UNSUPERVİSED STATİSTİCAL LEARNİNG TAKE HOME

03/05/2020 – Caner KONUK – canerkonuk@gmail.com

1. VERİ SETİ İÇİN TANIMLAYICI İSTATİSTİKLER

Tanımlayıcı istatistikleri ve analizleri daha rahat yapabilmek için öncelikle veri setindeki Name, Team ve Position sütunlarında sayısal veri olmadığı için bu sütunları çıkardım. Öncelikle veri seti

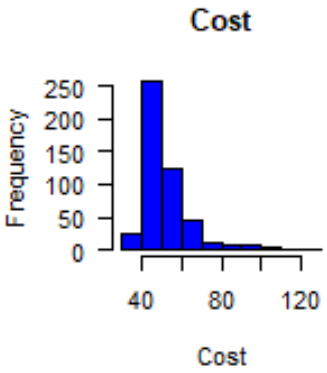
```
> dim(fpl)
[1] 480    14
```

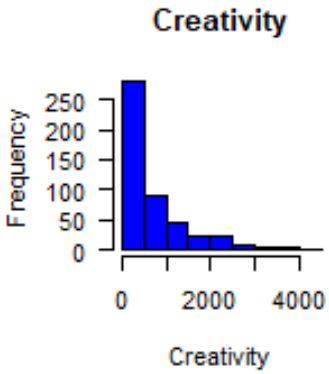
480 gözlem ve 14 değişkenden oluşuyor. (Sayısal olmayan 3 sütunu çıkardıktan sonraki hali ile bu sonucu elde ettim.) Veri setindeki verilerin aritmetik ortalaması ve min, max değerleri incelendiğinde,

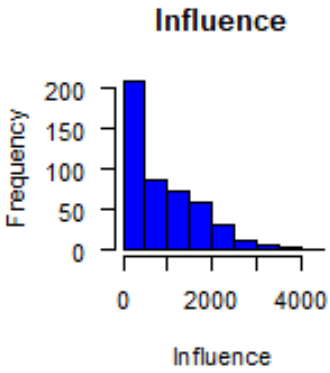
```
> summary(fpl)
      Cost         Creativity       Influence         Threat       Goals_conceded    Goals_scored       Assists       
 Min.   : 39.00   Min.   :   0.00   Min.   :   0.0   Min.   :   0.0   Min.   : 0.00   Min.   : 0.000   Min.   : 0.000  
 1st Qu.: 44.62   1st Qu.:  78.92   1st Qu.: 151.0   1st Qu.:  67.0   1st Qu.: 14.00   1st Qu.: 0.000   1st Qu.: 0.000  
 Median : 49.33   Median : 372.10   Median : 673.5   Median : 293.0   Median : 48.00   Median : 1.000   Median : 1.000  
 Mean   : 52.96   Mean   : 649.01   Mean   : 894.4   Mean   : 713.4   Mean   : 59.52   Mean   : 5.952   Mean   : 5.506  
 3rd Qu.: 56.02   3rd Qu.: 943.62   3rd Qu.:1454.7   3rd Qu.: 973.2   3rd Qu.: 96.00   3rd Qu.: 7.000   3rd Qu.: 8.000  
 Max.   :129.50   Max.   :4310.30   Max.   :4033.0   Max.   :7018.0   Max.   :215.00   Max.   :71.000   Max.   :57.000  
   Own_goals       Yellow_cards      Red_cards           TSB             Minutes          Bonus            Points      
 Min.   : 0.0000   Min.   : 0.000   Min.   : 0.0000   Min.   : 0.000   Min.   :   0.0   Min.   :  0.00   Min.   :  0.0  
 1st Qu.: 0.0000   1st Qu.: 0.000   1st Qu.: 0.0000   1st Qu.: 0.130   1st Qu.: 895.8   1st Qu.:  0.00   1st Qu.: 34.0  
 Median : 0.0000   Median : 3.000   Median : 0.0000   Median : 0.585   Median : 3505.5   Median :  3.00   Median :124.5  
 Mean   : 0.1958   Mean   : 6.725   Mean   : 0.3375   Mean   : 2.681   Mean   : 3878.0   Mean   : 13.11   Mean   :163.9  
 3rd Qu.: 0.0000   3rd Qu.:10.000   3rd Qu.: 0.0000   3rd Qu.: 2.360   3rd Qu.: 6549.2   3rd Qu.: 19.25   3rd Qu.:254.8  
 Max.   :10.0000   Max.   :50.000   Max.   :13.0000   Max.   :46.490   Max.   :10192.0   Max.   :117.00   Max.   :767.0  
```

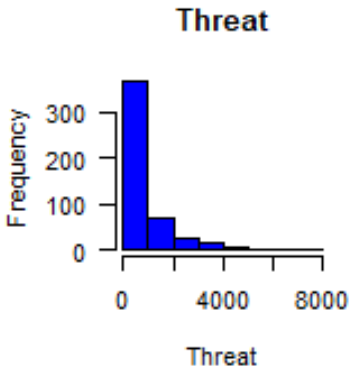
Burada min, max ve ortalama değerlerine bakıldığında Creativity, Influence, Threat gibi verilerin ortalamaları kendi içerisinde bir puanlamaya sahip ve birbirlerine yakın görünüyor. Aynı zamanda skor ile alakalı veriler yani Own_goals, Yellow_cards, Red_cards, Goals_scored, ve Goals_conceded, Assists gibi verilerde kendi içerisinde skora dayalı bir puanlamaya sahip görünüyor. Burada ayrıca Bonus ve Points verileri ayrıca temel olarak puanlamaya bağlı veriler olarak tanımlanmış. Zaten ortalamalarına ve min, max değerlerine baktığımızda bunu görebiliyoruz. Aynı zamanda burada Cost verisi parasal bir veri tipi, Minutes veri tipi zamansal bir veri tipi ve ayrıca TSB verisi veri setinin açıklamasında verildiği üzere yüzdesel bir veri tipine sahip. Yani burada asıl söylemek istediğim şey bu verilerin aslında birimlerinin farklı olması. Bazı verilerin birimi aynı olsa da hepsi birim olarak aynı değil. Bu durum ortalamalarında da kendini belli etmiş oluyor. Standart sapmalarına bakmadan önce verilerin histogramlarınıda eklemek istedim.

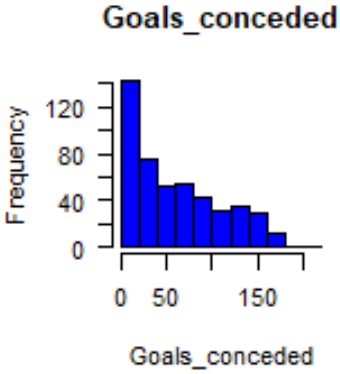
Histogramları tek tek atmak istemedim, sırası ile bu sayfaya ekledim. Verilerden elde edilen histogramlar:

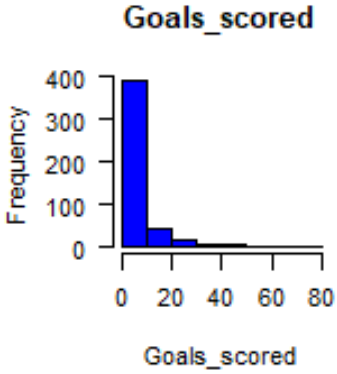


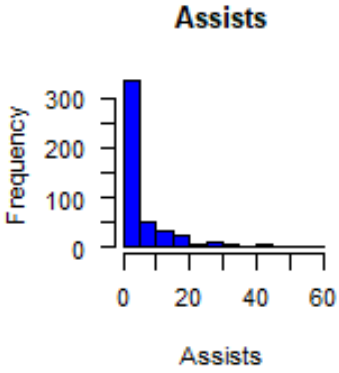


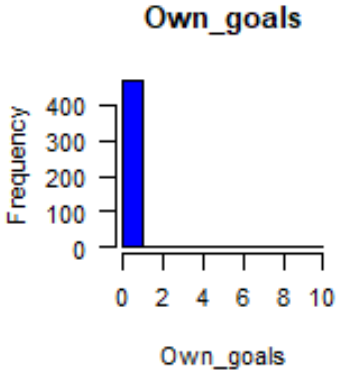


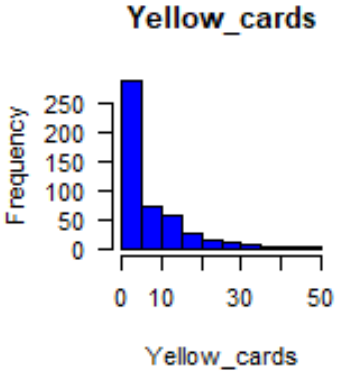


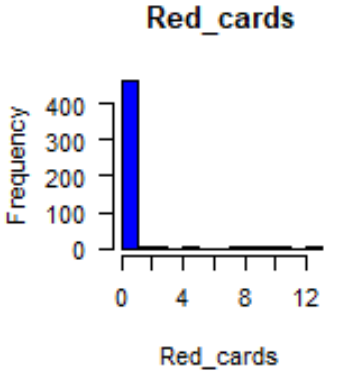


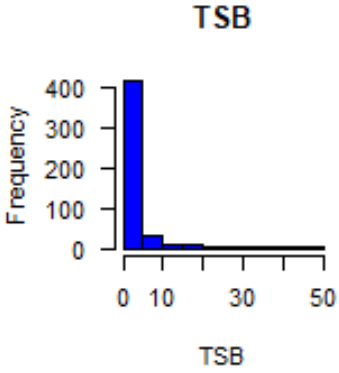


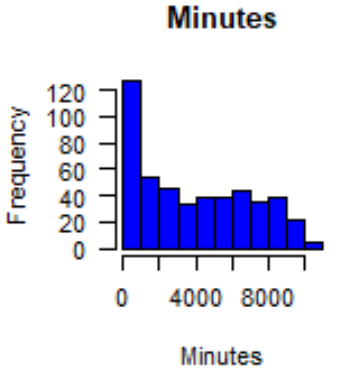


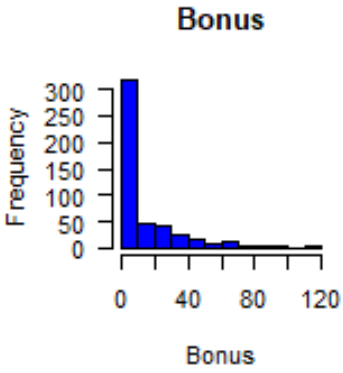


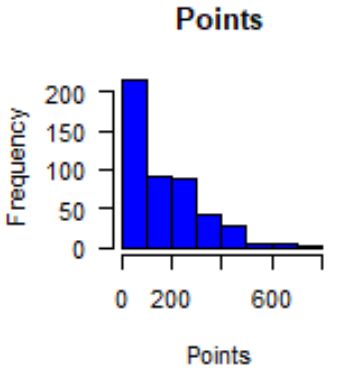








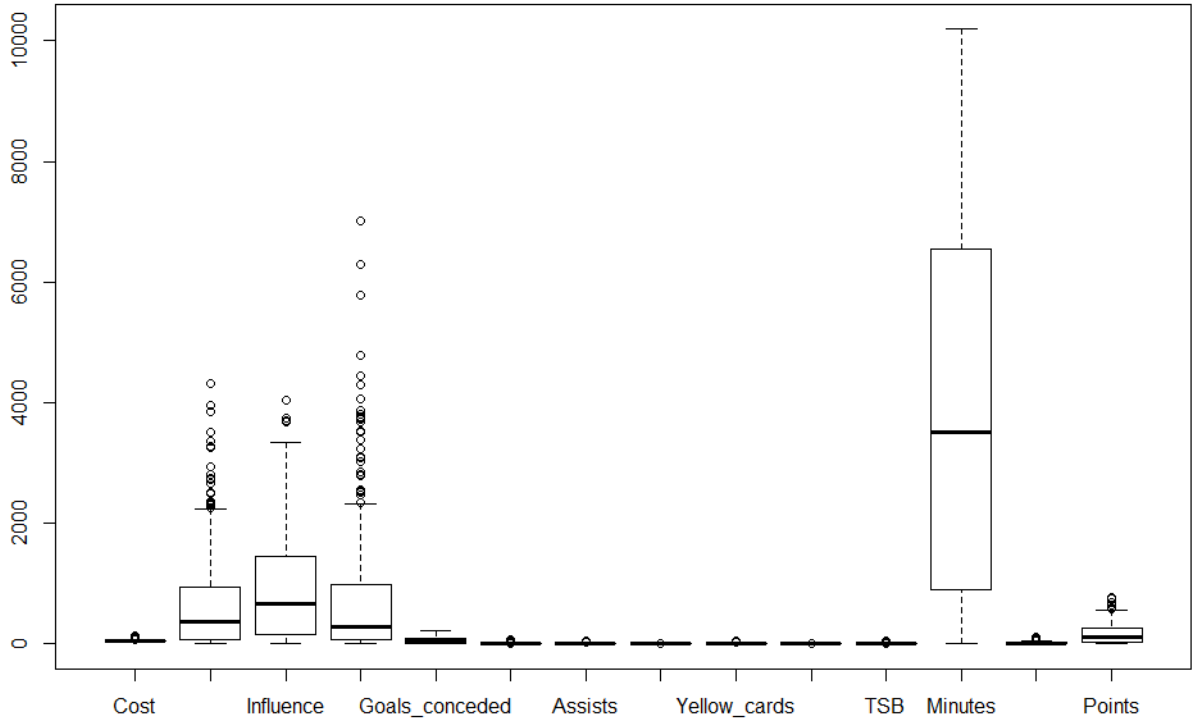




Standart sapmaları:

```
> sapply(fpl,sd)
        Cost     Creativity      Influence         Threat Goals_conceded   Goals_scored        Assists
  12.860188     758.610643     839.479626     998.804518      50.844677      11.540425       9.338450
   Own_goals  Yellow_cards      Red_cards            TSB        Minutes          Bonus         Points
   1.178789      8.600590       1.587612       5.824237    3112.595354      20.213780     154.602506
```

Genel olarak verilerin standart sapmalarına baktığımda Own_goals, Yellow_cards, Red_cards, Goals_scored, ve Goals_conceded, Assists verilerinin standart sapmalarının diğer verilerinkine göre çok düşük olduğu görülüyor. Bunun nedeninin zaten yukarıda da söylediğim gibi birimlerinin farklı olmasından kaynaklı olduğunu düşünüyorum. Ayrıca verilerin boxplot'ını incelediğimde:



Boxplot'tan görüldüğü üzere en fazla değişim Minutes verisinde meydana geliyor. Bunun dışında Creativity, Influence ve Threat verilerinde de aynı şekilde farklı bir değişim söz konusu. Bu grafik bize verilerin scale edilmesi gerektiğini söylüyor.

2. TEMEL BİLEŞEN ANALİZİ

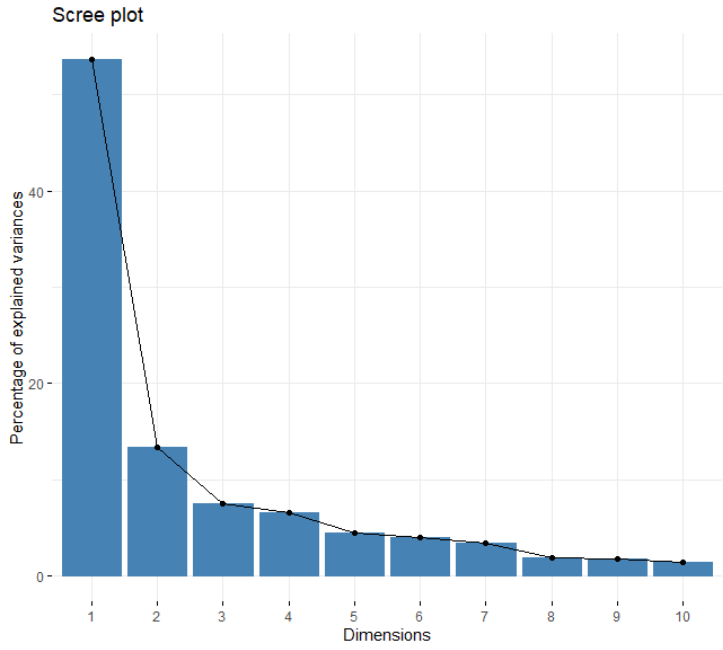
Öncelikle temel bileşen analizi yapabilmek için nümerik olmayan Name, Team ve Position verilerini veri setinden ayırdım. Temel bileşen analizine başlamadan önce ilk olarak veri setinin korelasyon matrisini inceledim. Korelasyon matrisinden gördüğüm kadarıyla en çok negatif korelasyona sahip verinin Own_goals verisi olduğunu gördüm. Aşırı derecede bir negatif korelasyona sahip olmasada Own_goals verisinin Cost, Threat, Goals_scored, Yellow_cards ve Red_cards verileri ile negatif korelasyona sahip olduğunu gördüm. Bunun dışında diğer verilerin genel olarak birbirleri ile ilişkisi pozitif yönde korelasyona sahipti. Burada en çok dikkatimi çeken şey futbolcuların fiyatını(Cost) belirleyen en önemli etkenler arasında futbolcuların kendi attıkları golün(Own_goals) fazla etkili olmaması bunun yerine futbolcu sahadayken atılan gollerin(Goals_conceded) ve futbolcunun hedefe yönelik atağının (Thread) daha etkili olduğunu gördüm. Bunun dışında korelasyon matrisinde aşırı farklı birşeye rastlamadım. Bundan dolayı korelasyon matrisini buraya eklemedim.

