"         "iter"         "ifault"
```

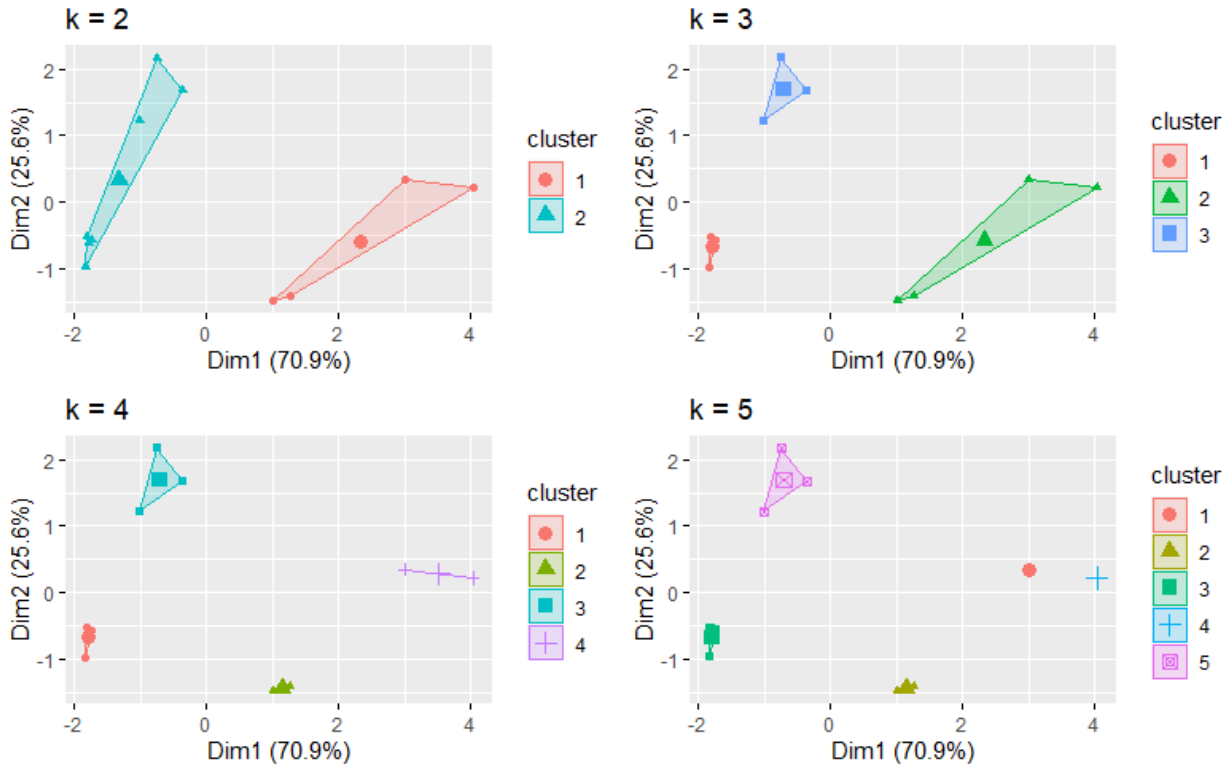
ما همچنین می توانیم نتایج خود را با استفاده از **fviz_cluster** مشاهده کنیم. این یک تصویر خوب از خوشه ها را ارائه می دهد. اگر بیش از دو بعد (متغیر) وجود داشته باشد، **fviz_cluster** تجزیه و تحلیل مؤلفه اصلی (**PCA**) را انجام می دهد و نقاط داده را بر اساس دو مؤلفه اصلی اول که اکثر واریانس را توضیح می دهد رسم می کند.

