

DSM 5008 DENETİMSİZ İSTATİSTİKSEL ÖĞRENME YARIYIL SONU DEĞERLENDİRME

Simay UĞUR

22 06 2020

Contents

1	Veri Setinin Açıklaması	2
2	Tanımlayıcı İstatistikler	3
2.1	Veri Setinin Düzenlenmesi	5
2.2	Korelasyon Matrislerinin Elde Edilmesi	6
2.3	Veri Setinin Değişiminin Box-Plot ve Histogram Grafiği ile Gösterimi	8
2.4	Box-Plot	9
3	Temel Bileşenler Analizi	9
3.1	Pca Uygulanabilirliğini Ölçme ve Bileşen Sayısına Karar Verme	9
3.2	Görselleştirme	11
3.2.1	Scree Plot	11
4	Kümeleme Analizi İçin Verinin Ön Hazırlığı	15
4.1	Kümelenme Eğiliminin Değerlendirilmesi	16
4.1.1	Hopkins İstatistiği	16
4.1.2	VAT İstatistiği	16
4.1.3	En İyi Küme Algoritması Seçimi	17
5	K-Means Yöntemi	19
6	K-medoids Yöntemi	23
7	Hiyerarsik Kumeleme Analizi	25
7.1	Birlestirici Hiyerarsik Kumeleme	25
7.2	Birlestirici Methodların Karsilastırılması	26
7.3	Bolumleyici Hiyerarsik Kumeleme	28
8	Model Tabanlı Kümeleme	30
9	Yoğunluk Temelli Kümeleme	33
9.1	Küme Geçerliliği	34
10	Finalde Elde Edilen Kümelerin Tanımlayıcı İstatistikleri Ve Yorumlanması	36
10.1	Verinin Son Hali	37

KÜTÜPHANELER

Amacımız ID değerlerinin kümelenmesidir.

Veri setinin diagnosis sutununda kanser təşhisini koyulan kişilərin tümörlerinin huyu belirtilmişdir.

Üçüncü sütundan on ikinci sütuna kadar olan değerler ise aşağıda belirtilen değerlerin ortalamasıdır.

radius_mean: merkezden noktaya olan uzaklıklar

texture_mean: gri tonlamalı değerlerin standart sapmaları

perimeter_mean: tümörün çevre ortalaması

area_mean: tümörün ortalama alanı

smoothness_mean: yarıçap uzunluklarındaki yerel değişim

compactness_mean: en küçük kanser hücreleri (perimeter^2 / area - 1.0)

concavity_mean: kontürün içbükey kısımlarının şiddeti

concave.points_mean: kontürün içbükey kısımlarının sayısı

symmetry_mean: simetri ortaması

fractal_dimension_mean: fraktal boyutu

1 Veri Setinin Açıklaması

Meme kanseri, meme dokusundan gelişen kanserdir ve dünya çapında kadınlar arasında en yaygın kanserdir. Rutin meme kanseri taraması hastalığın təşhis edilmesine ve tedavi edilmesinden önce belirgin semptomlara neden olmasına izin verir.

Bir makine öğrenimi, kanserin təşhis sürecini otomatikləştirebilirse, doktorların hastalığı erken evrede tanı veya tedavi etmek için daha fazla zaman yaratabilmesini sağlar.

UCI web sitesinde belirtildiği gibi, “Özellikler, bir göğüs kitlesinin ince iğne aspiratının sayısallaştırılmış görüntüsüyle hesaplanır. Görüntüde bulunan hücre çekirdeklerinin özelliklerini tanımlarlar”.

Ayrıca FNA, BT taraması veya ultrason monitörleri kılavuzu ile anormal doku veya hücreler alanına çok ince bir iğnenin sokulduğu bir biyopsi prosedürü türündür.

Klinisyen Kitlenin malign veya benign (kötü huylu iyi huylu tümörler) olup olmadığını belirlemek için memeden küçük bir hücre örneği çıkarır ve hücreleri mikroskop altında inceler.

Meme kanseri tümörlerinin iyi veya kötü huylu olup olmadıklarına içeren veri seti 569 gözlem ve 11 değişkenden oluşan şekilde tekrar düzeltildi.

```
data1<- read.csv(file="wdbc.csv",sep=",",header = TRUE, row.names = 1)
```

```
data1<- data1[,1:11]
```

```
dim(data1)
```

```
## [1] 569 11
```

```
head(data1)
```

```
##      diagnosis radius_mean texture_mean perimeter_mean area_mean
## 1  M           2.34        10.42       17.92      106.50
## 2  M           2.44        10.43       18.75      153.85
## 3  M           2.38        10.53       18.55      158.85
## 4  M           2.36        10.58       18.46      151.30
## 5  M           2.42        10.49       18.32      162.70
## 6  M           2.35        10.51       18.56      153.45
```

```

## 843786          M      12.45      15.70      82.57      477.1
##           smoothness_mean compactness_mean concavity_mean concave.points_mean
## 842302          0.11840      0.27760      0.3001      0.14710
## 842517          0.08474      0.07864      0.0869      0.07017
## 84300903        0.10960      0.15990      0.1974      0.12790
## 84348301        0.14250      0.28390      0.2414      0.10520
## 84358402        0.10030      0.13280      0.1980      0.10430
## 843786          0.12780      0.17000      0.1578      0.08089
##           symmetry_mean fractal_dimension_mean
## 842302          0.2419       0.07871
## 842517          0.1812       0.05667
## 84300903        0.2069       0.05999
## 84348301        0.2597       0.09744
## 84358402        0.1809       0.05883
## 843786          0.2087       0.07613

```

2 Tanımlayıcı İstatistikler

Aşağıda numerik değişkenlerin tanımlayıcı istatistiklerinin değerleri görülmektedir.

Profiling_num yöntemi ile sayısal verilerin ortalamaları standart sapmaları çeyreklik bilgileri medyan çarpıklık ve aralık değerleri hakkında bilgi sahibi olabiliriz.

Değişkenlerdeki değişkenlik oldukça fazladır. area_mean ile compacteness_mean arasındaki fark çok büyktür. Bu durum Temel Bileşen Analizi işlemi uygulanırken ve diğer kümeleme analizlerinde sorun oluşturabileceğinin ileri ki analizlerde değişkenliğin sabitlenmesi gereklidir yani ileri ki analizlerde **normalleştirme işlemi** uygulanacaktır.

```

library(funModeling)
profiling_num(data1)

```

```

##           variable      mean      std_dev variation_coef      p_01
## 1      radius_mean 14.12729174 3.524049e+00 0.2494497 8.4583600
## 2      texture_mean 19.28964851 4.301036e+00 0.2229712 10.9304000
## 3      perimeter_mean 91.96903339 2.429898e+01 0.2642083 53.8276000
## 4      area_mean 654.88910369 3.519141e+02 0.5373645 215.6640000
## 5      smoothness_mean 0.09636028 1.406413e-02 0.1459536 0.0686540
## 6      compactness_mean 0.10434098 5.281276e-02 0.5061555 0.0333508
## 7      concavity_mean 0.08879932 7.971981e-02 0.8977525 0.0000000
## 8      concave.points_mean 0.04891915 3.880284e-02 0.7932036 0.0000000
## 9      symmetry_mean 0.18116186 2.741428e-02 0.1513248 0.1295080
## 10     fractal_dimension_mean 0.06279761 7.060363e-03 0.1124304 0.0515040
##           p_05      p_25      p_50      p_75      p_95      p_99      skewness
## 1 9.5292e+00 11.70000 13.37000 15.78000 20.57600 2.43716e+01 0.9398934
## 2 1.3088e+01 16.17000 18.84000 21.80000 27.15000 3.06520e+01 0.6487336
## 3 6.0496e+01 75.17000 86.24000 104.10000 135.82000 1.65724e+02 0.9880370
## 4 2.7578e+02 420.30000 551.10000 782.70000 1309.80000 1.78660e+03 1.6413905
## 5 7.5042e-02 0.08637 0.09587 0.10530 0.11878 1.32888e-01 0.4551199
## 6 4.0660e-02 0.06492 0.09263 0.13040 0.20870 2.77192e-01 1.1869833
## 7 4.9826e-03 0.02956 0.06154 0.13070 0.24302 3.51688e-01 1.3974832
## 8 5.6208e-03 0.02031 0.03350 0.07400 0.12574 1.64208e-01 1.1680903
## 9 1.4150e-01 0.16190 0.17920 0.19570 0.23072 2.59564e-01 0.7236947
## 10 5.3926e-02 0.05770 0.06154 0.06612 0.07609 8.54376e-02 1.3010474
##           kurtosis      iqr      range_98      range_80
## 1 3.827584 4.08000 [8.45836, 24.3716] [10.26, 19.53]

```

```

## 2 3.741145 5.63000      [10.9304, 30.652]      [14.078, 24.992]
## 3 3.953165 28.93000     [53.8276, 165.724]     [65.83, 129.1]
## 4 6.609761 362.40000    [215.664, 1786.6]     [321.6, 1177.4]
## 5 3.837945 0.01893      [0.068654, 0.132888]  [0.079654, 0.11482]
## 6 4.625140 0.06548      [0.0333508, 0.277192]  [0.0497, 0.17546]
## 7 4.970592 0.10114      [0, 0.351688]          [0.013686, 0.20304]
## 8 4.046680 0.05369      [0, 0.164208]          [0.011158, 0.10042]
## 9 4.266117 0.03380      [0.129508, 0.259564]  [0.14958, 0.21494]
## 10 5.969017 0.00842     [0.051504, 0.0854376000000001] [0.055338, 0.072266]

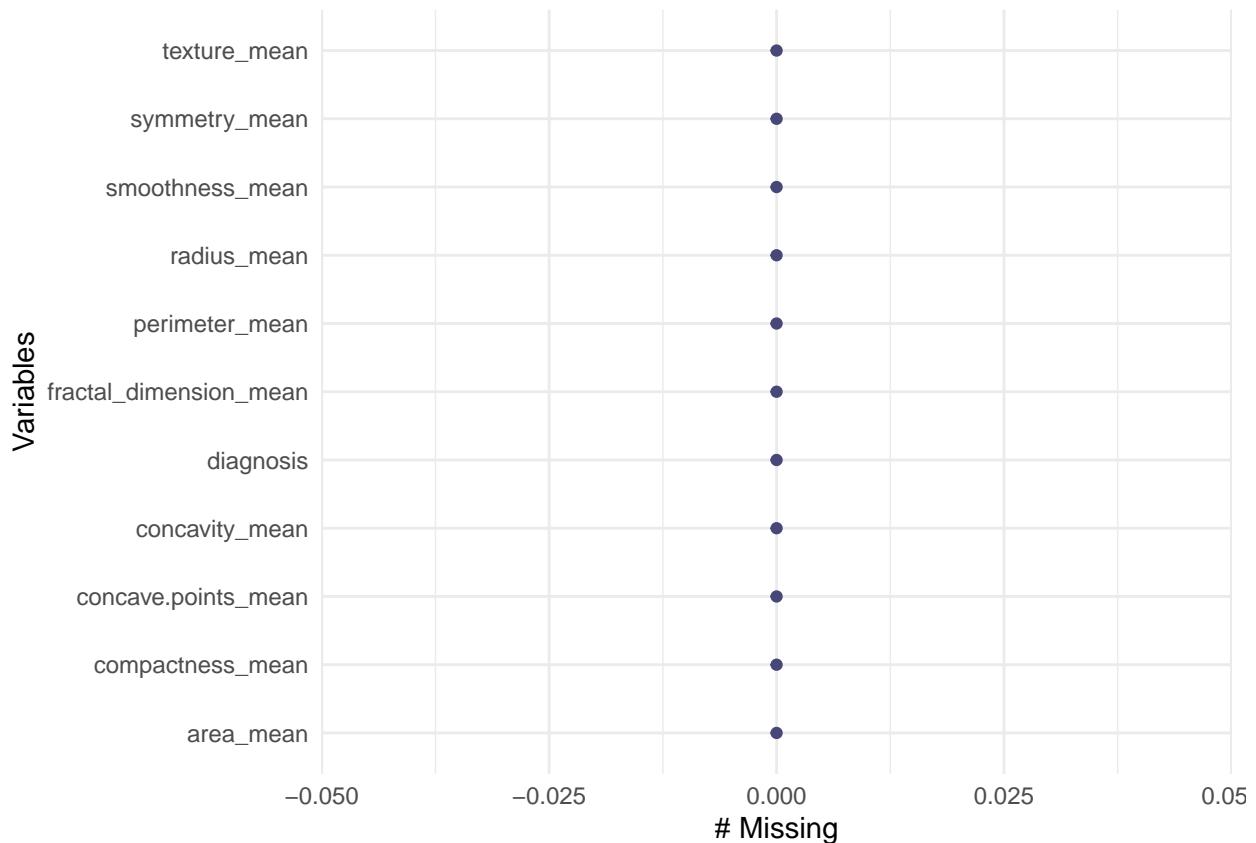
```

Hiçbir değişkende NA değerleri bulunmamaktadır.

```

library(naniar)
gg_miss_var(data1)

```



Veri setinde toplamda 357 iyi huylu tümör verisi, 212 tane kötü huylu tümör verisi bulunmaktadır.

```

mytable<-with(data1,table(data1$diagnosis))
mytable

```

```

##
##      B      M
## 357 212

```

M = Malign (Kötü Huylu kanser hücrelerinin varlığını gösterir); B = İyi huylu kanser hücresi (yokluğu gösterir)

357 gözlem den oluşan benign iyi huylu tümörler tüm gözlemlerin % 62,7'sini oluşturur. Bütün gözlemlerin % 37,3'ünü oluşturan 212 gözlemin kötü huylu kanserli hücreleri vardır.

Yüzde alışılmadık derecede büyük; veri seti bu durumda tipik bir tıbbi analiz dağılımını temsil etmez. Tipik olarak, pozitif (malign) tümörü temsil eden az sayıda vakaya karşı negatif temsil eden çok sayıda vaka olacaktır.



2.1 Veri Setinin Düzenlenmesi

```
data<-data1[,-1]
head(data)

##          radius_mean texture_mean perimeter_mean area_mean smoothness_mean
## 842302      17.99      10.38      122.80    1001.0      0.11840
## 842517      20.57      17.77      132.90    1326.0      0.08474
## 84300903    19.69      21.25      130.00    1203.0      0.10960
## 84348301    11.42      20.38      77.58     386.1      0.14250
## 84358402    20.29      14.34      135.10    1297.0      0.10030
## 843786      12.45      15.70      82.57     477.1      0.12780
##          compactness_mean concavity_mean concave.points_mean symmetry_mean
## 842302      0.27760      0.3001      0.14710      0.2419
## 842517      0.07864      0.0869      0.07017      0.1812
## 84300903    0.15990      0.1974      0.12790      0.2069
## 84348301    0.28390      0.2414      0.10520      0.2597
## 84358402    0.13280      0.1980      0.10430      0.1809
## 843786      0.17000      0.1578      0.08089      0.2087
##          fractal_dimension_mean
## 842302      0.07871
## 842517      0.05667
```

```

## 84300903      0.05999
## 84348301      0.09744
## 84358402      0.05883
## 843786        0.07613

```

2.2 Korelasyon Matrislerinin Elde Edilmesi

Gelişmiş scatter ile değişkenlerin dağılımları ve ilişkileri; korelasyon matris plot ile değişkenlerin ilişkileri hakkında bilgi sahibi olabiliriz.

