DSM 5008 Take Home – I

Fırat Melih Yılmaz

Gerekli Paketlerin ve Veri Setinin Yüklenmesi

```
library(tidyverse)  # Easily Install and Load the 'Tidyverse'
library(GGally)  # Extension to 'ggplot2'
library(knitr)  # A General-Purpose Package for Dynamic Report Generation in R
library(ggcorrplot)  # Visualization of a Correlation Matrix using 'ggplot2'
library(factoextra)  # Extract and Visualize the Results of Multivariate Data Analyses
library(cluster)  # 'Finding Groups in Data': Cluster Analysis Extended Rousseeuw et al.
library(pastecs)  # Package for Analysis of Space-Time Ecological Series
library(reshape2)  # Flexibly Reshape Data: A Reboot of the Reshape Package
library(clValid)  # Validation of Clustering Results
library(naniar)  # Data Structures, Summaries, and Visualisations for Missing Data
library(DEGreport)  # Report of DEG analysis
library(scatterplot3d)  # 3D Scatter Plot
library(ggfortify)  # Data Visualization Tools for Statistical Analysis Results
library(NbClust)  # Determining the Best Number of Clusters in a Data Set
library(gridExtra)  # Miscellaneous Functions for 'Grid' Graphics
library(magrittr)  # A Forward-Pipe Operator for R
library(kableExtra)  # Construct Complex Table with 'kable' and Pipe Syntax
library(qgraph)  # Graph Plotting Methods, Psychometric Data Visualization and Graphical Model Estimati
library(ggdendro)  # Create Dendrograms and Tree Diagrams Using 'ggplot2'
library(clustertend)  # Check the Clustering Tendency
library(flexclust)  # Flexible Cluster Algorithms
```

Veri setinde bulunan değişkenler aşağıdaki gibidir:

- Name : Player Name
- Team : Team Name (3 letter abbreviation)
- Position : Player Position (3 letter abbreviation)
- Cost : Average Cost of the player
- Creativity : Assesses player performance in terms of producing goalscoring opportunities for others.
- Influence : This evaluates the degree to which that player has made an impact on a single match or throughout the season.
- Threat : A value that examines a player's threat on goal
- Goals_conceded: Number of goals conceded while the player was on the field
- Goals_scored : Goals scored by the player
- Assists : Assists provided by the player
- Own_goals : Own goals scored by the player
- Yellow_cards : Yellow cards received by the player
- Red_cards : Red cards received by the player
- TSB : % of teams in which the player has been selected
- Minutes : Minutes played by the player
- Bonus : Bonus points received by the player
- points : Points scored by the player

```r
# reading data
raw_lines <- readLines("data/FPL.csv")  # reading data by line
raw_lines <- gsub("(^\"|\"$)", "", raw_lines)  # removing the outer quotes and then using double quotes
raw_data <- read.csv(textConnection(raw_lines), quote = "\"\"",
    header = TRUE, row.names = 1)  # reading data
```

S1: Tanımlayıcı istatistikleri elde ederek, yorumlayınız.