```
> fviz_cluster(k2, data = Data)
```



از آنجا که تعداد خوشه ها (k) باید قبل از شروع الگوریتم تنظیم شود، اغلب استفاده از چندین مقدار مختلف k و بررسی تفاوت در نتایج مفید است. ما می توانیم همان فرآیند را برای ۳، ۴ و ۵ خوشه اجرا کنیم و نتایج در شکل نشان داده شده است:

```
> k3 <- kmeans(Data, centers = 3, nstart = 11)
> k4 <- kmeans(Data, centers = 4, nstart = 11)
> k5 <- kmeans(Data, centers = 5, nstart = 11)
>
> # plots to compare
> p1 <- fviz_cluster(k2, geom = "point", data = Data) + ggtitle("k = 2")
> p2 <- fviz_cluster(k3, geom = "point",  data = Data) + ggtitle("k = 3")
> p3 <- fviz_cluster(k4, geom = "point",  data = Data) + ggtitle("k = 4")
> p4 <- fviz_cluster(k5, geom = "point",  data = Data) + ggtitle("k = 5")
>
> library(gridExtra)
> grid.arrange(p1, p2, p3, p4, nrow = 2)
```



اگرچه این ارزیابی بصری به ما می گوید که در کجا تقسیم بندی های واقعی رخ می دهد بنظر میرسد ۲ خوشه نتایج جالبی دارد.

تعیین خوشه های بهینه

همانطور که به یاد دارید، تحلیلگر تعداد خوشه های مورد استفاده را مشخص می کند. ترجیحاً تحلیلگر مایل است از تعداد بهینه خوشه ها استفاده کند. برای کمک به تحلیلگر، موارد زیر سه روش رایج برای تعیین خوشه های بهینه را توضیح می دهد که شامل:

1. Elbow method
2. Silhouette method
3. Gap statistic

محاسبه الگوریتم خوشه بندی به عنوان مثال، **k-means** خوشه بندی برای مقادیر مختلف **k.** به عنوان مثال، با تغییر **k** از ۱ تا ۱۰ خوشه

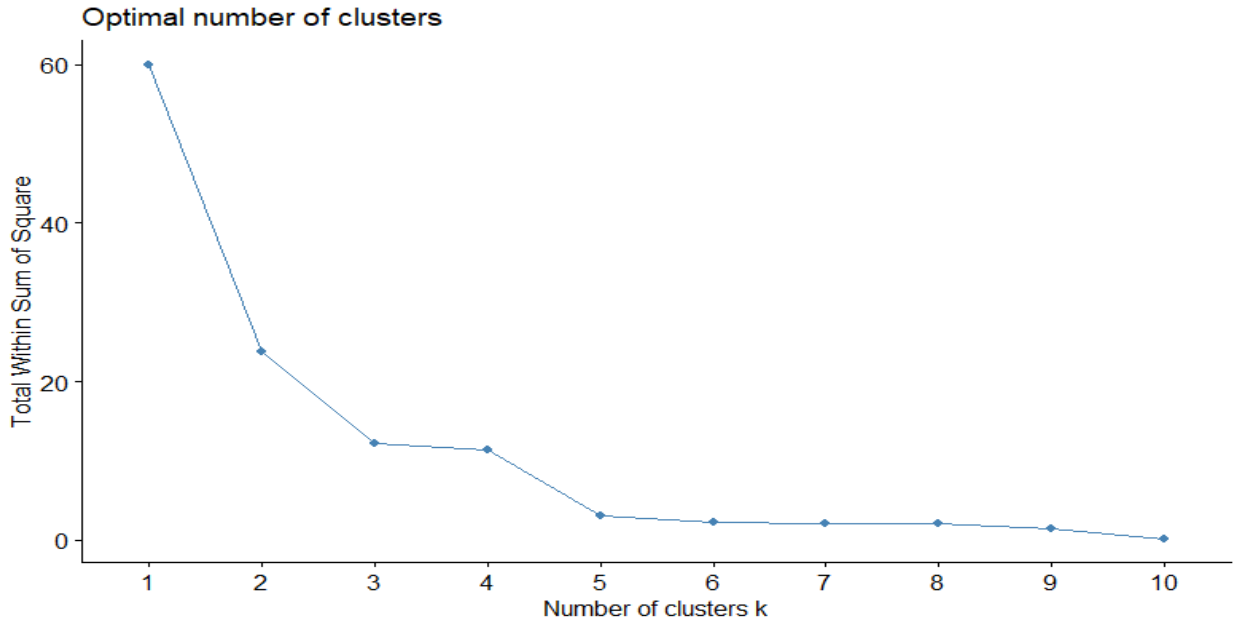
برای هر **k** ، مجموع مجموع مربع درون خوشه ای **(wss)** را محاسبه کنید.

منحنی **wss** را با توجه به تعداد خوشه های **k** رسم کنید.

محل یک خم (زانو) در طرح به طور کلی به عنوان شاخص تعداد مناسب خوشه در نظر گرفته می شود.

با کد زیر می توانیم این را در **R** پیاده سازی کنیم. نتایج نشان می دهد که **3** تعداد بهینه خوشه ها است زیرا به نظر می رسد خم شدن زانو (یا آرنج) باشد.