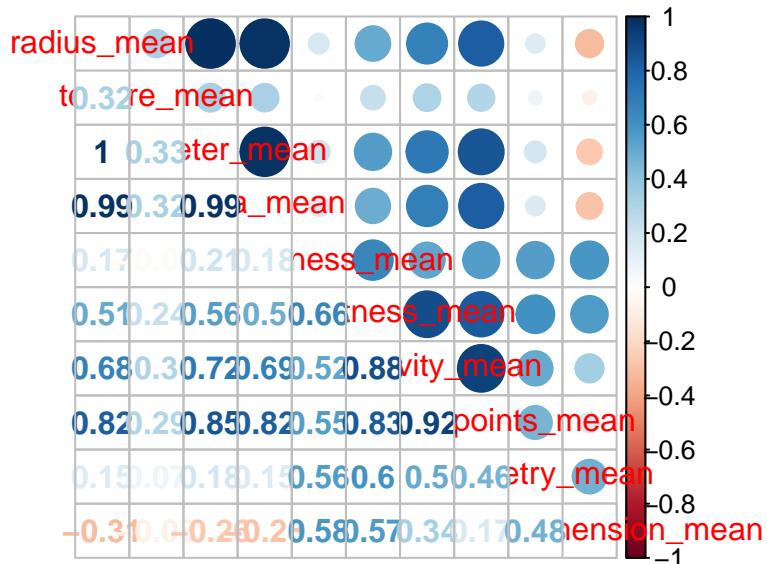
-fractal_dimension_mean (fraktal boyutu) değişkeni ile tümörün ortalama alan değişkeni(area_mean), texture_mean, radius_mean ve perimeter_mean(tümörün ortalama çevresi) değişkenleri ters yönde ilişkilidir ve bu ilişkilerin gücü oldukça düşüktür. *Yani, fraktal boyutu arttığında (azaldığında), merkezden noktaya olan uzaklıklar, gri tonlamalı değerlerin standart sapmaları, hücre çekirdeğinin çevresi ve hücre çekirdeğinin alanları azalacaktır (artacaktır).

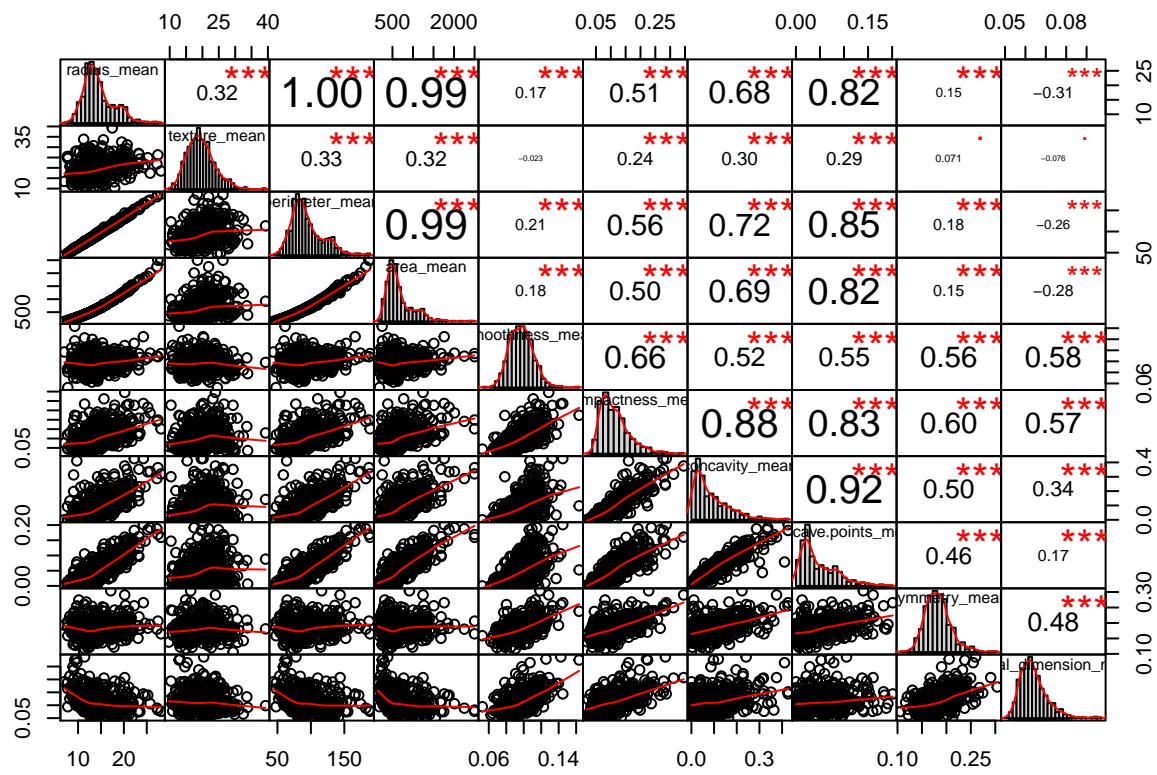
- Perimeter_mean(tümörlerin çevresi) ile area_mean (tümörlerin alanı) arasında pozitif yönlü **çok güclü ilişki** bulunmaktadır.

-Tümörün yarıçap ortalaması(radius_mean) değişkeni ile Perimeter_mean (tümörlerin çevresi) arasında pozitif yönlü **tam ilişki** bulunmaktadır.

*Hücre çekirdeğinin alanı arttığında (azaltığında), tümörün yarıçapı, tümörün ortalama çevresi, kontürün içbükey kısımlarının şiddeti ve gri tonlamalı değerlerin standart sapmaları artacaktır, (azalacaktır).

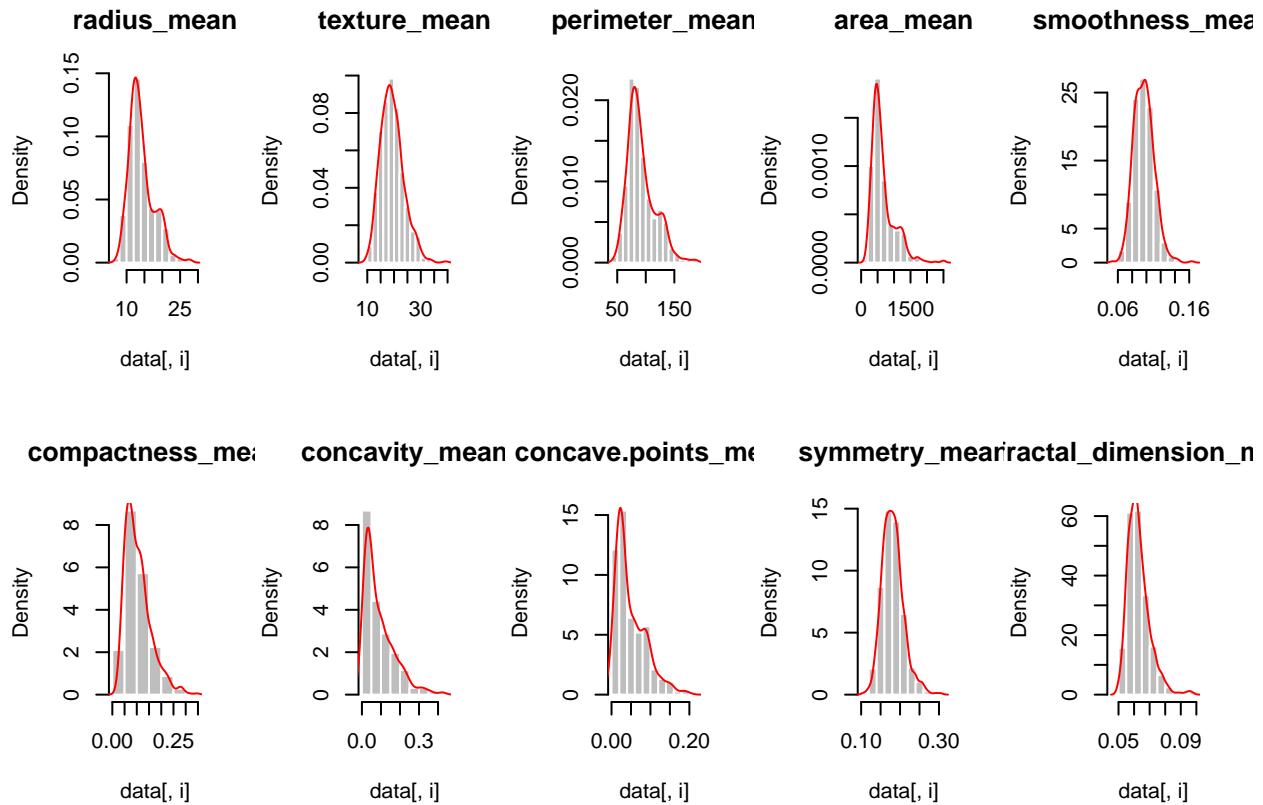
- concave.points_mean değişkeni ile concavity_mean **pozitif yönlü çok güclü ilişkiye** sahip oldukları görülmektedir.
- concavity_mean ile smoothness_mean değişkenleri arasında **pozitif yönlü Orta derecede ilişki** vardır.





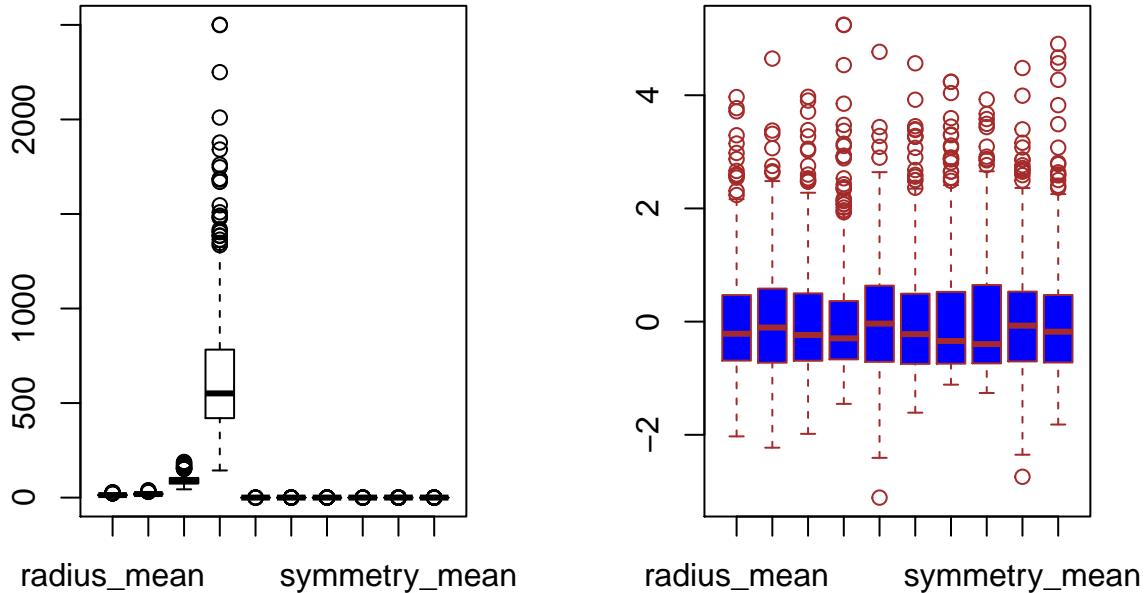
2.3 Veri Setinin Değişiminin Box-Plot ve Histogram Grafiği ile Gösterimi

-concavity_mean,area_mean,concave.points_mean,fractal_dimension_mean değişkenlerinin veri seti içindeki dağılımı sağdan çarpiktır. -symmetry_mean,smoothness_mean değişkenleri normal dağılmıştır.



2.4 Box-Plot

Veriyi ölçeklendirerek daha güzel bir box plot elde edildi. Her değişkenin aykırı gözlemleri vardır.



3 Temel Bileşenler Analizi

Temel bileşenler analizinin ana fikri çok değişkenli verinin ana özelliklerini az sayıda değişken ile temsile etmektir. Diğer bir ifade ile küçük bir bilgi kaybını göze alıp değişken boyutunu azaltmaktadır. Oluşabilecek bilgi kaybının görece hata ve gürültü ile kıyaslanabilir düzeyde küçük olması beklenir.

Temel bileşenler yaklaşımı bağımlılık yapısını yok etme ve boyut indirgeme amaçları için kullanılmaktadır. Tanıma, sınıflandırma, boyut indirgenmesi ve yorumlanması sağlayan, çok değişkenli bir istatistik yöntemdir.

PCA Verinin içindeki en güçlü örüntüyü bulmaya çalışır. Verideki gürültüler, örüntülerden daha gücsüz olduklarından, boyut küçültme sonucunda bu gürültüler temizlenir.

PCA ilk adım olarak kovaryans/korealasyon matris hesabı gerektirir. Varyans matris, özdeğerler ve özvektörlerin elde edilmesi için kullanılmaktadır. Örneğin Veri iki boyutlu olduğundan kovaryans matris de 2x2 boyutlu olacaktır.

Teknik olarak PCA :

- Değişken gruplarının varyanslarını ifade eden öz değerler ile veri setindeki değişkenleri gruplandırır.
- Gruplar arasında en fazla varyansa sahip gruplar en önemli gruplardır kibunlar asal bileşenlerdir.

3.1 Pca Uygulanabilirliğini Ölçme ve Bileşen Sayısına Karar Verme

Kaiser, Meyer, Olkin ölçümünü hesaplayarak korelasyonların oldukça yüksek olduğu kesinleştirilebilir.

Veri setinin korelasyon matrisinin KMO değerine baktığımızda $0.79 > 0.5$ olduğu görülmektedir. 0.79 değeri veri setinin pca(temel bileşenler analizi) için uygun olduğunu göstermektedir.

```
corr=cor(data, method = "pearson")
psych:: KMO(corr)

## Kaiser-Meyer-Olkin factor adequacy
## Call: psych::KMO(r = corr)
## Overall MSA = 0.79
## MSA for each item =
##          radius_mean      texture_mean      perimeter_mean
##                0.72            0.93            0.70
##          area_mean      smoothness_mean      compactness_mean
##                0.87            0.79            0.69
##          concavity_mean concave.points_mean      symmetry_mean
##                0.88            0.88            0.95
## fractal_dimension_mean
##                0.68
```

Ozdeğerler ve Ozdeğer Vektörlerinin Oluşturulması (eigenvalues & eigenvectors):

R da özdeğer vektörleri negatif yönde oluşturulur. Eksen döndürme işlemi yapıldı. ($p < -p$). Her temel bileşen vektörü özellik uzayında bir yön tanımlar. Özdeğer vektörler birleşenleri birbirile korelasyonsuzdur. Bileşenlerin değişkenlerden etkilenme durumunu gösterir.

```
cov_df <- cov(dt)
ei_df <- eigen(cov_df)

p <- ei_df$vectors[,1:2]
p <- -p
```

Temel Bileşen Sayısını Bulma

```
pca <- prcomp(data, scale = TRUE, center = TRUE)
eigen(cor(data))$ values

## [1] 5.4785879917 2.5187135854 0.8806151792 0.4990094357 0.3725391897
## [6] 0.1241417485 0.0800853104 0.0348897928 0.0111354606 0.0002823059

pca_result<-data.frame(predict(pca))
head(pca_result)
```

	PC1	PC2	PC3	PC4	PC5	PC6
## 842302	-5.219562	-3.2016111	-2.1694307	-0.1691271	1.5129208	-0.11302355
## 842517	-1.726575	2.5386054	-1.0187821	0.5470581	0.3120551	0.93481161
## 84300903	-3.966267	0.5495913	-0.3232843	0.3976143	-0.3225932	-0.27125473
## 84348301	-3.593551	-6.8989994	0.7921346	-0.6042963	0.2429625	0.61642725
## 84358402	-3.148321	1.3568784	-1.8605969	-0.1850886	0.3110679	-0.09069778
## 843786	-1.380105	-3.3114977	-0.6973879	-0.4723685	-0.5006185	-0.16262179
	PC7	PC8	PC9	PC10		
## 842302	-0.34438133	-0.231727880	0.02196273	-0.011247764		
## 842517	0.42055208	-0.008335534	0.05612189	-0.022971998		
## 84300903	0.07643917	-0.354737945	-0.02009818	-0.022654878		
## 84348301	-0.06799091	-0.100074939	0.04344302	-0.053409301		
## 84358402	0.30781641	0.098969999	0.02655061	0.034082655		
## 843786	0.06429129	-0.099994918	0.04840093	-0.006434524		

Bileşenlerin Açıklanabilen varyans oranını bulalım.