Öncelikle temel bileşenler analizi yaptıktan sonra temel bileşen sayısını belirlemek için sırasıyla özdeğerleri, screeplotu ve yüzde değerlerini inceleyeceğim. Temel bileşenler analizi yaptıktan sonra elde ettiğim özdeğerleri aşağıya ekledim:

```
> (fpl.pca$sdev)^2
 [1] 7.51355227 1.87078775 1.04420202 0.91806270 0.62034755 0.56318775 0.47141291
 [8] 0.26662810 0.25427460 0.19712484 0.13421007 0.08965748 0.04489954 0.01165241
```

Özdeğerlerde dikkat etmemiz gereken şey 1'den büyük olan özdeğerleri bileşen olarak seçebileceğimizdi. Burada görüldüğü üzere ilk 3 bileşenin özdeğerleri 1'den büyük. Bu yüzden bileşen sayım 3 olabilir.

Screeplot'u ve açıklayıcılığı incelediğimde ise:

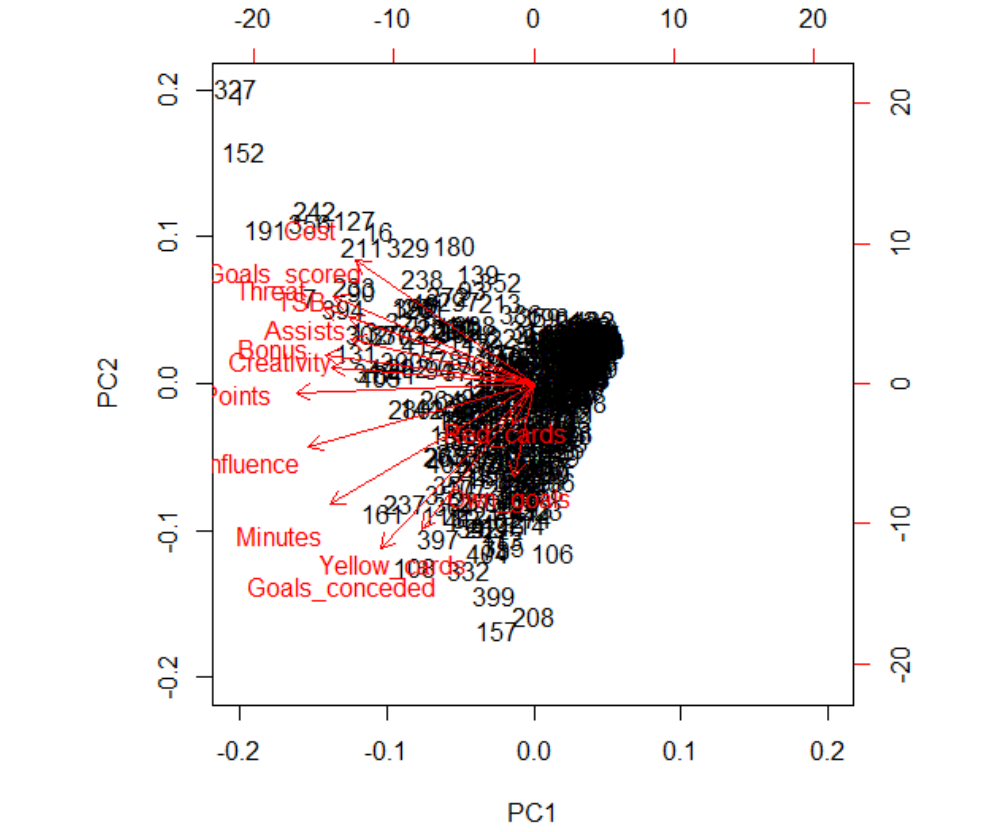


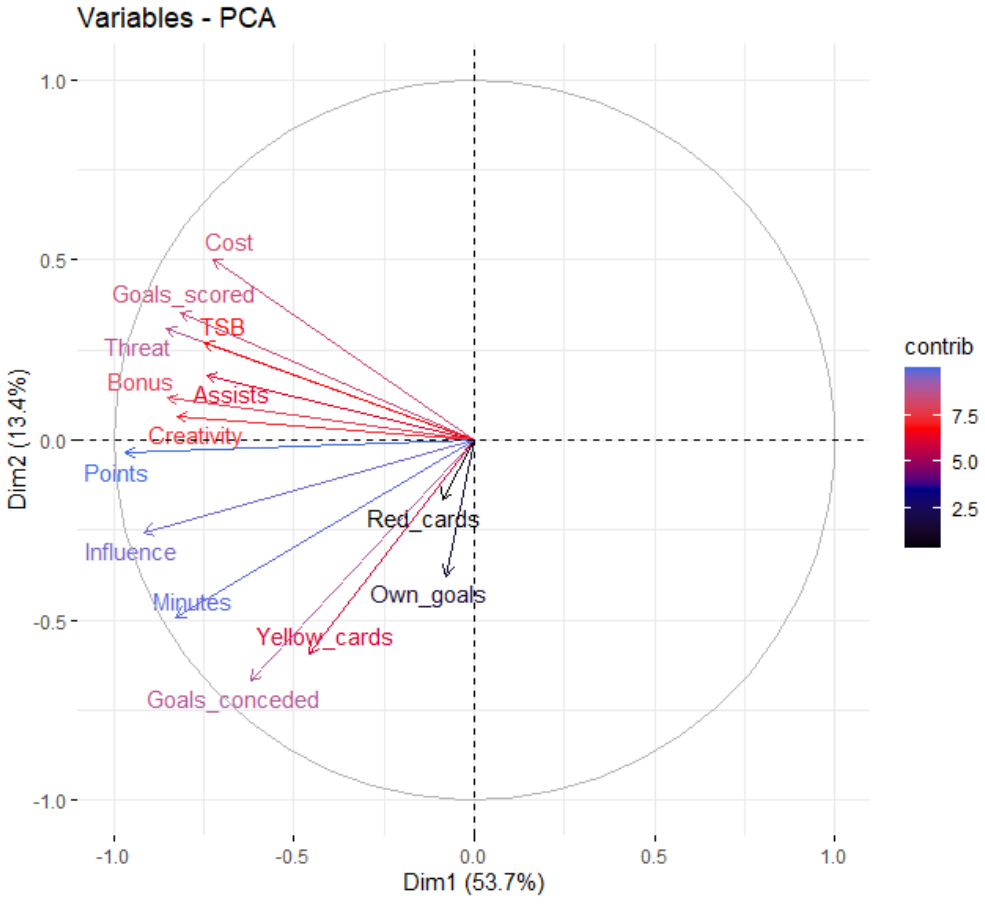

```
> summary(fpl.pca)
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6     PC7
Standard deviation     2.7411 1.3678 1.02186 0.95816 0.78762 0.75046 0.68660
Proportion of Variance 0.5367 0.1336 0.07459 0.06558 0.04431 0.04023 0.03367
Cumulative Proportion  0.5367 0.6703 0.74490 0.81047 0.85478 0.89501 0.92868
                           PC8     PC9    PC10    PC11   PC12    PC13    PC14
Standard deviation     0.51636 0.50426 0.44399 0.36635 0.2994 0.21190 0.10795
Proportion of Variance 0.01904 0.01816 0.01408 0.00959 0.0064 0.00321 0.00083
Cumulative Proportion  0.94773 0.96589 0.97997 0.98956 0.9960 0.99917 1.00000
```

Burada screeplot ve summary'i beraber takip ederek açıklayıcılığı elde etmeye çalıştım. Burada ilk 3 bileşeni incelediğimde; 1. Bileşen yaklaşık %54 açıklayıcılığa sahip, ikinci bileşene baktığımda yaklaşık %14'e yakın bir açıklayıcılığa sahip olmakla beraber en son olarak 3. Bileşene baktığımda yaklaşık %8 açıklayıcılığa sahip olduğunu görüyorum. O zaman ilk 3 bileşeni seçersem toplam açıklayıcılığımın yaklaşık olarak %75'e yakın olacağını görmüş oluyorum. Fakat açıkçası ben 4 bileşen seçmenin screeplot ve summary'i inceledikten sonra daha iyi olacağını düşünüyorum. Çünkü eğer 4 bileşen seçersem yaklaşık olarak %81 gibi bir açıklayıcılığa yaklaşıyorum. Her ne kadar 4. Bileşenin özdeğer'i çok olmasa bile 1'in altında olsada 4. Bileşeni seçmenin burada açıklayılıcık anlamında yararlı olacağını düşünüyorum. Bunun yanı sıra preProcess metodu ile de %81 açıklayıcılığı kontrol ettiğimde buradan da 4 temel bileşen seçmemin uygun olacağı çıktısını elde ettim.

```
> pca2 = preProcess(fpl, method = 'pca', thresh = 0.81)

PCA needed 4 components to capture 81 percent of the variance
```

Temel bileşen analizinden elde ettiğim biplot aşağıdaki gibi oldu.