```
> set.seed(555)
> fviz_nbclust(Data, kmeans, method = "wss")
```



Average Silhouette Method

به طور خلاصه، رویکرد **silhouette** متوسط کیفیت یک خوشه‌بندی را اندازه‌گیری می‌کند.

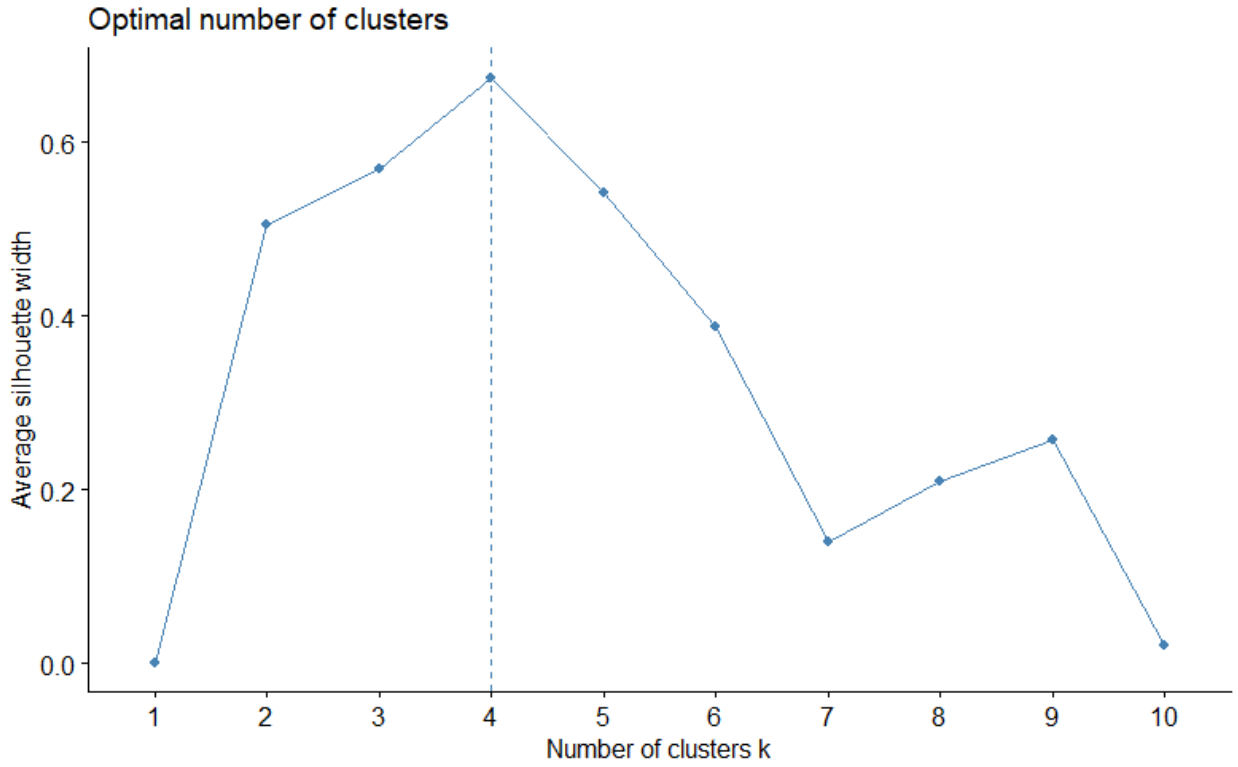
یعنی تعیین می کند که هر شی چقدر در خوشه خود قرار دارد.

میانگین عرض **Silhouette** بالا نشان دهنده خوشه بندی خوب است.

روش **silhouette** میانگین، **Average Silhouette** مشاهدات را برای مقادیر مختلف **k** محاسبه می کند.

تعداد بهینه خوشه‌های **k** عددی است که **Average Silhouette** را در محدوده‌ای از مقادیر ممکن برای **k.2** به حداکثر می‌رساند.