İlk temel bileşen %55, ikinci temel bileşen %25 ve üçüncü temel bileşen ise %0.09'unu açıklamaktadır.

İlk iki bileşen veri setindeki varyansın %80'ini açıklamaktadır.

```
avo<-ei_df$values/sum(ei_df$values)
round(avo,2)

## [1] 0.55 0.25 0.09 0.05 0.04 0.01 0.01 0.00 0.00 0.00
```

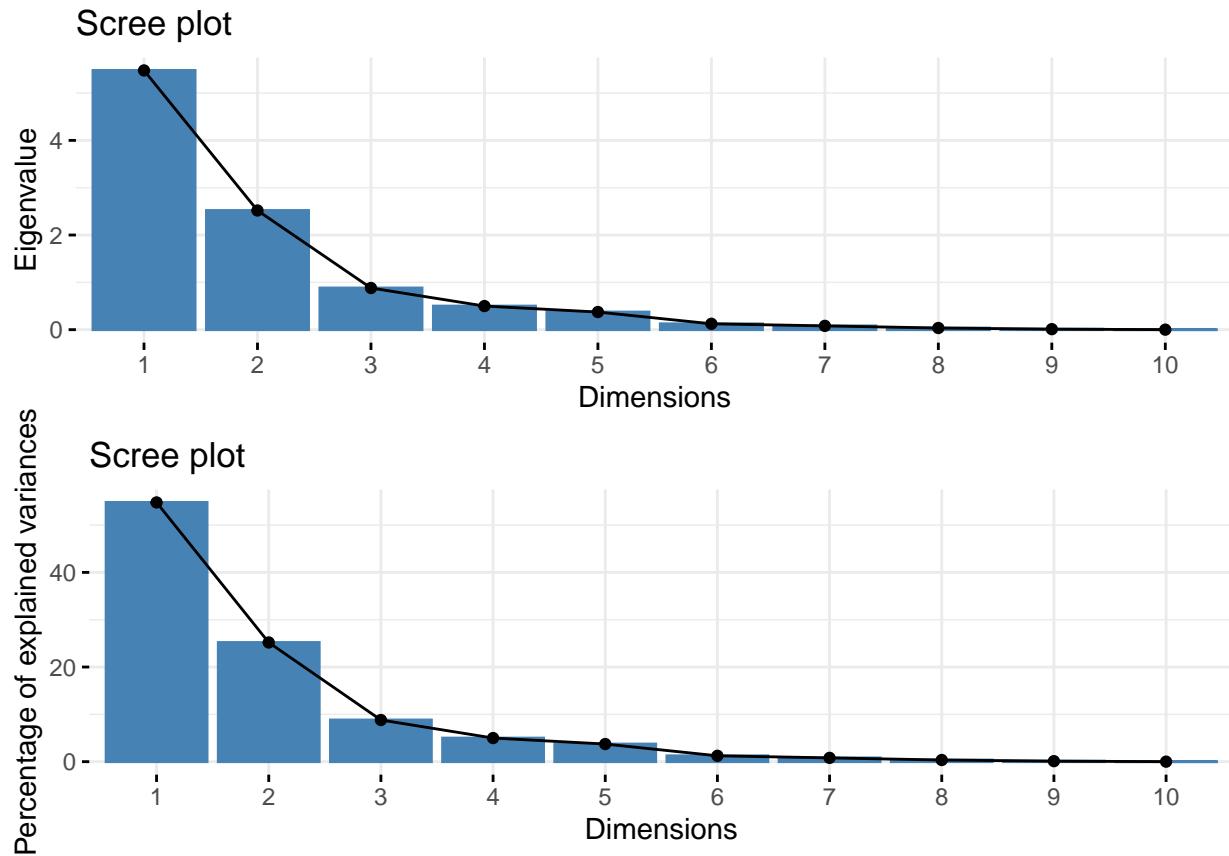
3.2 Görselleştirme

3.2.1 Scree Plot

Açıklanan varyans oranları görsel olarak gösterelmiştir. İlk görselde Eigen değerleri 1'den büyük olan ilk iki bileşen vardır. Bu durumda ilk iki bileşen seçilmesi uygunudur.

2. görselde her bir bileşenin verideki açıklanabilen varyans yüzdelğini görmekteyiz. 3. bileşenden sonra eğim azalmıştır. 3. bileşen seçilebilir.

```
p1<-fviz_eig(pca,choice='eigenvalue')
p2<-fviz_eig(pca)
grid.arrange(p1,p2)
```



Scree plotlarda eigen value ya baktığımızda 2 temel bileşen seçilebilir üçüncü temel bileşen de alınabilir. 2 temel bileşenin toplam açıklanan varyans %79.97'di.

Bileşenlerin önem derecesini summary fonksiyonu ile incelediğimizde PC1 temel bileşeni %54.79, ikinci temel bileşen (PC2)'%25.19'sini ve üçüncü temel bileşen ise %0.88'ini açıklamaktadır. 3 temel bileşenle toplam açıklanan varyans %88.78'dir. 2 temel bileşen seçilerek verinin açıklanan toplam varyansı %79.97'dir.

```

summary(pca)

## Importance of components:
##                               PC1      PC2      PC3      PC4      PC5      PC6      PC7
## Standard deviation    2.3406 1.5870 0.93841 0.7064 0.61036 0.35234 0.28299
## Proportion of Variance 0.5479 0.2519 0.08806 0.0499 0.03725 0.01241 0.00801
## Cumulative Proportion  0.5479 0.7997 0.88779 0.9377 0.97495 0.98736 0.99537
##                               PC8      PC9      PC10
## Standard deviation     0.18679 0.10552 0.01680
## Proportion of Variance 0.00349 0.00111 0.00003
## Cumulative Proportion  0.99886 0.99997 1.00000
pca$rotation[,c(1,2)]

```

```

##                               PC1      PC2
## radius_mean      -0.36393793 0.313929073
## texture_mean     -0.15445113 0.147180909
## perimeter_mean   -0.37604434 0.284657885
## area_mean        -0.36408585 0.304841714
## smoothness_mean  -0.23248053 -0.401962324
## compactness_mean -0.36444206 -0.266013147
## concavity_mean   -0.39574849 -0.104285968
## concave.points_mean -0.41803840 -0.007183605
## symmetry_mean    -0.21523797 -0.368300910
## fractal_dimension_mean -0.07183744 -0.571767700

```

Birinci temel bilesen, verideki degiskenligi en çok açıklayan bilesendir.

Birinci temel bilesende en belirgin katkısı olan değişkenler :

-concavity_mean -compactness_mean -concave.points_mean -perimeter_mean -area_mean -radius_mean

ikinci temel bilesende en belirgin katkısı olan değişkenler :

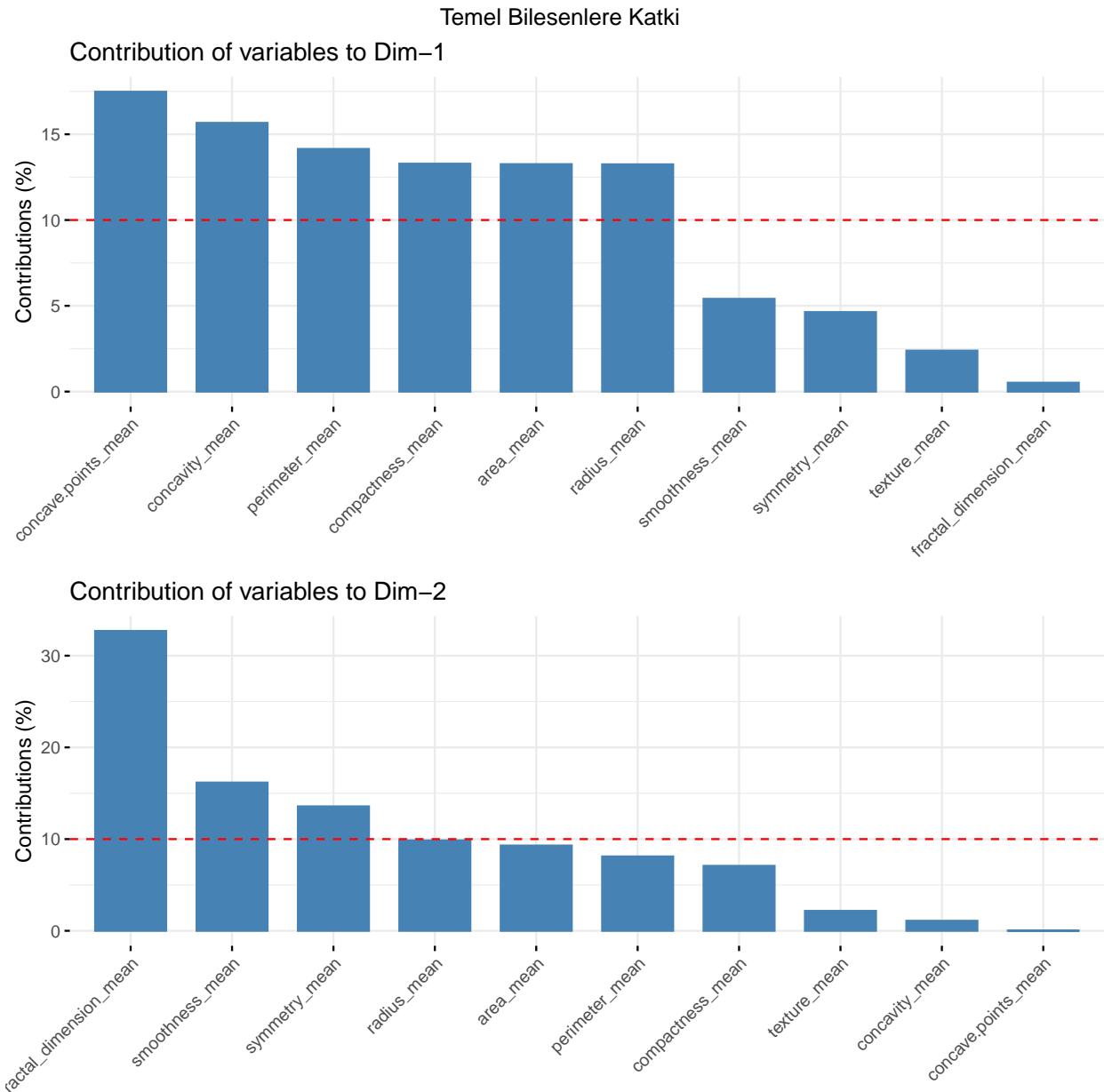
-fractal_dimension_mean -smoothness_mean

-symmetry_mean

```
var <- get_pca_var(pca)
```

```
a<-fviz_contrib(pca, "var",axes = 1)
b<-fviz_contrib(pca, "var",axes = 2)
```

```
grid.arrange(a,b,top='Temel Bileşenlere Katkı')
```



smoothness_mean ile texture_mean arasında 90 derecelik açı birbiriyle zıt ilişkide olduğunu gösterir.

parameter_mean ,radius mean ve area_mean çakışık olması onların arasında pozitif ilişki olduğunu gösterir. Bir tümörün hücresinin alanı arttıkça yarıçapıda çevreside artar.

Bütün veriler merkez çevresinde toplanmış olarak görülmektedir.

ID'leri: 842302, 84300903, 8910988, 89812, 8820612 olan gözlemler PCA_1'i açıklayan concavity_mean, perimeter_mean, concave_p
erimeter_mean, area_mean değişkenleri açısından yüksek değerdedir.

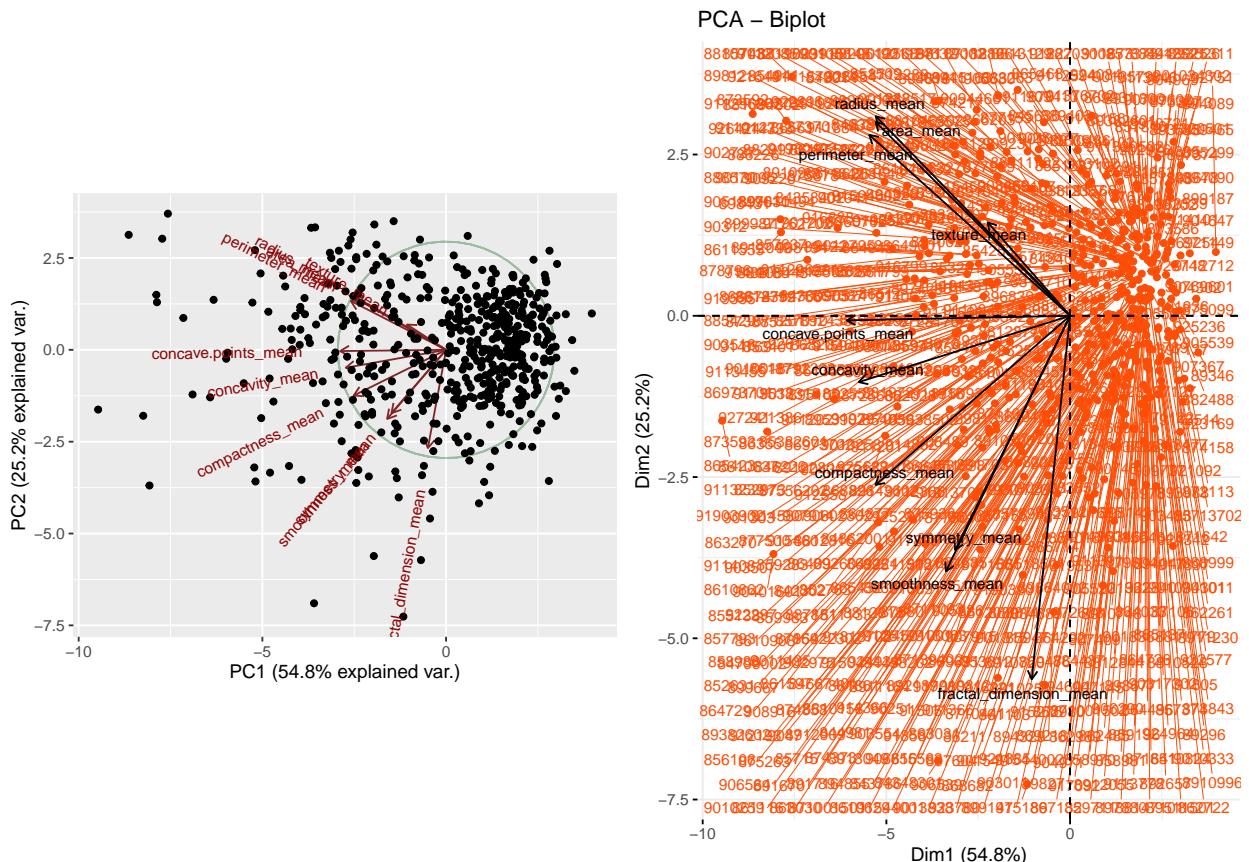
ID'leri : 873592, 8611555, 8810703, 911296202 ve 899987 olan gözlemler PCA_2'yi açıklayan fraktal boyutu
ortalaması, simetri ortalaması ,smoothness_mean değişkenleri açısından yüksek değerdedir.