Ayrıca biplot ile özvektörleride beraber incelemek istedim.

```
> fpl.pca$rotation[,1:4]
                    PC1         PC2          PC3          PC4
Cost          -0.26499598  0.36803152  0.016331619  0.030663078
Creativity    -0.30084497  0.04763925  0.084476038 -0.103493040
Influence     -0.33437177 -0.18748151 -0.011472534 -0.002861455
Threat        -0.31170294  0.22541057 -0.027068365  0.031376138
Goals_conceded -0.22649726 -0.48817946 -0.061525478 -0.095387154
Goals_scored  -0.29754561  0.25854723 -0.085851101  0.061897153
Assists       -0.27106197  0.13191671  0.160745247 -0.015017563
Own_goals     -0.02865884 -0.27614983  0.470712172  0.803813326
Yellow_cards  -0.16720793 -0.43265453  0.004263023 -0.357253671
Red_cards     -0.03255975 -0.12020429 -0.854610760  0.423524973
TSB           -0.27383679  0.19712380 -0.042098269  0.087801911
Minutes       -0.30197911 -0.35927994 -0.010784034 -0.049605173
Bonus         -0.31031375  0.08504255 -0.010109622  0.104116314
Points        -0.35310718 -0.02619817  0.028275314  0.008444883
```

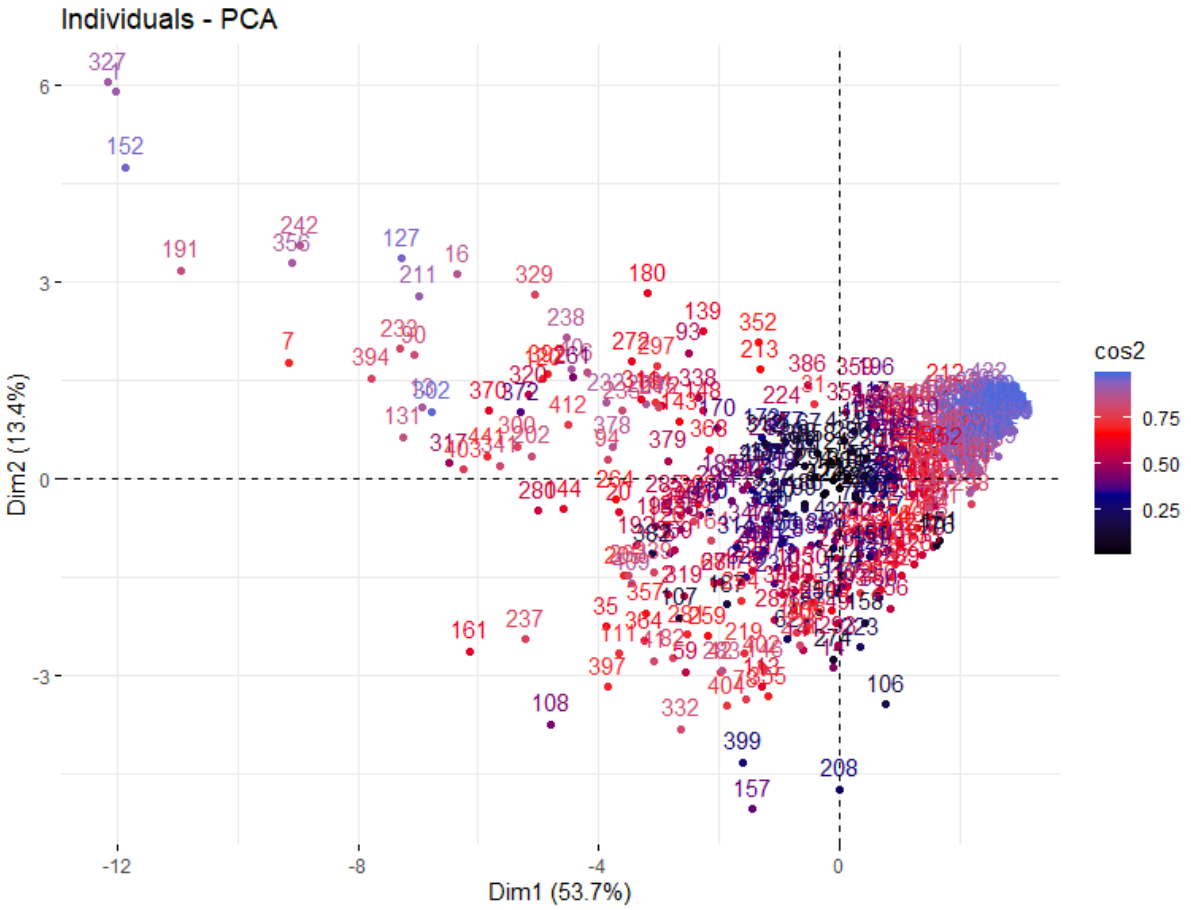
Birinci bileşen için özvektörleri incelediğimde en çok katkıyı sağlayan değerler sırasıyla Points, Influence, Threat, Bonus, Minutes ve Creativity olduğu görülüyor.

İkinci bileşen için özvektörleri incelediğimde en çok katkıyı sağlayan değerlerin sırasıyla Goals_conceded, Yellow_cards, Costs ve Minutes olduğu görülüyor.

Üçüncü bileşen için özvektörleri incelediğimde en çok katkıyı sağlayan değerlerin sırasıyla Red_cards ve Own_goals olduğu görülüyor.

Dördüncü bileşen için özvektörleri incelediğimde en çok katkıyı sağlayan değerlerin sırasıyla Own_goals, Red_cards ve Yellow_cards olduğu görülüyor.

Ben özdeğerleri ve biplot'u incelerken direk veri setindeki değerlerin isimlerini yazdım. Aynı zamanda aşağıdaki grafiğe göre de yorum yapmak istedim:



Öncelikle burada yorum kısmına geçmeden önce ben verilerin tanımına göre şöyle bir ayrım yaptım.

Verilerin tanımına göre Creativity, Influence, Threat, Goals_conceded, Assists, TSB verilerinin takımsal veriler olduğunu gördüm. Yani oyuncunun takım içerisindeki performansını belli eden veriler. Bunun yanında da Goals_scored, Minutes, Points, Bonus, Yellow_cards, Red_cards verilerinin bireysel veriler olduğunu gördüm. Yani oyuncunun direk kendisinin bireysel verileri şeklinde bir ayrım yaptım.

Grafikte yatay eksene göre en soldaki sporculara baktığımda yani 327. , 1. ,152.ve 191. sporcuları incelediğimde Thread verisinin yani oyuncunun bir hedefe yönelik atağının en fazla değerleri aldığını gördüm. Bunun yanı sıra grafiğin en soluna gidildikçe diğer verilerinde max değerlerine yaklaştığını söyleyebilirim. En sağa gidildiğinde ise Costs verisi dışındaki tüm verilerin min değerini aldığını gördüm. Hatta 422. Sporcu grafikte mor bölümün olduğu yerde bulunan bir sporcu, yani grafiğin en sağına yakın bulunan bir sporcu olup bu sporcu için bireysel ve takım değerlerinin hepsinin min değerler olduğunu gördüm. 422. Sporcu için sadece Cost verisi min değerinde değildi fakat min değerine çok yakın bir degere sahip idi. Yani buradan şunu söyleyebilirim yukarıdaki grafikte oyuncu ne kadar sol üste çıkıyor ise o kadar iyi değerlere sahip oluyor. Yukarıdaki grafikte sağa gidildikçe bireysel veriler o kadar düşüyor.