```
> fviz_nbclust(Data, kmeans, method = "silhouette")
```



مشاهده میشود که ۴ خوشه برای ما بهینه است.

Gap Statistic Method

آمار شکاف کل تغییرات درون خوشه ای را برای مقادیر مختلف **k** با مقادیر مورد انتظار آنها تحت توزیع مرجع صفر داده ها (یعنی توزیعی بدون خوشه بندی آشکار) مقایسه می کند.

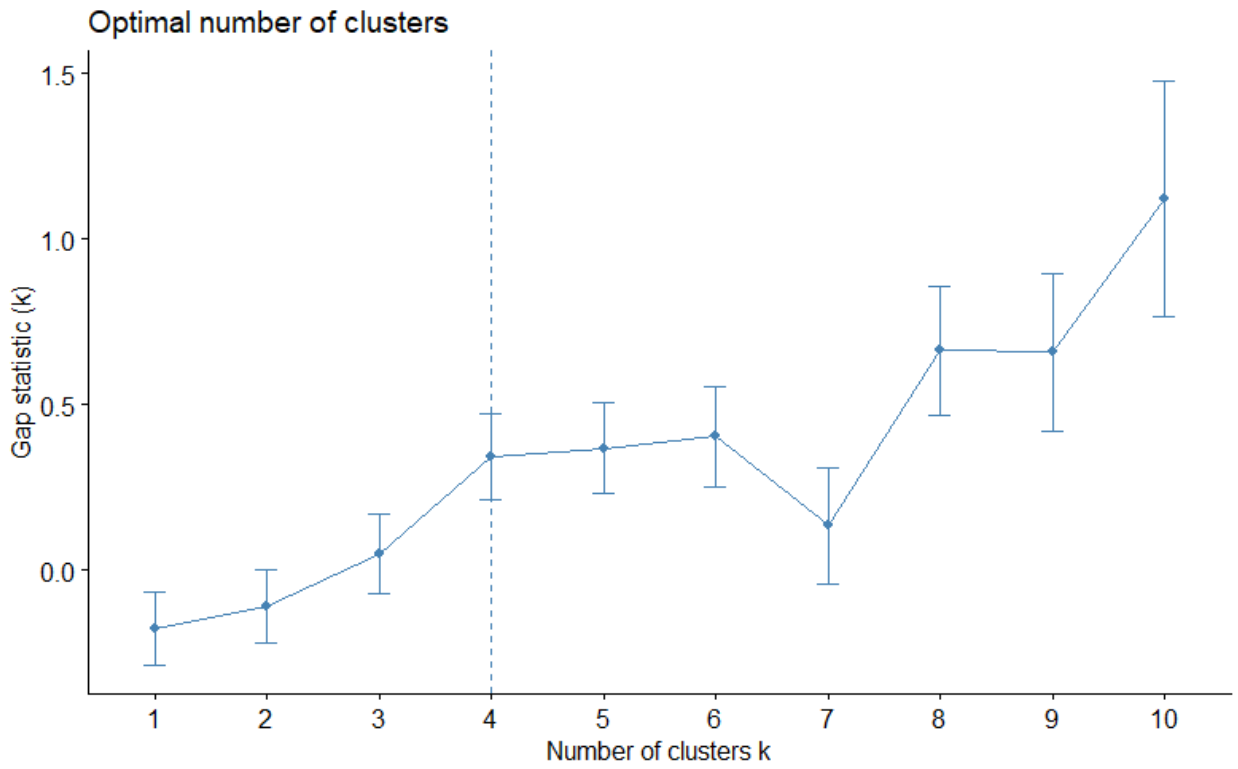
مجموعه داده مرجع با استفاده از شبیه سازی مونت کارلو از فرآیند نمونه گیری تولید می شود.

برای محاسبه روش آماری شکاف می‌توانیم از تابع **clusGap** استفاده کنیم که آمار شکاف و خطای استاندارد را برای خروجی ارائه می‌کند.