Temel bileşen biri(PCA1) açıklayan değişkenler (perimeter_mean, radius_mean, area_mean, compactness_mean, concavity_mean ve concave.points_mean) açısından düşük değer alan ve PCA2'yi açıklayan
değişkenler (smoothness_mean, symmetry_mean ve fractal_dimension_mean) açısından da düşük değerli
olan ID'ler; 865423, 86355, 8610862, 84348301, 815186, 8710441, 866714 ve 955186 ID'leridir.

ID'leri 8610862,915186,88110703,863555,84348301 911296202 ve 865423 olan gözlemler **aykırı gözlemlerdir**. Diğer gözlemler koordinat ekseniinde 0'a oldukça yakın konumlanmışlardır.

```
#install_github("vqv/ggbiplot")
library(ggbiplot)
g<-ggbiplot(pca, obs.scale=1, var.scale=1, circle=TRUE)
b<-fviz_pca_biplot(pca, repel = TRUE,
                     col.var = "#000000", # Variables color
                     col.ind = "#FC4E07" # Individuals color
                     ,labelsize=3)

grid.arrange(g,b,nrow=1)
```



4 Kümeleme Analizi İçin Verinin Ön Hazırlığı

Kümeleme aynı küme içerisindeki gözlemlerin birbirine benzer, diğer kümelerdeki gözlemlerden farklı olacak şekilde yapılmalıdır. Benzerlik ve farklılık ölçümleri gözlemlerin birbirinden ayırt edilmesini sağlar ve bu sayede gözlemler gruplara ayrılır. Gözlemlerin (Bireylerin) benzerliğini belirlemek için birbirleri arasındaki uzaklıklar esas alınmaktadır.

Uzaklığın bir benzerlik ölçüyü olarak kullanıldığı durumlarda gözlenen bireyler arasındaki uzaklıklar hesaplanır ve uygulanan kümeleme tekniğine göre bireyler uygun kümelere atanır.

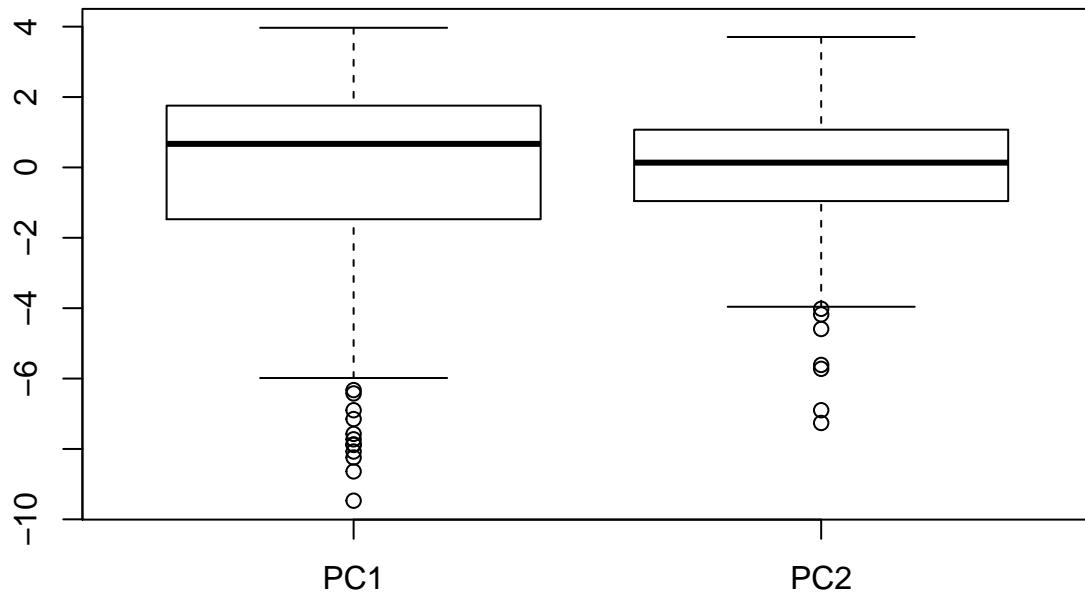
Nicel kümeleme yapmak istediği durumlarda öklid, manhattan, minkovski, Ölçekli Öklit Uzaklığı, Mahalanobis Uzaklığı, Hotelling T 2 Uzaklığı ve Canberra Uzaklığı kullanılmaktadır.

```
pca_data<-pca_result[,c(1,2)]  
head(pca_data)
```

```
##          PC1         PC2  
## 842302 -5.219562 -3.2016111  
## 842517 -1.726575  2.5386054  
## 84300903 -3.966267  0.5495913  
## 84348301 -3.593551 -6.8989994  
## 84358402 -3.148321  1.3568784  
## 843786 -1.380105 -3.3114977
```

Temel bileşen analizi uygulanmış veri setinin Box plot ile incelemesini yapalım. Değişkenlik ortadan kalkmış bulunmaktadır. Aykırı gözlemler bulunmakta ve veriseti çarpaktır.

```
boxplot(pca_data)
```



Kümeleme analizine geçmeden önce gözlem noktalarının birbirine olan uzaklığını hesaplanmalıdır. Veri kümemizdeki bütün degiskenlerde aykırı değer olduğu için bu aykırılıktan en az etkilenenek olan manhattan distance ölçüsü kullanılmalıdır. Scale edilen veriler de bu ölçütler arasında büyük farklılıklar yoktur.

Mavi renkte olan hücreler bize en uzak olan, yukarıda belirtilmiş olan ID'lerdir. Birbirlerine en benzer olan gözlemler kırmızı birbirine en yakın uzaklıktadır, yaklaşık 0 değerlerini almıştır.

```
dist_man=dist(pca_data, method="manhattan")
```

4.1 Kümelenme Eğiliminin Değerlendirilmesi

4.1.1 Hopkins İstatistiği

Hopkins istatistiği, belirli bir veri setinin tekdüze dağılımdan üretilme olasılığını ölçerek veri kümelerinin kümelenme eğilimini değerlendirmek için kullanılır.

Paydada yer alan iki toplam birbirine çok yakın ise H istatistiğinin değeri 0.5 olacaktır. Hopkins istatistiğinin 0'a yakın çıkması durumunda H_0 rededilir. Bu da veri setinin önemli ölçüde kümelenebilir bir veri olduğunu gösterir.

sub_veri= veri setinden alınan veri

H_0 = sub_veri uniform dağılıma uyar.

H_1 = sub_veri uniform dağılıma uymaz.

```
library(clustertend)
set.seed(123)
h_data=hopkins(pca_data, nrow(pca_data)-1)
h_data

## $H
## [1] 0.219282
```

$0.219282 < 0.5$, 0.219282 0 yakın olduğu için H_0 reddedilir. Veri seti uniform dağılmaz. Bu da önemli ölçüde kümelenebilir bir veri olduğunu gösterir. Veri Kümelemeye uygundur.

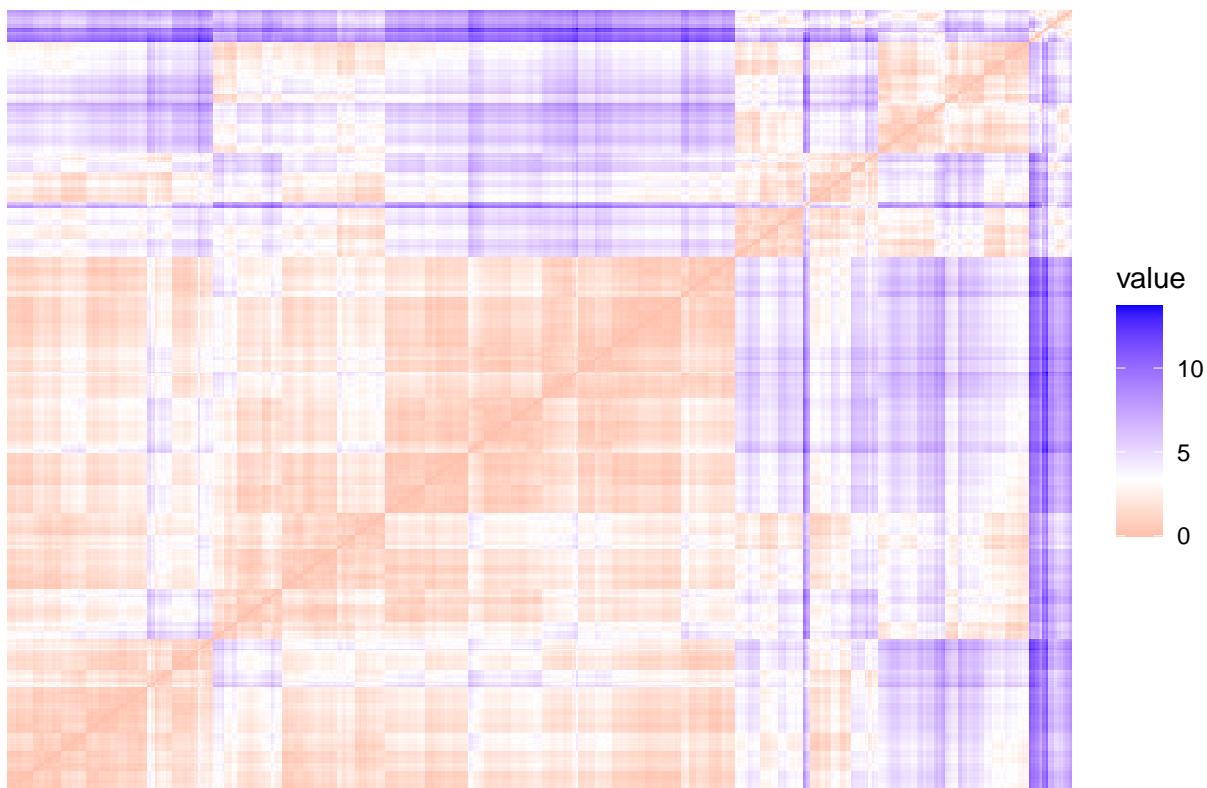
4.1.2 VAT İstatistiği

Kümelenme Eğiliminin görsel olarak değerlendirilmesidir.

Aynı kümeye ait nesneler ardışık sıraya göre sıralanır. Sol alt köşede çok net kümelenme gözükmürken sağa doğru sağ alt köşede kümelenme belirgin değildir

```
c<-fviz_dist(dist(pca_data), show_labels = FALSE )+
  labs(title = "Meme Kanseri Verisi")
c
```

Meme Kanseri Verisi



4.1.3 En İyi Küme Algoritması Seçimi

Hiyerarşik kümeleme yöntemleri ve Hiyerarşik olmayan kümeleme yöntemlerinde hangi kümeleme yönteminin daha iyi sonuç verdiği ölçmek için `clValid` komutu kullanılır.

Connectivityi sıfıra yakın olsun isteriz. Down ve silhouette değeri 1'e en yakın olanı seçeriz.

-connectivity score değeri 14.9956 ile hiyerarşik kümeleme yöntemi ve 2 küme seçme kararı verilebilir. -dunn index scoru için 0.0781 değeri bulunan hiyerarşik yöntemi ve 3 kümeleme, -silhouette yöntemi için 0.5068 değeri bulunan hiyerarşik yöntemi ve 2 küme seçme kararı verilebilir.

```
set.seed(123)
clmethods <- c("kmeans", "pam", "hierarchical")
internal <- clValid(pca_data, nClust = 2:5,
                     validation = "internal")
summary(internal)

##
## Clustering Methods:
## hierarchical
##
## Cluster sizes:
##  2 3 4 5
##
## Validation Measures:
##                               2      3      4      5
## hierarchical Connectivity 10.0647 17.1679 20.6960 38.8679
```

```

##           Dunn      0.0637  0.0719  0.0719  0.0294
##           Silhouette 0.5363  0.4703  0.4538  0.4200
##
## Optimal Scores:
##
##           Score  Method  Clusters
## Connectivity 10.0647 hierarchical 2
## Dunn         0.0719 hierarchical 3
## Silhouette   0.5363 hierarchical 2

```

APN, ADM ve FOM 0 ile 1 arasında değişir. Küçük değerlerde çıkması yüksek tutarlılıkta kümelenme olduğunun göstergesidir. AD 0 ile sonsuz arasında değer alır. Yine küçük değere sahip olması tercih edilir.

-APN skoru 0.0166 değeri ile hiyerarşik yöntemlerde 2 küme seçilebilir. -AD değeri için 2.5069 değeri ile kmeans kümele yöntemi 6 küme, -ADM değeri için 0.3923 değeri ile hiyerarşik yöntemi 2 küme, -FOM değeri için 1.9316 değeri ile clara yöntemi 6 küme optimum küme sayısı olarak görülebilir.

```

set.seed(123)

clmethods <- c("kortalamalar", "pam", "hierarchical", "clara")
sta <- clValid(pca_data, nClust = 2:6, clMethods = clmethods,
                validation = "stability")
summary(sta)

##
## Clustering Methods:
##   pam hierarchical clara
##
## Cluster sizes:
##   2 3 4 5 6
##
## Validation Measures:
##                               2      3      4      5      6
##
## pam          APN  0.2613  0.4383  0.4923  0.5425  0.5709
##          AD   2.9120  2.7748  2.6784  2.5672  2.5069
##          ADM  1.2107  1.5969  1.7997  1.7190  1.7089
##          FOM  1.9655  1.9506  1.9448  1.9359  1.9323
## hierarchical APN  0.0166  0.3886  0.5466  0.5554  0.5571
##          AD   3.2805  3.2119  3.1907  2.8659  2.8581
##          ADM  0.3923  1.3657  1.5427  1.6180  1.6150
##          FOM  1.9625  1.9630  1.9548  1.9365  1.9374
## clara        APN  0.2947  0.4334  0.4676  0.5190  0.5741
##          AD   2.9514  2.7700  2.6597  2.5866  2.5362
##          ADM  1.3076  1.5700  1.7148  1.6209  1.6803
##          FOM  1.9655  1.9493  1.9340  1.9328  1.9316
##
## Optimal Scores:
##
##           Score  Method  Clusters
## APN 0.0166 hierarchical 2
## AD  2.5069 pam       6
## ADM 0.3923 hierarchical 2
## FOM 1.9316 clara      6

```

5 K-Means Yöntemi

Mac Queen tarafından geliştirilmiştir. Bu yöntemde önce araştırmacının ön bilgisine ve tecrübesine dayanarak küme sayısı belirlenir. Sonra her kümenin tipik bir gözlemi seçilir, benzer gözlemler tipik gözlemin etrafında birer birer kümelendirilir. Burada bazı istatistiksel testler kullanılarak her kümeyi oluşturan gözlemlerin değişkenlere göre ortalamalarına bakılır. Güvenilir olması en belirgin üstünlüğüdür.