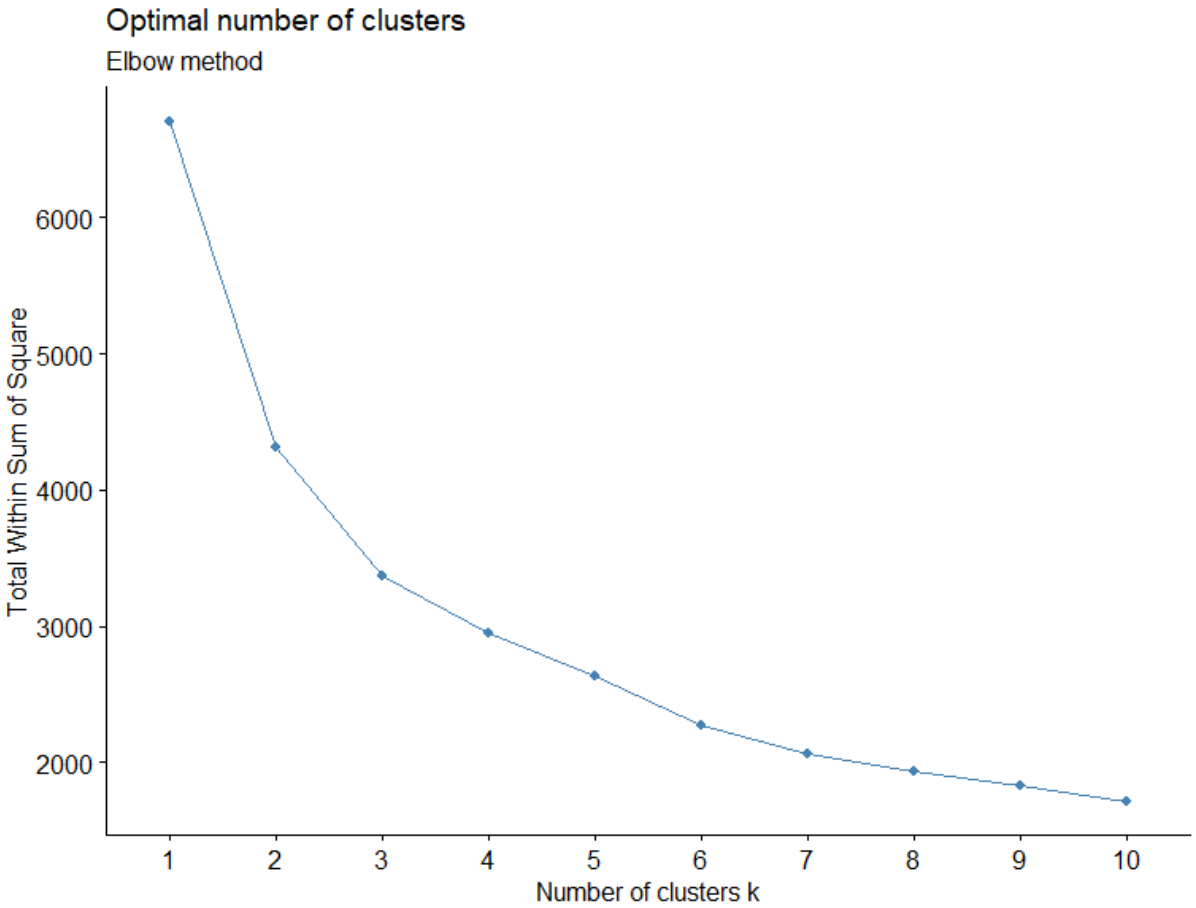
Grafikte dikey eksene göre baktığımda ise önce en alttaki sporculara baktığımda yani 157. , 208. Ve 399. Sporcuları incelediğimde bu sporcuların Minutes, Bonus ve Points verilerinin ortalamanın üzerinde olduğunu gördüm. Fakat Own_goals, Assits, Yellow_cards, Red_cards gibi bireysel verilerinin min değerlerine yaklaştığını gördüm. En üstteki değerleri yani 327., 1. , 152. sporcuları zaten yatay eksende incelemiştim. Bundan dolayı daha net farkı görebilmek adına 180., 352., 139. Sporcuları inceledim. Buradan anladığım kadarıyla grafikte dikey eksene göre ne kadar aşağıya inilirse bireysel oyun değerleri ortalamanın altına düşmüyor fakat takımsal oyun değerlerinin min değerlerine o kadar yaklaştığını gördüm. Hatta grafikte orta bölümde bulunan verilerin çoğunun takımsal verileri ortalamanın altında görünüyor. Yani grafikte dikey eksene göre ne kadar aşağıya inilirse takımsal oyun değerleri o kadar min değerleri alıyor. Bireysel oyun değerleri ise ortalamanın üzerinde kalıyor. Zaten bireysel veriler için de yatay eksene göre sağa gidildikçe düştüğünü söylemiştim.

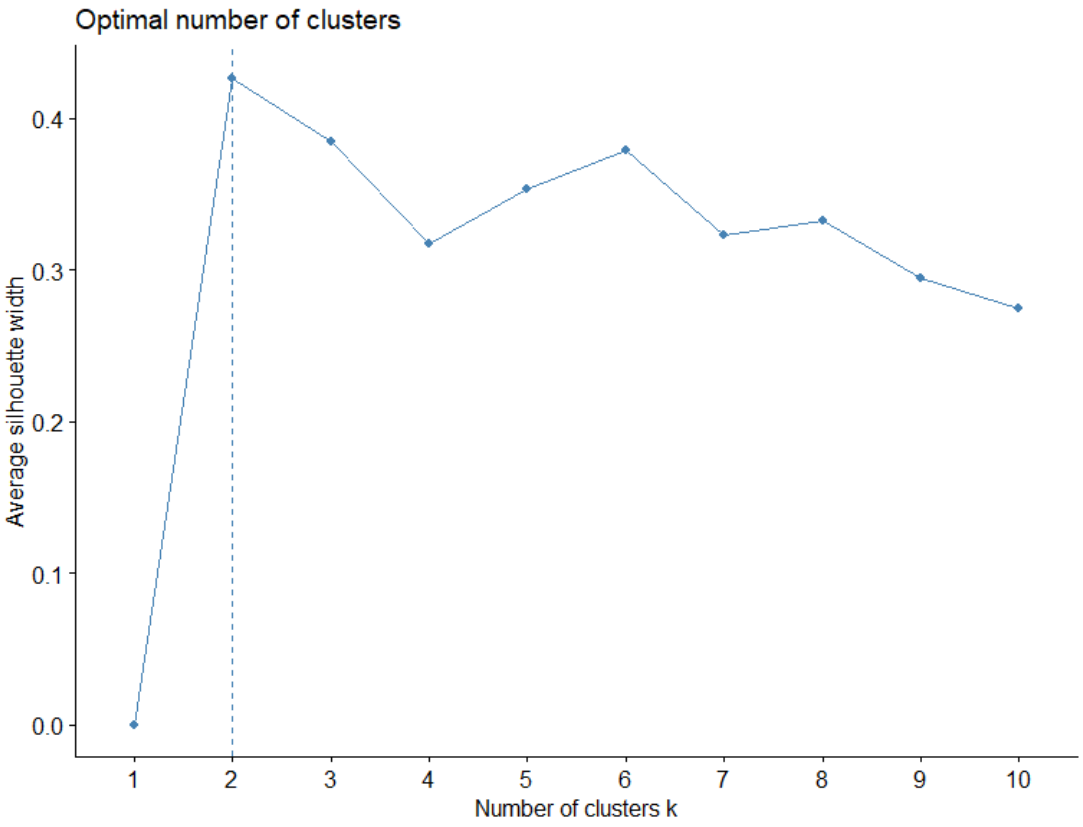
Burada yapılan temel bileşenler analizi ile sporcu profilini en iyi derecede belirten değerlerin takımsal oyun değerleri(Creativity, Influence, Threat, Goals_conceded, Assists, TSB) olduğunu gördüm. Eğer takımsal oyun değerlerinin üzerine sporcu iyi derece bireysel oyun değerlerine de(Goals_scored, Minutes, Points, Bonus, Yellow_cards, Red_cards) sahip ise o zaman yukarıdaki grafikte görüldüğü üzere sol üst köşede yer alıyor. Ayrıca bu sporcuların fiyatları(Cost) da haliyle en yüksek değerlere ulaşıyor. Yani kısacası sporcunun profilini, takım oyununda ne kadar etkili olduğu tam anlamıyla belirleyebiliyor.