```
> # compute gap statistic
> set.seed(555)
> gap_stat <- clusGap(Data, FUN = kmeans, nstart = 11,
+                     K.max = 10, B = 50)
Clustering k = 1,2,..., K.max (= 10): .. done
Bootstrapping, b = 1,2,..., B (= 50)  [one "." per sample]:
.................................................. 50
> # Print the result
> print(gap_stat, method = "firstmax")
Clustering Gap statistic ["clusGap"] from call:
clusGap(x = Data, FUNcluster = kmeans, K.max = 10, B = 50, nstart = 11)
B=50 simulated reference sets, k = 1..10; spaceH0="scaledPCA"
 --> Number of clusters (method 'firstmax'): 6
           logW     E.logW         gap    SE.sim
 [1,]  2.0511502  1.8756916 -0.17545863 0.1117934
 [2,]  1.5216853  1.4121090 -0.10957631 0.1086127
 [3,]  1.0267224  1.0767012  0.04997878 0.1202571
 [4,]  0.4257883  0.7697560  0.34396769 0.1294819
 [5,]  0.1161890  0.4852060  0.36901700 0.1380535
 [6,] -0.2167312  0.1871733  0.40390445 0.1518605
 [7,] -0.2743961 -0.1405058  0.13389031 0.1775723
 [8,] -1.2008905 -0.5371663  0.66372417 0.1948159
 [9,] -1.7376781 -1.0796498  0.65802830 0.2379769
[10,] -3.1287173 -2.0055315  1.12318583 0.3554866
```

ما می توانیم نتایج را با **fviz_gap_stat** تجسم کنیم که چهار خوشه را به عنوان تعداد بهینه خوشه ها پیشنهاد می کند.

```
> fviz_gap_stat(gap_stat)
```



استخراج نتایج

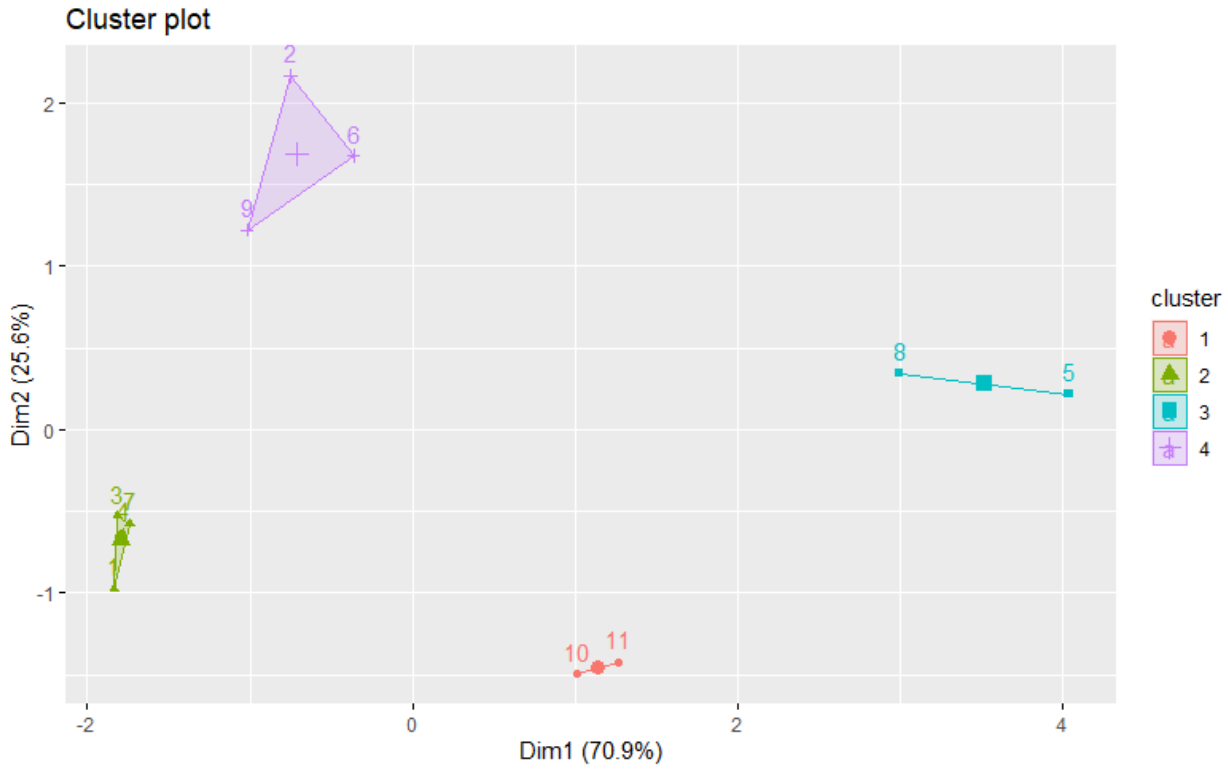
با توجه به اینکه اکثر این رویکردها ۴ را به عنوان تعداد خوشه های بهینه پیشنهاد می کنند، می توانیم تجزیه و تحلیل نهایی را انجام داده و نتایج را با استفاده از ۴ خوشه استخراج کنیم.