Amaç küme içi benzerliği yüksek kümeler arası benzerlik düşük olmalıdır.

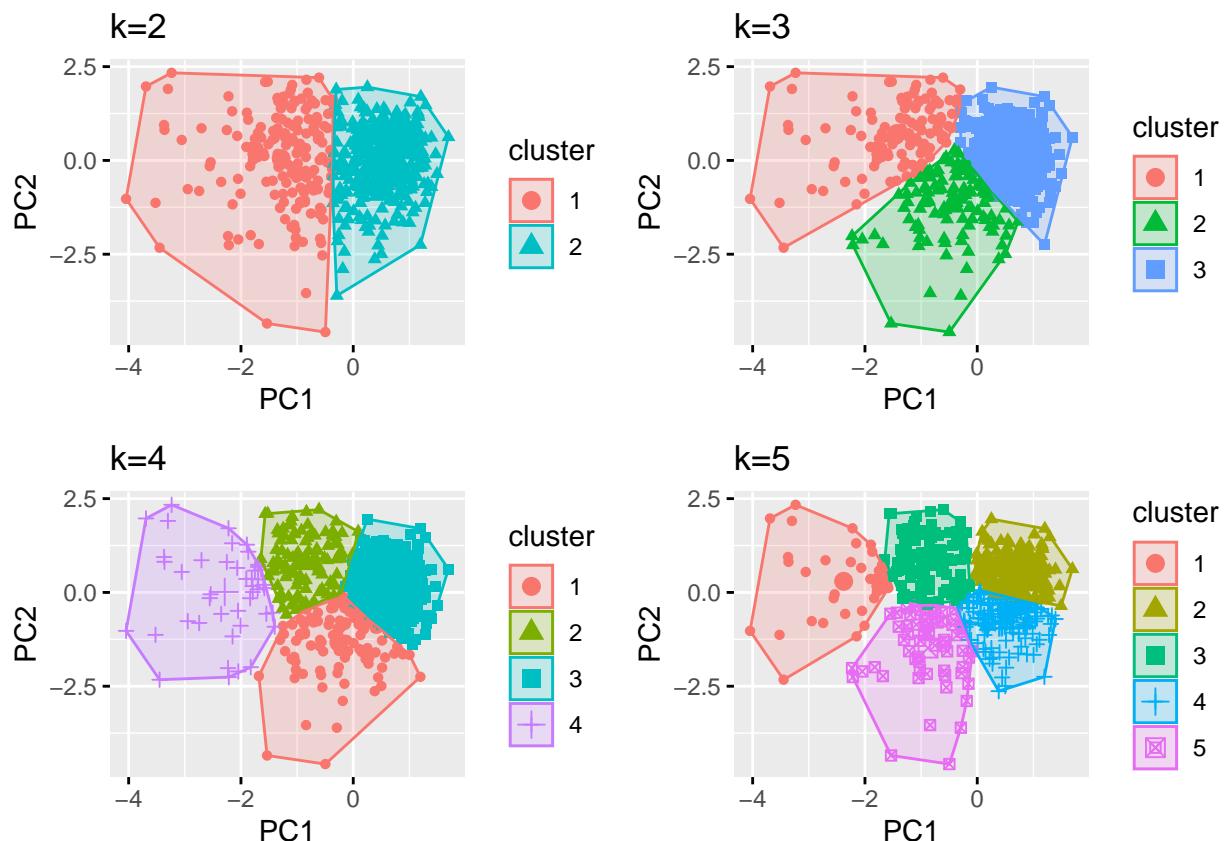
Kumelerin Görselleştirilmesi

Kümeler ayrimı en iyi 2 kümelemede gözükmektedir.

```
set.seed(123)
k2 <- kmeans(pca_data, centers = 2, nstart = 25)
k3 <- kmeans(pca_data, centers = 3, nstart = 25)
k4 <- kmeans(pca_data, centers = 4, nstart = 25)
k5 <- kmeans(pca_data, centers = 5, nstart = 25)

p1 <- fviz_cluster(k2, geom = "point", data = pca_data) + ggtitle("k=2")
p2 <- fviz_cluster(k3, geom = "point", data = pca_data) + ggtitle("k=3")
p3 <- fviz_cluster(k4, geom = "point", data = pca_data) + ggtitle("k=4")
p4 <- fviz_cluster(k5, geom = "point", data = pca_data) + ggtitle("k=5")
library(gridExtra)

grid.arrange(p1, p2, p3, p4, nrow = 2)
```



Optimum K-means Kume Sayisinin Belirlenmesi

- Küme içi hata minimum olmalı, kümeler arası hata maximum olmalı. • Küme içindeki gözlemlerin Küme merkezlerine olan uzaklıklarını üzerinden yapılan kare toplamı hesabı minimum olmalıdır. • Farklı sayıdaki k değerlerine göre oluşturulan kümeleme çalışmalarının herbirisinin hesaplanan toplam hata kareleri toplam değerleri karşılaştırılarak optimum k belirlenir.

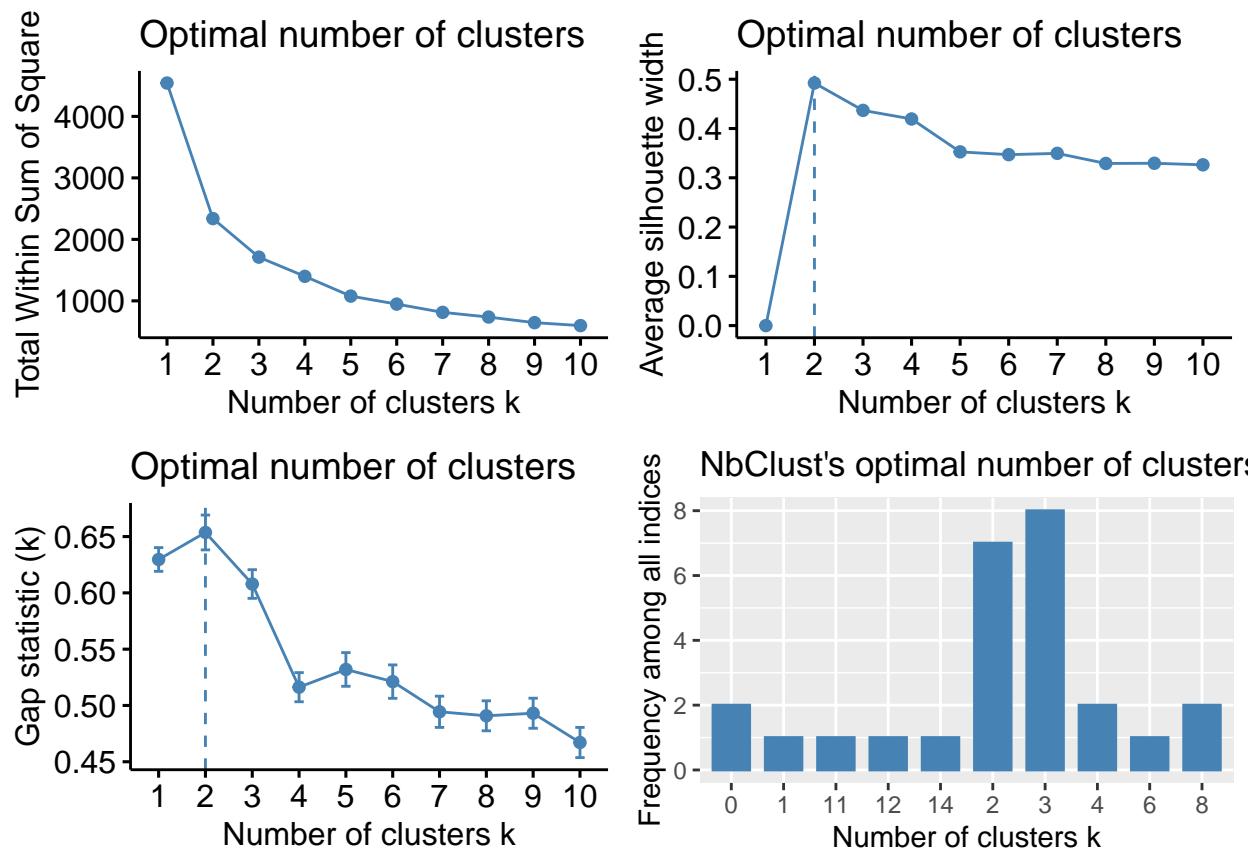
wss, Silhouette, gap istatistigi, nbclust fonksiyonunda yer alan istatistiklerden elde ettigimiz sonuçlara göre wss ye göre dirsek k = 2 den itibaren kırılmıştır.

Silhouette width en optimal küme sayısını 2 olarak göstermiştir.

Gap istatistigi en optimal küme sayısını 2 olarak belirlemiştir.

nbclust fonksiyonunda yer alan istatistikler çoğunlukla 3 küme nin en optimal küme sayısı olarak belirlemiştir.

```
grid.arrange(elbow, silhouette,gap,nbclust, nrow = 2)
```



```

set.seed(123)
final <- kmeans(pca_data, 2, nstart = 25)
str(final)

## List of 9
## $ cluster      : Named int [1:569] 1 1 1 1 1 1 1 1 1 1 ...
## ..- attr(*, "names")= chr [1:569] "842302" "842517" "84300903" "84348301" ...
## $ centers      : num [1:2, 1:2] -3.0017 1.2897 0.0748 -0.0321
## ..- attr(*, "dimnames")=List of 2
## ...$ : chr [1:2] "1" "2"
## ...$ : chr [1:2] "PC1" "PC2"
## $ totss        : num 4542
## $ withinss     : num [1:2] 1217 1122
## $ tot.withinss: num 2338
## $ betweenss    : num 2204
## $ size         : int [1:2] 171 398
## $ iter         : int 1
## $ ifault       : int 0
## - attr(*, "class")= chr "kmeans"

```

-cluster: 1 den k ya kadar oluşan kümeleri ifade eden vektördür. Veri setimizde hangi ID'lerin hangi kümede olduğunu göstermek istersek cluster birlesine ulaşırıp küme değerini çekmiş oluruz.

-centers: Kümelerin merkezlerini ifade eden matriztir.

- totss : Kare toplamlarının toplamıdır 4542 dir.

-withinss :Küme içi kareler toplamıdır. 1217 birinci kümenin kareler toplamıdır. ikinci kümenin kareler toplamı 1217 dir.

-tot.withinss: Tüm küme içi kareler toplamının toplamıdır. Çıktı sonucuna göre bu 2838 tir.

-betweenss:Kümeler arası kareler toplamıdır ve bu sayı 2204 dir.

-size :Her kümede bulunan gözlem sayısı 1. kümede 171 2. Kümede 116 ve 3. kümede 398 dir.

İlk 20 gözlemin hangi kümeye ait olduğu bilgisi aşağıda verilmiştir.

```
head(final$cluster,20)
```

```

##  842302  842517 84300903 84348301 84358402  843786  844359 84458202
##      1      1      1      1      1      1      1      1      1
##  844981 84501001 845636 84610002  846226  846381 84667401 84799002
##      1      1      2      1      1      2      1      1
##  848406 84862001 849014 8510426
##      2      1      1      2

```

ID'leri 2 kümeye ayırdık 1.cluster daki gözlemlerin tümörlerin ortalama_yarıçap(radius_mean) ortalaması 18.04555 ,texture_mean ortalaması 21.43696 ,tümörlerin alan ortalaması (area_mean) 1042.5199 ,1. kümeye düşen gözlemlerin tümörlerin çevresel ortalaması 119.75468 dir.

2.kümeye düşen gözlemlerin tümörlerin yarıçap ortalaması (radius_mean) 12.44382 ,texture_mean ortalaması 18.36706 ,2. kümeye düşen gözlemlerin çevresel ortalaması(perimeter_mean) 80.03098 ve tümötlerin alan ortamasları ortalaması 488.3442 tür.

```

data %>%
  mutate(Cluster = final$cluster) %>%
  group_by(Cluster) %>%
  summarise_all("mean")

```

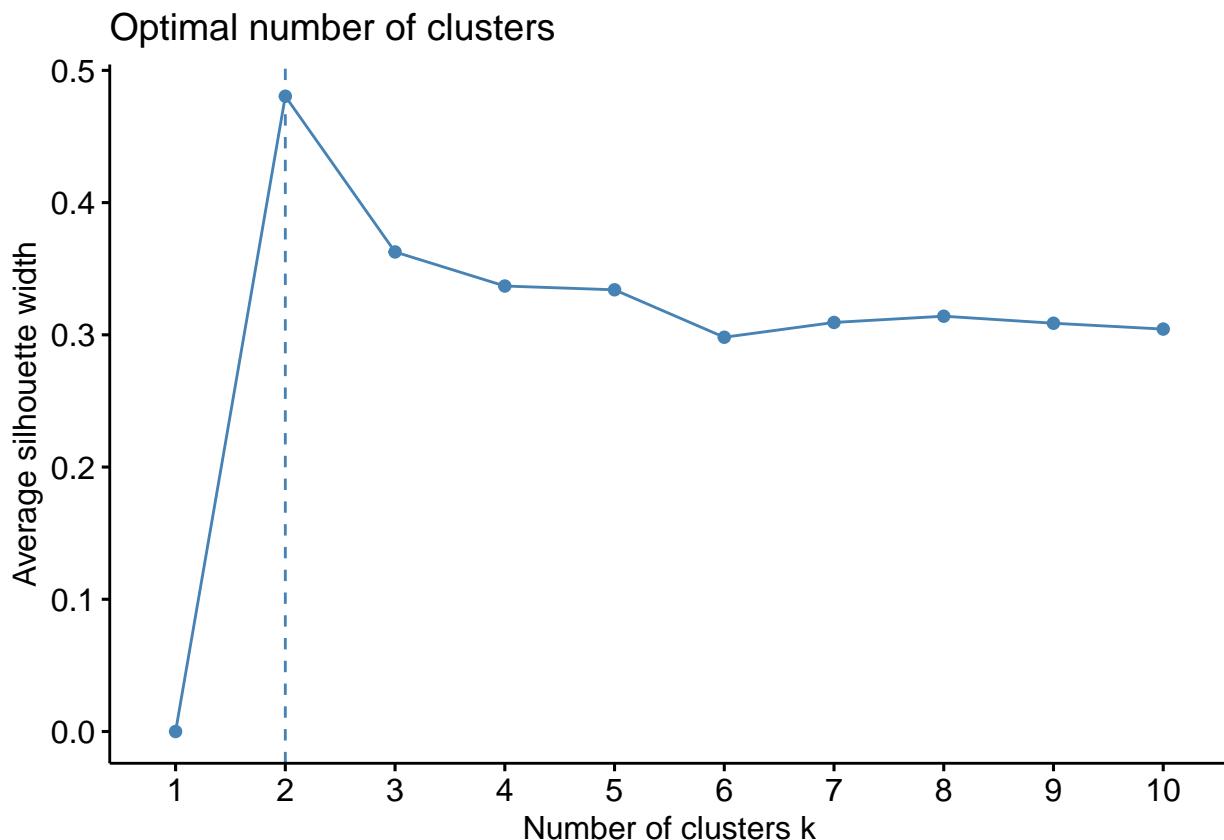
```
## # A tibble: 2 x 11
##   Cluster radius_mean texture_mean perimeter_mean area_mean smoothness_mean
##   <int>      <dbl>      <dbl>      <dbl>      <dbl>      <dbl>
## 1       1      18.0      21.4     120.     1043.      0.106
## 2       2      12.4      18.4      80.0      488.      0.0924
## # ... with 5 more variables: compactness_mean <dbl>,
## #   concavity_mean <dbl>, symmetry_mean <dbl>,
## #   fractal_dimension_mean <dbl>
```

6 K-medoids Yöntemi

K-medoids algoritmasının temeli, verinin çeşitli yapısal özelliklerini temsil eden k tane temsilci nesneyi bulma esasına dayanır (Kaufman ve Rousseeuw, 1987). Temsilci nesne medoid olarak adlandırılır ve kümenin merkezine en yakın noktadır. Bir grup nesneyi k tane kümeye bölerken asıl amaç, birbirine çok benzeyen nesnelerin bir arada bulunduğu ve farklı kümelerdeki nesnelerin birbirinden benzersiz olduğu kümeleri bulmaktadır. k adet temsilci nesne tespit edildikten sonra her bir nesne en yakın olduğu temsilciye atanarak k tane küme oluşturulur. Sonraki adımlarda her bir temsilci nesne temsilci olmayan nesne ile degistirilerek kümelemenin kalitesi yükseltilinceye kadar ötelenir. k -medoids yöntemi için *optimum k sayısı silhouette yöntemi* ile belirlenebilir.