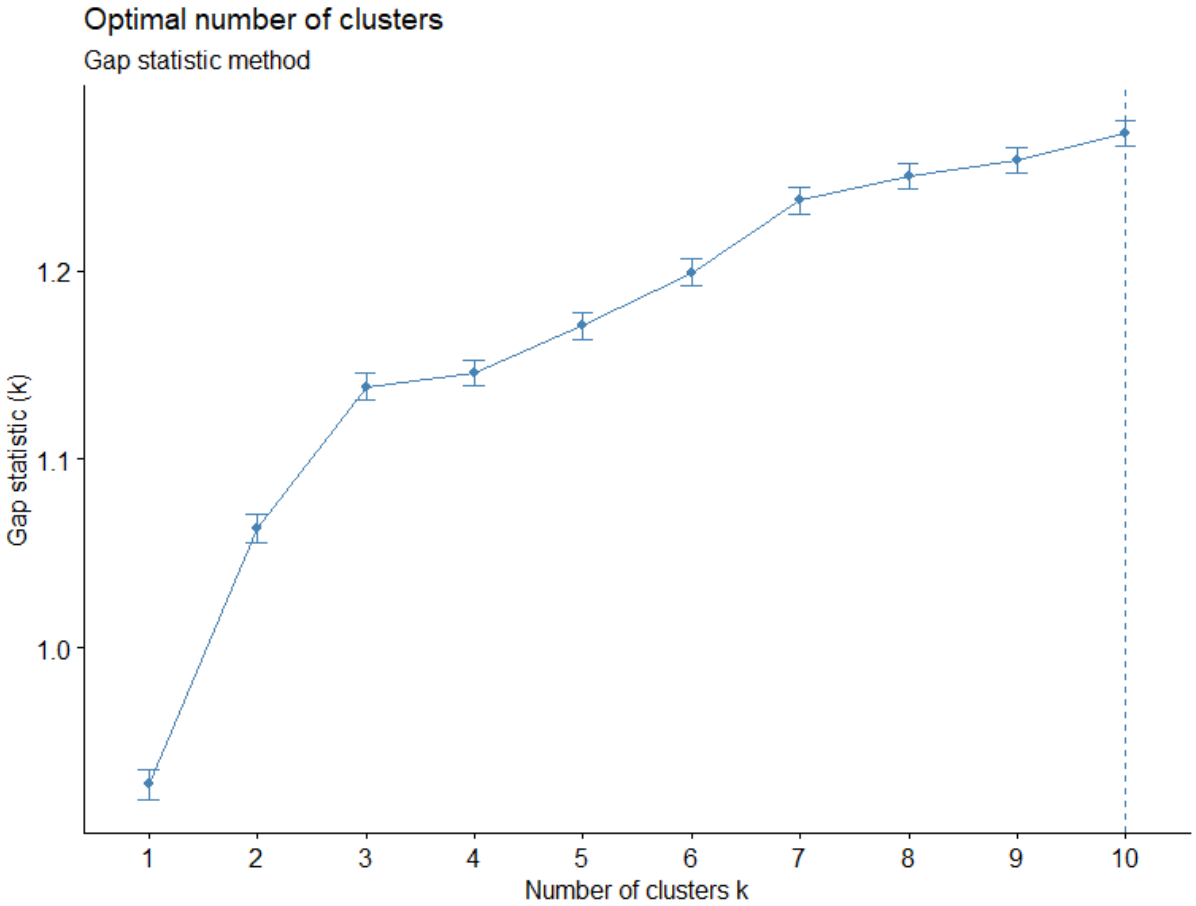
3. VERİ SETİ İÇİN KÜMELEME ANALİZİ

Öncelikle temel bileşen analizinde veri setimden 3 sütunu nümerik olmadığı için çıkarmıştım. Aynısını kümelemeyi daha rahat yapabilmek adına burada da yapacağım. Veri setine temel bileşenlar analizinde olduğu gibi 480 gözlem ve 14 değişken ile kümeleme analizi uygulayacağım. Ayrıca kümeleme analizine başlamadan önce verileri scale ettim. Kümeleme analizinde kmeans yöntemini kullanacağım.Çünkü CLARA yöntemi daha büyük verilerde uygulandığı için bu veride uygulanmasının doğru olmadığını düşünüyorum. Bunun yanı sıra kmedoids ile kümelemeyi denediğimde pek başarılı sonuçlar alamadım. Bu nedenlerden dolayı bu veri seti için en ideal kümeleme yönteminin kmeans kümeleme yöntemi olduğunu düşünüyorum.

İlk olarak ideal küme sayımı belirlemem gerekiyor. Bunun için “Elbow”, “Silhoutte” ve “Gap” metodları ile grafikleri çizdirdim. Sırasıyla grafikler:

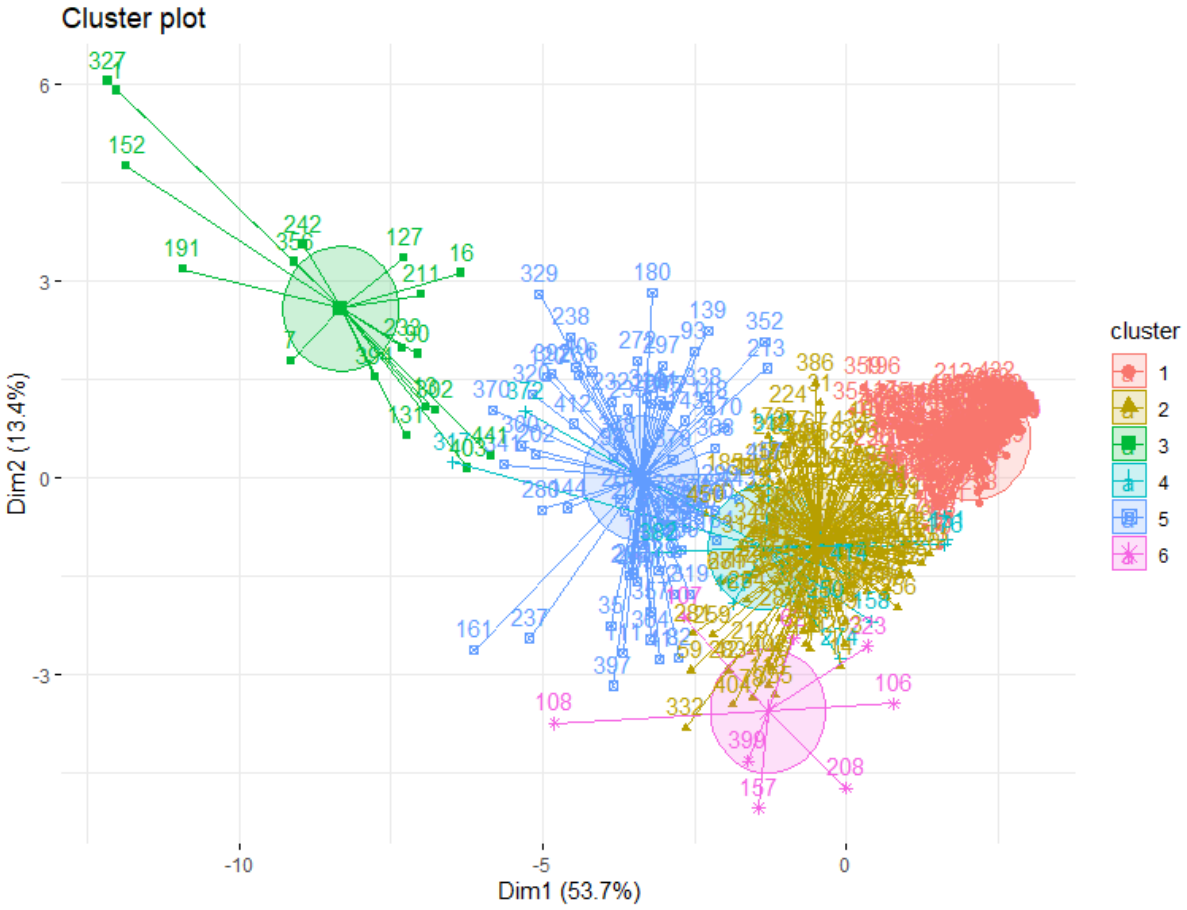



Optimal number of clusters
Average silhouette width
0.0
0.1
0.2
0.3
0.4
1
2
3
4
5
6
7
8
9
10
Number of clusters k


Optimal number of clusters
Gap statistic method
Gap statistic (k)
1.0
1.1
1.2
1
2
3
4
5
6
7
8
9
10
Number of clusters k