```
> set.seed(555)
> final <- kmeans(Data, 4, nstart = 11)
> print(final)
K-means clustering with 4 clusters of sizes 2, 4, 2, 3

Cluster means:
        Open   Momentum      iMA10     iMA100     StdDev        RSI
1  0.7865320 -0.8988999  0.8060680  0.7989351  0.7061987 -0.4619170
2 -0.6955224 -0.7728956 -0.6971629 -0.6962679 -0.7827851 -1.0093140
3  1.6355306  0.8376681  1.6216544  1.6262911  1.4675984  1.2905300
4 -0.6873452  1.0713487 -0.6889310 -0.6884602 -0.4054847  0.7933433

Clustering vector:
 1  2  3  4  5  6  7  8  9 10 11
 2  4  2  2  3  4  2  3  4  1  1

Within cluster sum of squares by cluster:
[1] 0.1249954 0.5269366 1.3290276 1.4028015
 (between_SS / total_SS =  94.4 %)
```

```
Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betwe
enss"    "size"
[8] "iter"         "ifault"
```

ما می توانیم نتایج را با استفاده از fviz_cluster تجسم کنیم

```
> fviz_cluster(final, data = Data)
```



نظرات اضافی

خوشه بندی K-means یک الگوریتم بسیار ساده و سریع است. علاوه بر این، می تواند به طور موثر با مجموعه داده های بسیار بزرگ مقابله کند. با این حال، برخی از نقاط ضعف رویکرد k-means وجود دارد.

یکی از معایب بالقوه خوشه بندی K-means این است که از ما می خواهد تعداد خوشه ها را از قبل مشخص کنیم. خوشه‌بندی سلسله مراتبی یک رویکرد جایگزین است که نیازی به متعهد شدن به انتخاب خاصی از خوشه‌ها ندارد. خوشه‌بندی سلسله مراتبی مزیت بیشتری نسبت به خوشه‌بندی K-means دارد، زیرا منجر به نمایش درختی جذابی از مشاهدات می‌شود که دندروگرام نامیده می‌شود. یک آموزش آینده رویکرد خوشه بندی سلسله مراتبی را نشان خواهد داد.

یکی دیگر از معایب **K-means** این است که به موارد پرت حساس است و اگر ترتیب داده های خود را تغییر دهید، نتایج متفاوتی ممکن است رخ دهد. رویکرد خوشه‌بندی پارتیشن‌بندی حول **Medoids (PAM)** نسبت به موارد دورافتاده حساسیت کمتری دارد و جایگزینی قوی برای **k-means** برای مقابله با این موقعیت‌ها ارائه می‌کند. یک آموزش آینده رویکرد خوشه بندی **PAM** را نشان خواهد داد.