Grafikte görüldüğü gibi ortalama silhouette genişliği en yüksek olana göre seçilen k Optimum sayısı 2 dir.

```
fviz_nbclust(pca_data, pam, method= "silhouette")
```

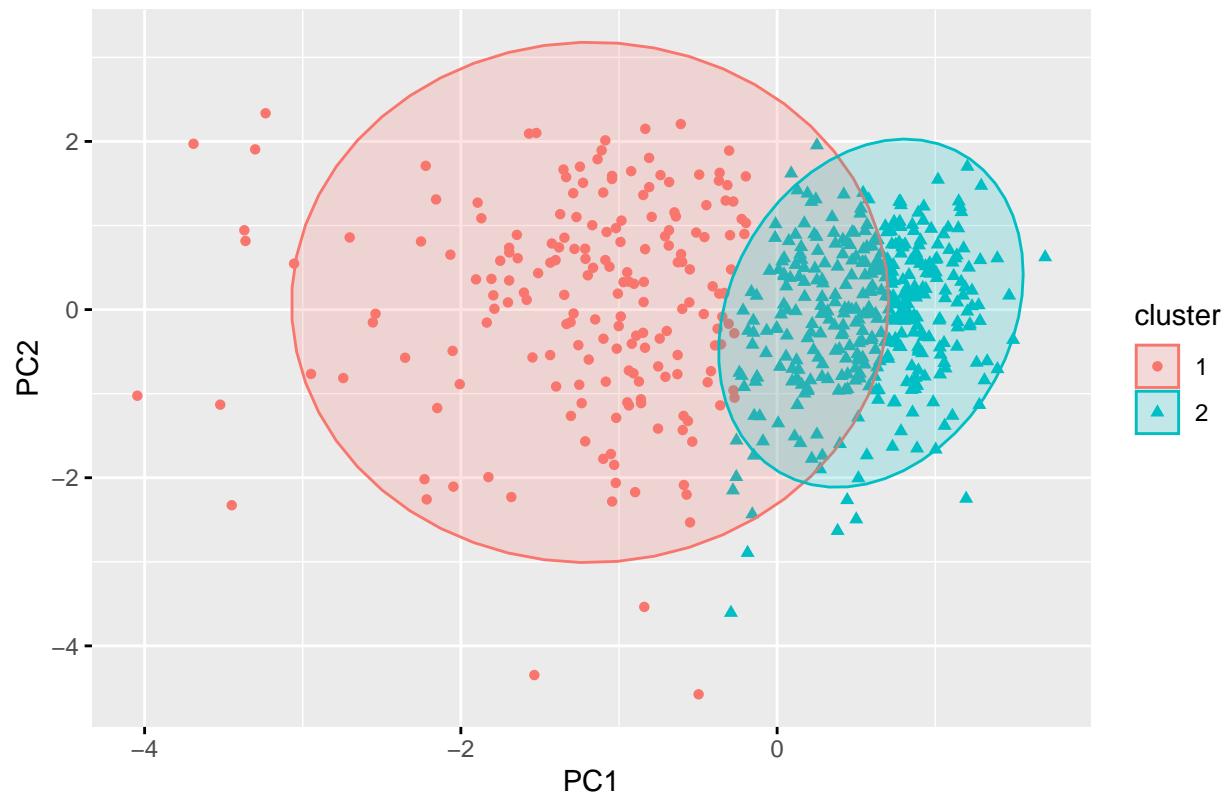


1. kümede 190 gözlem, 2. kümede 379 gözlem bulunmaktadır. Uzaklık matrisinde belirlendiği gibi, aykırı gözlemler net kümelemememiştir. Uç değerlerden varlığı ve verilerin çoğunun merkeze yakın olmasından dolayı görsel bu şekildedir. 3 veya daha fazla küme seçildiği durumda overfitting durumu ile karşılaşılabilir.

```
set.seed(123)
data_pam=pam(pca_data,2)
table(data_pam$clustering)
```

```
## 
##   1   2
## 190 379
fviz_cluster(data_pam,
  ellipse.type = "norm",geom = 'point' ,data=pca_data)
```

Cluster plot



7 Hiyerarsik Kumeleme Analizi

Amaç gözlemleri birbirlerine göre kümelere ayırmaktır. Gözlemler daha fazla sayıda alt kümeye ayrılmak istendiğinde kullanılır.

Hiyerarsik kumeleme iki grupta incelenebilir, bunlar yiğilmalı hiyerarsik kumeleme ve bölünmeli hiyerarsik kumelemedir. Yiğilmalı hiyerarsik kumeleme verideki her bir gözlemi bir küme olarak düşünür. Birleştirme işlemleri uygulanarak kümeler tek bir küme elde edilinceye kadar devam ettirilir. Bölünmeli hiyerarsik kumelemede, başlangıçta tüm birimlerin bir küme oluşturduğu kabul edilerek, birimleri aşamalı olarak kümelere ayırrı.

7.1 Birlestirici Hiyerarsik Kumeleme

Manhattan Uzaklığın ayarlanması

```
data_manh=dist(pca_data, method="manhattan")
round(as.matrix(data_manh) [1:2, 1:10], 2)

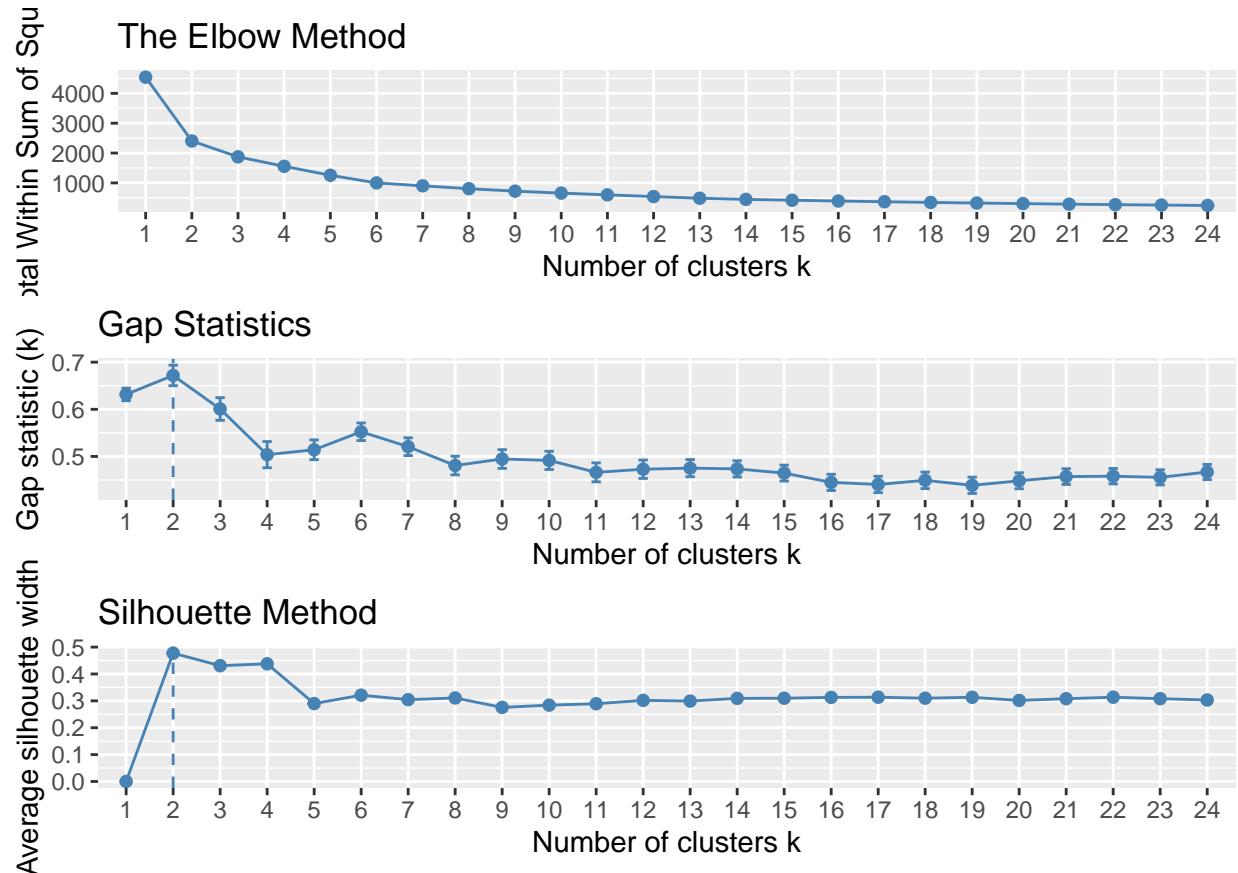
##          842302 842517 84300903 84348301 84358402 843786 844359 84458202 844981
## 842302    0.00   9.23     5.00     5.32     6.63     3.95     8.32     4.67     2.90
## 842517    9.23   0.00     4.23    11.30     2.60     6.20     1.17     5.50     6.47
##          84501001
## 842302     3.20
## 842517     6.88
```

Tüm gözlemlerin ayrı bir küme olarak kabul edilerek daha büyük bir kümede birleştirerek ilerleyen yöntem AGNES yöntemidir.

Optimal Hiyerarsik methodun Uygulanması:

Optimal Hiyerarsik kumeleme yöntemlerine göre en optimal kumeleme sayısı 2 dir.

```
set.seed(123)
# function to compute total within-cluster sum of squares
elbow <- fviz_nbclust(pca_data, FUN = hcut, method = "wss", k.max = 24) + ggtitle("The Elbow Method") +
# Gap Statistics
gap <- fviz_nbclust(pca_data, FUN = hcut, method = "gap_stat", k.max = 24) + ggtitle("Gap Statistics") +
# The Silhouette Method
silhouette1<- fviz_nbclust(pca_data, FUN = hcut, method = "silhouette", k.max = 24) + ggtitle("Silhouette") +
grid.arrange(elbow,gap,silhouette1)
```



7.2 Birleştirici Methodların Karşılaştırılması

Agnes fonksiyonu bize birleştiricilik katsayısını oluşturur. Bu oluşturulan katsayı ile oluşturulacak küme yapısının gücünü ölçebiliriz.

Aşağıdaki çıktıya göre ac istatistiği katsayı 0.9850311 dir.

```
hc2 <- agnes(data_manh, method = "complete")
hc2$ac
```

```
## [1] 0.9850311
```

Aşağıdaki fonksiyon bize en iyi birleştirici yöntemi seçmemizi sağlar. Elde edilen ac istatistiklerini karşılaştırarak buluruz.

Her bir method için agnes fonksiyonu çalıştırılır ve her bir method için ac istatistiği elde edilip kümeler arası karşılaştırma yapabiliriz. Görüldüğü gibi en iyi ac istatistik değeri ward methoduna ait 0.9965290, hiyearşik kümleme yönteminde ward methodu seçebiliriz.

```
m <- c("average", "single", "complete", "ward")
names(m) <- c("average", "single", "complete", "ward")

ac <- function(x) {
  agnes(data_manh, method = x)$ac
}
```

```
sapply(m, ac)

##   average   single  complete    ward
## 0.9717276 0.9193416 0.9850311 0.9965290
```

Kojenetik değeri ise 0.75'den büyük olması veri setini daha iyi yansittığını göstermektedir. Ne kadar yüksek bir değer alırsa veri seti o kadar iyi yansitılmış demektir.