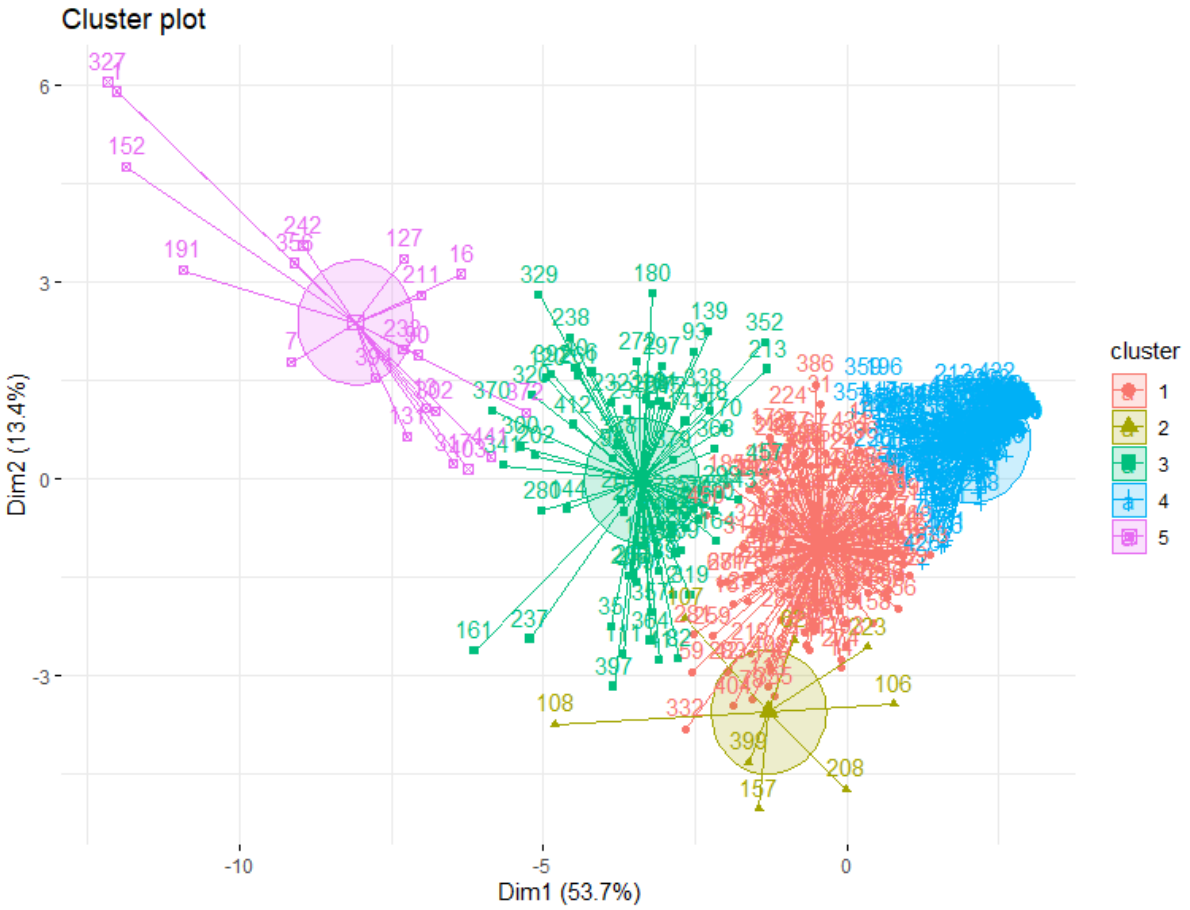
Öncelikle veriyi incelediğimde küme sayısının 2 veya 3 olabileceğini düşünmüyorum. Çünkü temel bileşenler analizinde de gördüğüm kadarıyla çok çeşitli sporcu profilleri var. Yani sporcuları sadece iyi, kötü, orta gibisinden 2 veya 3 kümeye ayıramayacağımızı düşünüyorum. Grafikleri incelediğimde elbow metodunun grafiğinde kırılma noktasının 6 olduğunu düşünüyorum. Silhouette metodunun grafiğine baktığım zaman çoğu zaman olduğu gibi 2'yi gösteriyor. Küme sayımı 2 veya 3 olabileceğini düşünmediğim için silhouette metodunun grafiğinde diğer iki yüksek değer gösteren küme sayısına baktım ve buradan 5 ve 6 değerlerini gördüm. Gap metodunun grafiğine baktığımda ise küme sayısının 10 olabileceğini söylemiş fakat zaten bu veri seti 14 değişkenden oluşuyor olduğu için açıkçası bana 10 küme sayısı çok fazla geldi. Bundan dolayı ilk olarak küme sayısını 6 alarak kmeans kümeleme analizine başlamaya karar verdim.

Küme sayısını 6 olarak aldığımda elde ettiğim grafik aşağıdaki gibi oldu.



Grafiği incelediğimde 2. Ve 4. Kümelerin neredeyse iç içe olduğunu gördüm. Açıkçası 4. Kümenin bulunduğu yer benim istediğim bir yer değil. Kümelerde her ne kadar homojen verileri kümeliyor olsakta bu grafikte bazı veriler kümelerin ortalamalarının çok uzağına çıkmış durumda öyle ki kümenin dışına çıkan birçok veri var. Açıkçası küme sayısını 6 belirlemekte ben çok kararsız kaldım. Bundan dolayı ideal küme sayısını 5 alarakta kümelemeyi denemek istedim. İdeal küme sayısını 5 aldığımda aşağıdaki grafiği elde ettim.

.



Yukarıdaki grafiği incelediğimde kümelerde bulunan elemanların çoğu ideal küme sayısı 6 olduğunda olduğu gibi kümelerin ortalamalarına uzak değerler almış. Bazı elemanların ortalamalara göre uzak değerler almasını küme sayısını arttırmama rağmen değişmediğini gördüm. Bu durumu 10 küme sayısı ile denediğimde de hatta 16 küme sayısı ile denediğimde bile gördüm. (Küme sayısını 10 veya 16 gibi büyük değerler seçmeyeceğim için grafiklerini buraya eklemedim.) Küme sayısını büyük değerler seçmeme rağmen kümelerin ortalamalarından uzak veriler aşırı bir şekilde bulunmaya devam ediyor.

İdeal küme sayısını 6 seçtiğimde 2. Ve 4. Kümelerin neredeyse iç içe olduğunu görmüştüm. Fakat yukarıdaki grafikte böyle bir durum yok ve açıkçası grafiğe göre baksaydım ben ideal küme sayımı 5 olarak seçerdim çünkü burada veriler diğer grafiğe göre daha düzenli bir şekilde kümelenmiş. Burada düzenden kastım kümelerin k=6 ya göre daha az iç içe olması durumu.

Fakat ben yinede istatistiksel olarak 5. Ve 6. Küme sayıları için kontrol yapmaya karar verdim. Çünkü sadece grafiğe göre belirlemek istemedim. Öncelikle Beetween_ss/total_ss oranlarına baktım, bunun dışında dunn index'ine de bakmanın doğru olacağını düşündüm.

Beetween_ss/total_ss oranlarına 5. Ve 6. Küme sayıları için sırasıyla baktığımda aşağıdaki çıktıları elde ettim.

```
## k=5 için

Within cluster sum of squares by cluster:
[1] 946.9405  63.1056 817.4539 368.7926 438.3007
 (between_SS / total_SS =  60.7 %)
```

```
## k=6 için

Within cluster sum of squares by cluster:
[1] 274.2187 703.9794 340.0052 132.3142 757.2670  63.1056
 (between_SS / total_SS =  66.1 %)
```

Burada k=6 için %66.1 sonucunu elde ettim. Fakat Beetween_ss/total_ss oranının genel olarak küme sayısı arttıkça daha yüksek değerler verdiğini gördüm. Bundan dolayı sadece bu değere göre küme sayımı 6 seçmek istemedim. Çünkü örneğin küme sayıma 20 verdiğimde bu oranın anca %80'e ulaşabildiğini gördüm.