Burada ise 0.60 değerini almış yani veri setini en iyi method kullanışmasına rağmen veri setini iyi yansitamadığı görülmektedir. 1.gruba 231 gözlem, 2.gruba 338 gözlem düşmektedir.

```
hc2 <- agnes(data_manh, method = "complete")
hc3 <- agnes(data_manh, method = "ward")
grup_veri=cutree(hc3, k=2)
table(grup_veri)
```

```
## grup_veri
##   1   2
## 231 338
```

Kojenetik degeri ise 0.75'den büyük olması veri setini daha iyi yansittığını göstermektedir. 0.60 veri setini iyi yansitmaz.

```
coph_veri=cophenetic(hc3)
cor(data_manh,coph_veri)
```

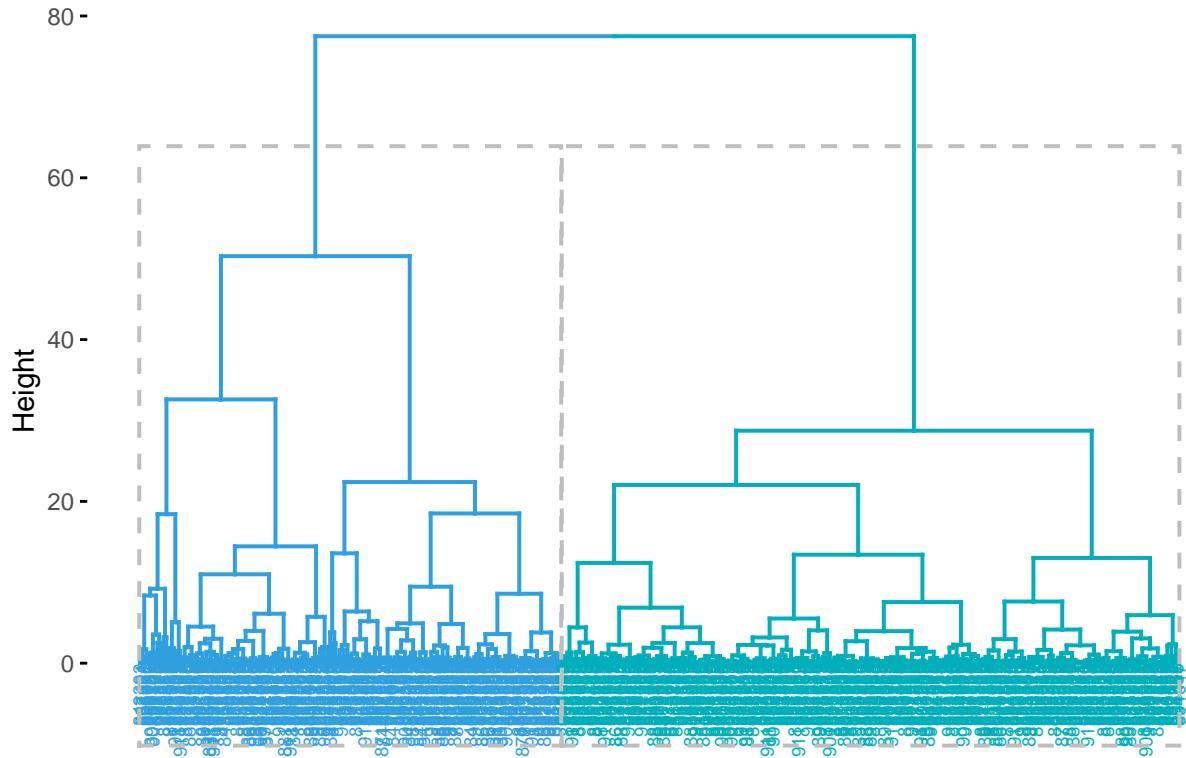
```
## [1] 0.6018289
```

Veri setinde gözlemler hangi grupta görselleştirmesi

Aşağıda gözlemlerin hangi kümede olduğunu gösteren grafik bulunmaktadır.

```
fviz_dend(hc3, k = 2,
           cex = 0.5,
           k_colors = c("#2E9FDF", "#00AFBB", "#FC4E07"),
           color_labels_by_k = TRUE,
           rect = TRUE
)
```

Cluster Dendrogram



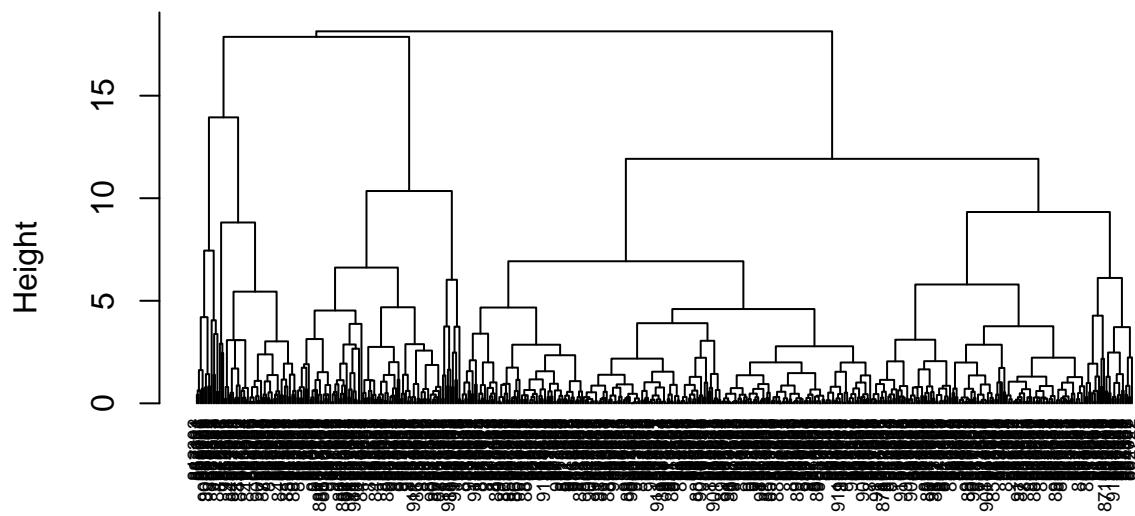
7.3 Bolumleyici Hiyerarsik Kumeleme

Bolumleyici yöntemin bölümleyicilik katsayısı 0.982939 dur.

```
hc4 <- diana(data_manh)
hc4$dc

## [1] 0.982939
pltree(hc4, cex = 0.6, hang = -1)
```

Dendrogram of diana(x = data_manh)



```
data_manh  
diana (*, "NA")
```

8 Model Tabanlı Kümeleme

Model-tabanlı kümeleme metotları, verilen veri ve bazı matematiksel modeller arasında uygunluğu optimize etmeye çalışır.

Model temelli kümeleme, verilerin bir model tarafından oluşturulduğunu varsayar ve veriden orijinal modele erişmeye çalışır. Erişilen model ile kümeler tanımlanır. Verilerin iki veya daha fazla kümenin karışımı olan bir dağılımdan geldiğini düşünen model tabanlı kümeleme bir alternatif kümelemedir (Chris Fraley ve Adrian E. Raftery, 2002 ve 2012).

k-ortalamadan farklı olarak, model tabanlı kümeleme, her veri noktasının her bir kümeye ait olma olasılığına sahip olduğu bir atama kullanır.

Bu analizde mclust paketi kullanılarak analiz yapılarak ve kaç küme olması gerekiğine karar verilmiştir.

Mclust yöntemine göre iki küme oluşturmuştur, birinci kümeye 253 ve ikinci kümeye 316 gözlem olarak kümelemiştir.

Bayesian Information Criteria (BIC)'nın en küçük olduğu modeli seçer. 569 gözlem için BIC değerinin küçük model seçenekler -4510.449 olarak bulunmuştur.

```
set.seed(123)
model_base = mclust:: Mclust(pca_data)
summary(model_base)

##
## Gaussian finite mixture model fitted by EM algorithm
##
## Mclust VVI (diagonal, varying volume and shape) model with 2 components:
##
##  log-likelihood    n  df      BIC      ICL
##            -2226.677 569  9 -4510.449 -4654.213
##
## Clustering table:
##   1   2
## 253 316
```

Veri setinin ;

-duyarlılığı($59/(59+298) = 0.16526$) %16.5 -doğruluk değeri($(59+18)/(58+194+298+18) = 0.135$) %13.5'dir.
-Seçiciliği ($18/(194+18) = 0.085$) %8.5'dir.

Bu değerlere göre kümeleme için model base yeterli bir yöntem değildir.

```
table(model_base$classification, data1$diagnosis)

##
##      B      M
## 1 59 194
## 2 298 18
```

Veri setindeki her gözlemin bütün kümeye ait olma olasılıklarının toplamı 1'dir. İlk 3 gözlemin hangi kümeye ait olduğu bilgisi aşağıda gösterilmiştir.

- 842302 ID'li gözlemin 1. kümede olma olasılığı %100'dür.
- 842517 ID'li gözlemin 1. kümede olma olasılığı %99'dur ve sıfıra oldukça yakın olasılıklarla ikinci kümeye aittir.
- 84300903 ID'li gözlemin 1. kümede olma olasılığı %100'dür.

```
head(model_base$z, 3)

##           [,1]           [,2]
## 842302    1.000000 3.332766e-18
## 842517    0.999902 9.797021e-05
## 84300903  1.000000 3.526042e-11
```

ilk 3 gözlemin satır toplamı verilmistir.her gözlemin kümelere atanma degerlerinin toplamının 1 oldugunu buradan görebiliriz.

```
rowSums(head(model_base$z, 3))
```

```
## 842302 842517 84300903
## 1       1       1
```

Üstteki kodda bulduğumuz değerlerin yüksek olasılıkla hangi kümedeyse o kümeye atanmış en son hali aşağıda bulunmaktadır.

```
head(model_base$classification, 3)
```

```
## 842302 842517 84300903
## 1       1       1
```

Model Temelli Kümeleme Parametrelerinin kestirimi:

Kümeler için İlk tanımlayıcı hacim,ikincisi şekil, üçüncüsü yönelim anlamına gelir.

Hacim-Şekil-Yönelim E(equal): Eşit / V(vary): (değişik) / I(identity): benzer

İlk grafik en optimal küme sayısını BIC değerleri hesaplanmış şekilde gösterir.Büyük bir BIC puanı, karşılık gelen model için güclü kanıtlar olduğunu gösterir.

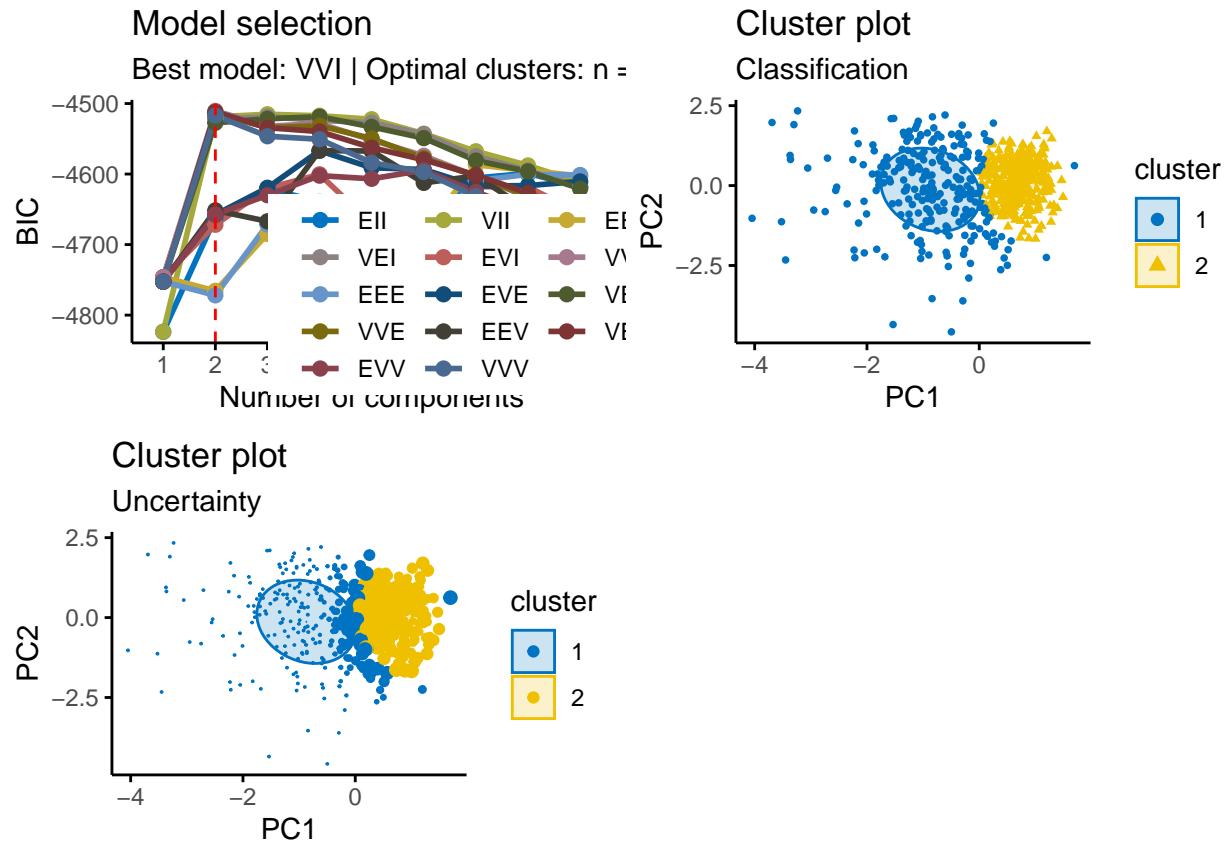
En optimal küme sayısı 2 dir en iyi model VVI yani tüm kümelerin hacimlerinin değişken olduğu (V), kümelerin şekillerinin değiştirebileceği (V) ve yönelimin (koordinat eksenleri) benzer olduğunu (I) belirtir.

İkinci grafik kümelenmeyi gösterir.

3. grafik kümelenme belirsizliğini göstermektedir.Görseldeki mavi ve yuvarlak büyük noktalar potansiyel gürültü gözlemleridir,kümeye çok küçük olasılıklarla dahil edildigini gösterir.

```
a<-fviz_mclust(model_base, "BIC", palette = "jco", size=1)

b<-fviz_mclust(model_base, "classification", geom = "point",
                 pointsize = 1, palette = "jco")
# Classification uncertainty
c<-fviz_mclust(model_base, "uncertainty", palette = "jco")
grid.arrange(a,b,c,nrow=2)
```



9 Yoğunluk Temelli Kümeleme

Ester ve ark. (1996) gürültü ve aykırı değerler içeren bir veri setinin herhangi bir şekildeki kümelerini tanımlamak için geliştirmiştir.

Eğer gözlemler birbirine yoğunca konumlanmış alanda yakın ise bunları bir kümeye alma mantığına dayanır. *eps* erişilebilirlik uzaklığı *çevre* ,*minpoints* o belirlenen alandaki gözlem birimi sayısını ifade eder, belirlenmiş alanda ki minimum gözlem sayısını ifade eder.

Avantajları

- k-means'den farklı olarak, küme sayısının belirtilmesine gerek yoktur.

-Düzensiz şekilli verilerde k-ortalama kümeleme yöntemi güçlük çekmektedir. Herhangi bir küme şeklini bulabilir. Kümenin dairesel olması gerekmektedir.

-Aykırı değerleri belirleyebilir.

DBSCAN algoritması “kümeler” ve “gürültü” kavramını temel alır. Ana fikir, bir kümenin her noktası için, belirli bir yarıçapın komşusunun en az minimum sayıda nokta içermesi gereklidir.

DBSCAN için iki önemli parametre gereklidir: *epsilon* (“*eps*”) ve minimum noktalar (“*MinPts*”). *eps* parametresi, x noktasının çevresindeki komşuların yarıçapını tanımlar. Buna x ’in *eps* komşuluğu denir. Grafiklerde Güçlü bir büükümme olan yer *epsilon* için uygun değerlerdir. Çok küçük seçildiği durumda herhangi bir kümeye atanacak veri gürültü olarak tanımlanabilir.

MinPts parametresi, “*eps*” yarıçapı içindeki minimum komşu sayısıdır. “Kaç komsuya sahip olursa bir kümeler olusturur?” sorusuna cevap veren parametredir. Sonucunda çekirdek nokta saptanmış olacak. Komsu sayısı minimum nokta az ise sınır nokta olarak tanımlanır, en az 3 seçilir. Veri setinin büyülüğine göre değişir.

Eğer bir nokta ne çekirdek ne de sınır nokta olarak tanımlanırsa, *gürültü* yada *aykırı* değer olarak tanımlanır.

Kısaca *eps* parametresi == erişilebilirlik uzaklığı ; *MinPts* == belirlenen alandaki gözlem birimi sayısı Birinci kümeye 107 gözlem düşmüş, ikinci kümede 366 tane gözlem vardır. 1.kümenin 58 tanesi sınır gözlemdir. İkinci kümenin 27 tanesi sınır gözlemdir.

```
set.seed(123)
density_data <- fpc:: dbscan(pca_data, eps = 0.6, MinPts = 10)
density_data
```

```
## dbscan Pts=569 MinPts=10 eps=0.6
##      0   1   2
## border 96  58  27
## seed    0   49 339
## total   96 107 366
```

Solda 1 den 2 ye kadar olan değerler kümeleri ifade eder. 0 değeri bir kümeye dahil olamayan gözlemlerin değeridir, gürültü değerleridir. 1.kümeye düşen İyi Huylu tümörlerin sayısı 2 iken, 2.kümeye düşen iyi huylu tümörlerin sayısı 322 tir.

1.kümeye düşen kötü huylu tümörlerin sayısı 105, 2.kümeye düşen tümörlerin sayısı 44 tür.

```
table(density_data$cluster, data1$diagnosis)
```

```
##
##      B     M
## 0   33   63
## 1   2   105
## 2 322   44
```

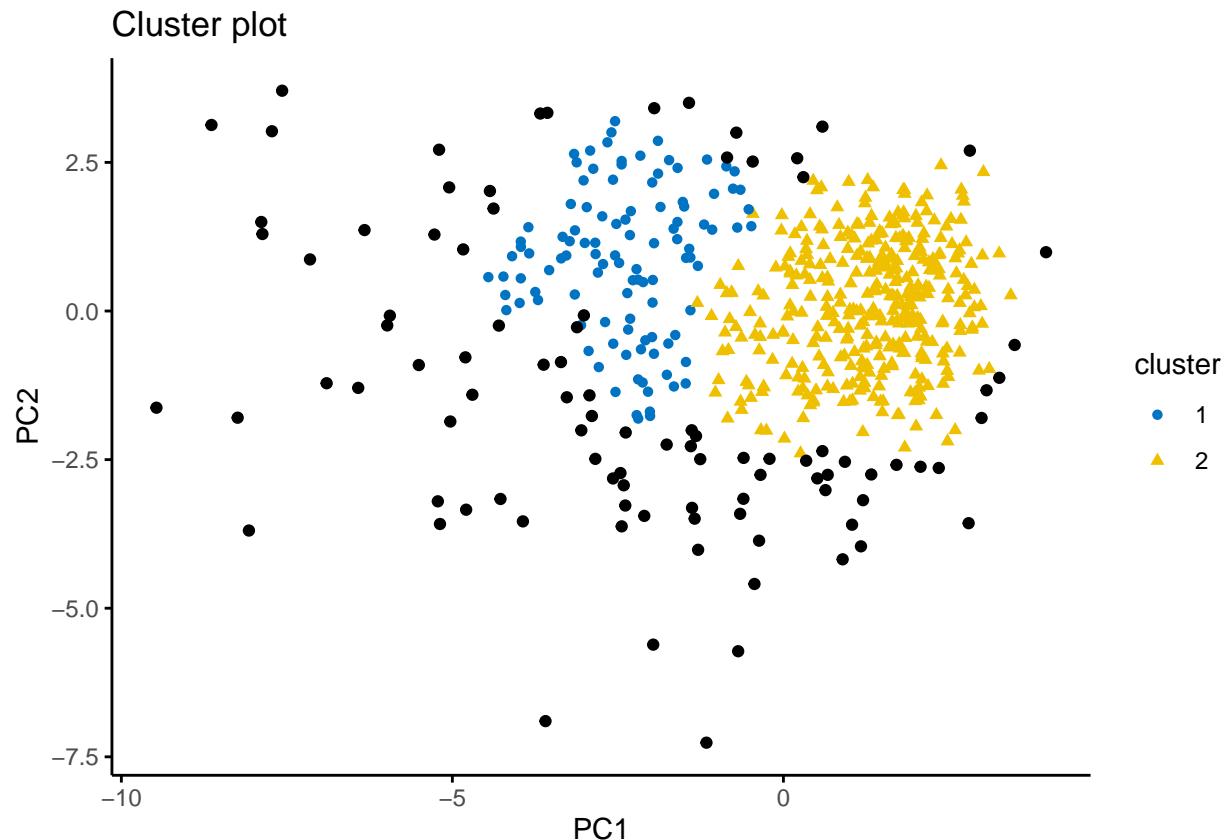
Veri setinde bütün gözlemler birbirlerine çok yakın olduğu için epsilon ve minimum nokta değeri uyumlu olması gereklidir.

Birinci kümende her iki temel bileşen tarafından açıklanabilen gözlemler olduğu görülmektedir. Her iki temel bileşen için de yüksek değerler almıştır.

İkinci küme birinci kümeye göre birinci temel bileşenle daha az ikinci temel bileşenle daha çok açıklanmıştır.

Aykırı gözlemlerde net bir şekilde görülmektedir.

```
factoextra:: fviz_cluster(density_data, data = pca_data, stand = FALSE,
ellipse = FALSE, show.clust.cent = FALSE,
geom = "point", palette = "jco", ggtheme = theme_classic())
```



Veri setinin ;

-duyarlılığı($59/(59+298) = 0.16526$) %16.5 -doğruluk değeri($51+82)/(51+82+306+130) = 0.2337$ %23.37 dir.
-Seçiciliği $82/(130+82) = 0.38679$ %38.67 dir.

Bu değerlere göre kümelenme için yoğunluk temelli kümelenme yeterli bir yöntem degildir.

```
table(density_data$isseed,data1$diagnosis)
```

```
##
```

	B	M
## FALSE	51	130
## TRUE	306	82

9.1 Küme Geçerliliği

Veri setinde üç değer fazla olduğu için manhattan uzaklık ölçütü kullanıldı.

Hopkins istatistiği ile veri setinin tekdüze dağılımdan üretilme olasılığı ölçüldü ve veri kümesinin kümelenme eğilimi değerlendirildi, veri seti kümülenebilir sonucuna varıldı.

En İyi Küme Algoritması Seçiminde Hiyerarsik olmayan ve hiyerarsik kümeleme yöntemlerinde hangi kümeleme yönteminin daha iyi sonuç verdigini ölçmek için clValid komutu kullanıldı.

Internal sonucunda 2 küme seçilerek hiyerarsik yöntemler uygulanmalı sonucu çıkmıştır. Duraganch ölçüm-lerinin sonuçlarına bakarak elde edilen AD/APN/ADM/FOM değerlerinin 0'a yakın olması optimum sonuca yönlendirir. APN değeri için 2 küme, AD değeri için pam yöntemi 6 küme, FOM değeri clara yöntemi 6 küme optimum görülebilir sonucu vermiştir. Bu durumda hiyerarsik yöntemler seçilmeli 2 küme kullanılabilir sonucuna varıldı.

En Optimal Küme Sayısını Belirleme optimum küme sayısının belirlenmesi için iki fonksiyon kullanılabilir. Analizde fviz_nbclust() fonksiyonu kullanılmıştır.

-Elbow Yöntemi:

k-ortalamaları kümeleme gibi bölümleme yöntemlerinin ardındaki temel düşünce, toplam küme içi değişim en aza indirgenmesidir.

-Ortalama silhouette Yöntemi:

Kümelemenin kalitesini ölçer. Yani, her nesnenin kendi kümesinde ne kadar iyi olduğunu belirler. Yüksek ortalama silhouettedeğişliği iyi bir kümelenmeyi gösterir.

-GAP İstatistiği:

Gapistatistiği, farklı k değerleri için küme içi varyasyon içindeki toplamı, verilerin sıfır referans dağılımı altında beklenen değerleriyle karşılaştırır. Optimal kümelerin tahmini, boşluk istatistiği maksimize eden değer olur. Böylece kümeleme yapısının noktaların rasgele düzgün dağılımdan çok uzakta olduğu anlamına gelir.

Hiyearşik yöntemde ac istatistiğine bakılarak en uygun birleştirici yöntemi seçmemizi sağlamış, en uygun yöntem ward yöntemi ve 2 kümeleme çıkmıştır.

Kmeans hiyearşik olmayan kümelemede en optimal kümeleme sayısı 2 olarak belirlenmiştir.

Model tabanlı ve yoğunluk tabanlı kümelemelerde en optimal küme sayısı sonucu 2 çıkmıştır. Pam algoritması ve 2 küme ile analiz yapmıştır.

10 Finalde Elde Edilen Kümelerin Tanımlayıcı İstatistikleri Ve Yorumlanması

finalde kümeleme hiyearşik yöntem de en iyi wards methodu ve 2 kümeleme en iyi kümelemenin yapılacağı sonucuna varılmıştır.

```
res.hc <- data %>%
  scale() %>%
  eclust("agnes", k = 2, graph = FALSE)
```

ID'leri 2 kümeye ayırdık 1.cluster daki gözlemlerin tümörlerin ortalama_yarıçap(radius_mean) ortalaması 16.65071 ,texture_mean ortalaması 20.53825 ,tümörlerin alan ortalaması (area_mean) 908.8392 ,1. kümeye düşen gözlemlerin tümörlerin çevresel ortalaması 110.15410 dir.

2.kümeye düşen gözlemlerin tümörlerin yarıçap ortalaması (radius_mean) 12.57166 ,texture_mean ortalaması 18.51991 ,2. kümeye düşen gözlemlerin çevresel ortalaması(perimeter_mean) 80.75835 ve tümörlerin alan ortamasların ortalaması 498.3347 tür.

```
data %>%
  mutate(Cluster = res.hc$cluster) %>%
  group_by(Cluster) %>%
  summarise_all("mean")
```

```
## # A tibble: 2 x 11
##   Cluster radius_mean texture_mean perimeter_mean area_mean smoothness_mean
##   <int>     <dbl>      <dbl>       <dbl>      <dbl>        <dbl>
## 1     1      16.7      20.5      110.      909.       0.106
## 2     2      12.6      18.5      80.8      498.       0.0904
## # ... with 5 more variables: compactness_mean <dbl>, concavity_mean <dbl>,
## #   concave.points_mean <dbl>, symmetry_mean <dbl>,
## #   fractal_dimension_mean <dbl>
```

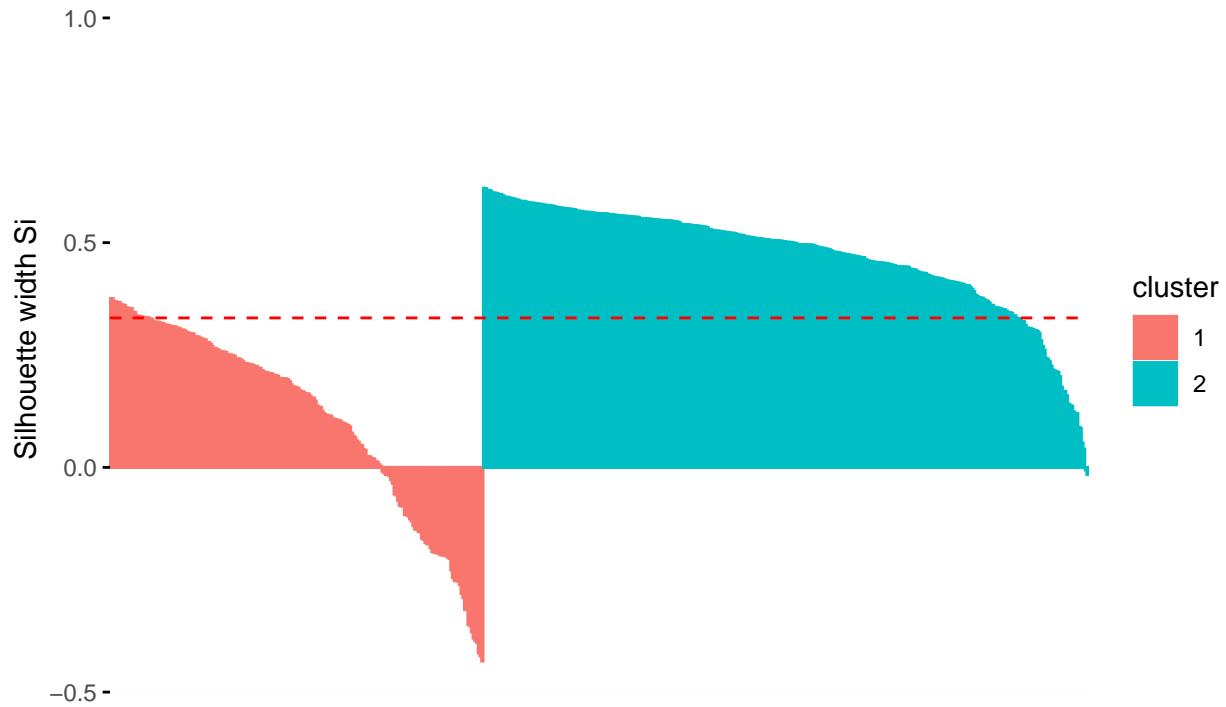
Silhouette katsayısı: bir gözlemin ne kadar iyi kümelendirildiğini ölçer ve kümeler arasındaki ortalama mesafeyi tahmin eder. Silhouette grafiği: bir kümedeki her noktanın komşu kümedeki noktalara ne kadar yakın olduğunu bir ölçüsünü görüntüler. 1 civarında bir değer alması iyi kümelendiğini 0 civarında değer alması iki küme arasında konumlandığını negatif değer alması büyük olasılıkla yanlış kümede konumlandığını gösterir.

2.kümeye olanlar en doğru kümelenenlerdir.

```
fviz_silhouette(res.hc)

##   cluster size ave.sil.width
## 1       1   217        0.11
## 2       2   352        0.47
```

Clusters silhouette plot
Average silhouette width: 0.33



10.1 Verinin Son Hali

Orjinal veriye hiyearşik yöntemle kümelenen verinin küme bilgilerini orjinal veri setine eklemiş olduk.

Id numarası 842302 olan gözlem kötü huylu olan tümörünün ortalama yarıçapı 17.990 texture mean değeri 10.38 dir ve 1.kümeye atanmıştır.

Id numarası 8210653 olan gözlemin iyi huylu olan tümörünün ortalama yarıçapı 13.080 alan ortalaması değeri 520.0 dir ve 1.kümeye atanmıştır.

```
data2 %>%
  mutate(Cluster = res.hc$cluster) %>%
  group_by(Cluster) %>%
  select(id, diagnosis, Cluster, radius_mean, texture_mean, area_mean, compactness_mean)
```

```
## # A tibble: 569 x 7
## # Groups:   Cluster [2]
##       id diagnosis Cluster radius_mean texture_mean area_mean compactness_mean
##   <int> <fct>     <int>     <dbl>      <dbl>      <dbl>          <dbl>
## 1 8.42e5 M         1      18.0      10.4      1001      0.278
## 2 8.43e5 M         1      20.6      17.8      1326      0.0786
## 3 8.43e7 M         1      19.7      21.2      1203      0.160
## 4 8.43e7 M         1      11.4      20.4      386.      0.284.
## 5 8.44e7 M         1      20.3      14.3      1297      0.133
## 6 8.44e5 M         1      12.4      15.7      477.      0.17
## 7 8.44e5 M         1      18.2      20.0      1040      0.109
## 8 8.45e7 M         1      13.7      20.8      578.      0.164
## 9 8.45e5 M         1      13.0      21.8      520.      0.193
```

```
## 10 8.45e7 M 1 12.5 24.0 476. 0.240
## # ... with 559 more rows
```