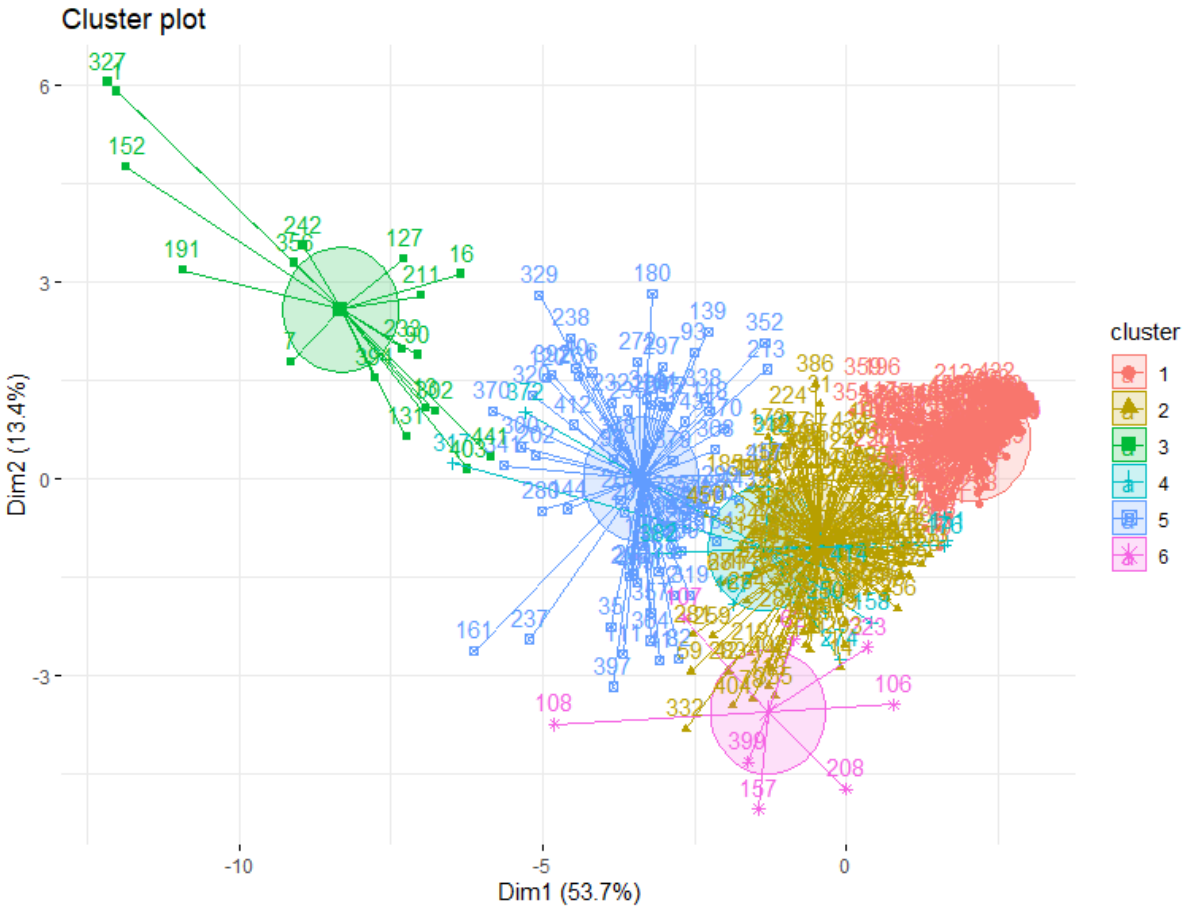
k=6 ve k=5 için dunn index'ine de baktım. Dunn index'e sırasıyla k=5 Ve k=6 için baktığımda aşağıdaki çıktıları elde ettim

```
## k=5 için

> fpl_km_stats_5$dunn
[1] 0.05179618
```

```
## k=6 için

> fpl_km_stats_6$dunn
[1] 0.06226613
```

Diğer ideal küme sayıları için de baktığımda en yüksek dunn index değerini k=6 da iken elde ettim. Bu yüzden ideal küme sayımı 6 olarak belirledim.

Bundan sonra İdeal küme sayısımı 6 olarak belirlediğim için buna göre kümeleri yorumlayacağım. Zaten temel bileşen analizi yaparken verilerin çoğunu incelemiştim. Kümelemeyi yorumlayabilmek adına k=6 için bulduğum grafiği tekrar buraya ekliyorum.



k=6 için bu grafiği elde etmiştim. Buradaki verilere göre kümeleri yorumlayacağım. Aynı zamanda burada yine bireysel oyun değerleri (Goals_scored, Minutes, Points, Bonus, Yellow_cards, Red_cards) ve takımsal oyun değerleri (Creativity, Influence, Threat, Goals_conceded, Assists, TSB) şeklinde sporcu verilerini ayıracağım.

Yorumlamaya başlamadan önce en çok Red_cards verisi içeren kümenin 4. Küme olduğunu söylemem gerek çünkü diğer kümelerde Red_cards verisi genel olarak min değerleri alıyor. Bunu her kümenin açıklamasında yazmak istemedim.

- **1. Küme (Kırmızı renkli küme):** Genel olarak bireysel ve takımsal verileri ortalamanın altında olan sporcular, aynı zamanda fiyatları da ortalamaya yakın veya altında. Goals_conceded, Goals_scored, Assists, Own_goals, Red_cards ve Yellow_cards verileri bu kümede genel olarak min ya da min'e yakın değerleri almış durumda.

- **2. Küme (Sarı renkli küme):** Genel olarak bireysel ve takımsal verileri ortalamanın biraz üzerinde olan sporcular, aynı zamanda fiyatları da ortalamanın çok fazla olmasada üzerinde. Bu kümede özellike Minutes verisi dikkatimi çekti, çünkü Minutes verisi bazı sporcularda ortalamanın çok üzerinde değrleri almış durumda. Ayrıca Goals_scored, Assists, Own_goals ve Yellow_cards verileri 1. Kümeye göre daha az min değerleri almış görünüyor.

- **3. Küme (Yeşil renkli küme):** Bireysel ve takımsal verilerin en iyi olduğu küme bu yüzden bu kümedeki sporcuların aynı zamanda fiyatları da yüksek.

- **4. Küme (Açık mavi renkli küme):** Bu kümedeki oyuncuların da bireysel ve takımsal verileri 2. Küme de olduğu gibi ortalamanın biraz üzerinde hatta bazı oyuncular (372. oyuncu) burada 2. Küme de bulunanlardan daha iyi fakat bu kümenin özellikle Red_cards verisinin fazla olduğu sporcuları içeren bir küme olduğunu söyleyebilirim. Sanırsam kırmızı kart sayısı fazla olduğu için haliyle buradaki oyuncuların fiyatı ortalamanın altında veya iyi değerlere sahip olmasına rağmen fiyatı ortalamanın çok yukarısında değil.

- **5. Küme (Koyu mavi renkli küme):** Bireysel ve takımsal verilerin en iyi olmasada iyi değerlere sahip olduğu küme, genel olarak veriler incelediğinde kendisini ortaya çıkarabilen oyuncular bu kümede yer alıyor. Bu kümedeki oyuncuların genel olarak fiyatları ortalamanın çok üstünde değil. Fakat Minutes verileri max veya max değerlere yakın değerler almış durumda. Yani ucuz bir fiyata ideal bir takım kurulması için bu kümedeki oyuncular incelenilebilir. Bence bu kümeye fiyat performans kümeside denilebilir. Ayrıca bazı oyuncuların Minutes değerlerinin bu kümede max değerlerini aldığı veya yakınlaştığını söylemiştim. Buradan bu küme içerisinde yıllarını futbola vermiş tecrübeli kişilerin de olduğunu söyleyebilirim.

- **6. Küme (Pembe renkli küme):** Bu kümedeki oyuncuların bireysel verileri ortalamanın üzerinde değerlere sahip hatta bazı oyuncuların Minutes değerleri ortalamanın çok üzerinde bulunuyor. Ayrıca aynı şekilde bu kümedeki oyuncuların takımsal verileri de ortalamanın genel olarak üzerinde. Fakat bu kümedeki oyuncuların fiyatı aşırı derecede verilerine göre düşük durumda. Ben burada yaş etkenini düşündüm. Büyük ihtimal buradaki oyuncular artık futbol oynayabilecek yaşı geçmiş sporcular diye düşünüyorum. Bundan dolayı performansları ortalama üzerinde olsa bile takımlar riske alıp yüksek bir fiyata bu oyuncuları satın almak istememiş olabilir.

Açıkçası başlangıçta ben ideal küme sayısını 5 seçtiğimde kümelerin karışmadığından dolayı daha iyi bir şekilde kümelenebileceğini düşünmüştüm. Fakat ideal küme sayısını 6 seçip istatistiksel açıdan kontrolü sağladığımda küme sayımı 6 seçmenin daha iyi olacağını görmüş oldum. Başlangıçta k=6 için 2. Ve 4. Kümelerin iç içe olduğunu görmüştüm. Fakat veriler incelendiğinde gerçekten 4. Kümede Red_Cards verisinin ayırıcı bir etken olduğunu görmüş oldum.

Ayrıca dunn index dışında silhouette katsayısınıda incelemek istemiştim fakat kmeans kullandığım için silhouette katsayısında hata aldım. Bundan dolayı dunn index'e biraz fazla yoğunlaşmamak adına beetween_ss/total_ss oranlarınıda dikkate almaya çalıştım. Burada küme sayısını 5 yerine 6 seçerek daha doğru bir sonuç elde ettiğimi gördüm.