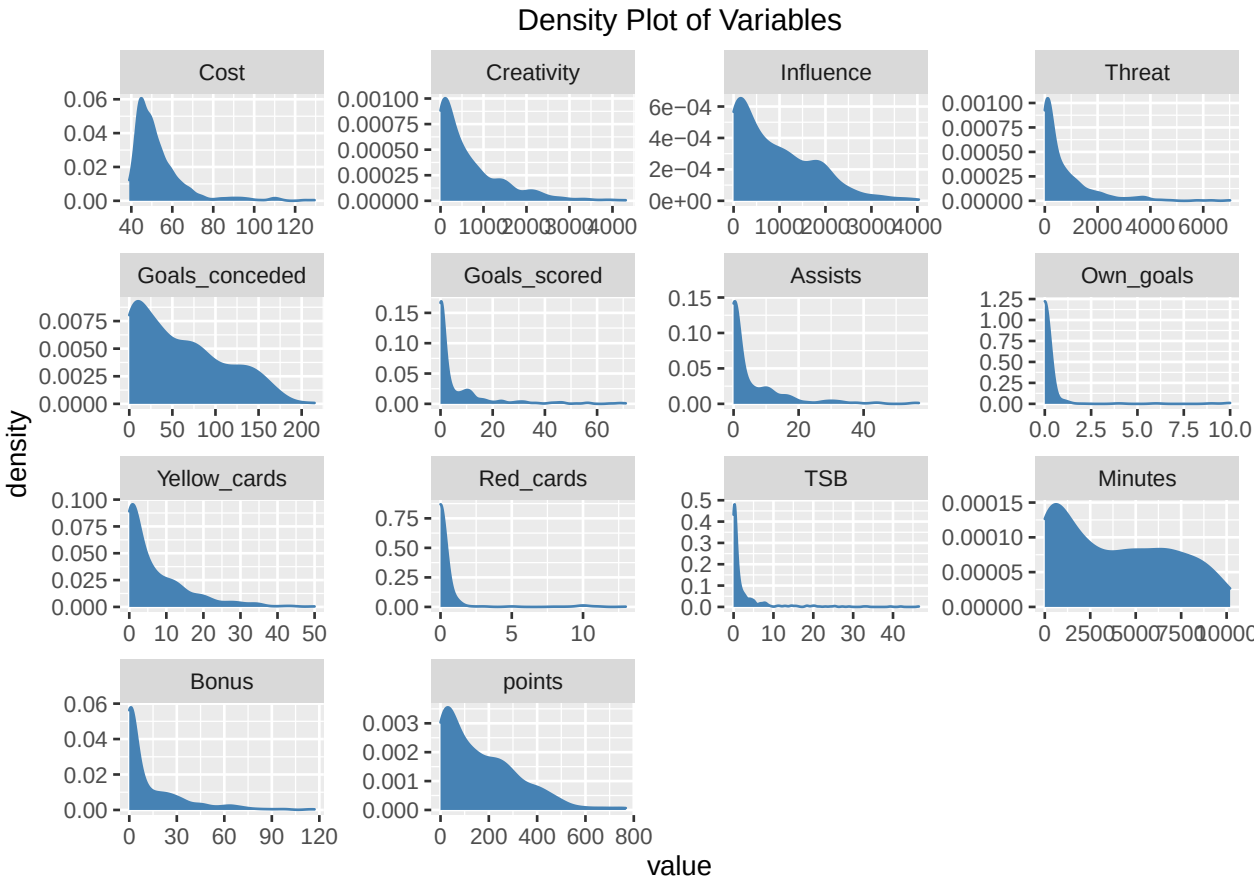
```r
dim(raw_data)  # dimensions: 480 x 16 matrix
str(raw_data)  # structure of data
summary(raw_data)  # summary of data
```

Table 1: Summary Table of Raw Data

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
Cost	480	52.958	12.860	39.000	44.625	56.020	129.500
Creativity	480	649.015	758.611	0.000	78.925	943.625	4,310.300
Influence	480	894.396	839.480	0	151	1,454.7	4,033
Threat	480	713.362	998.805	0	67	973.2	7,018
Goals_conceded	480	59.521	50.845	0	14	96	215
Goals_scored	480	5.952	11.540	0	0	7	71
Assists	480	5.506	9.338	0	0	8	57
Own_goals	480	0.196	1.179	0	0	0	10
Yellow_cards	480	6.725	8.601	0	0	10	50
Red_cards	480	0.338	1.588	0	0	0	13
TSB	480	2.681	5.824	0.000	0.130	2.360	46.490
Minutes	480	3,877.962	3,112.595	0	895.8	6,549.2	10,192
Bonus	480	13.113	20.214	0	0	19.2	117
points	480	163.906	154.603	0	34	254.8	767

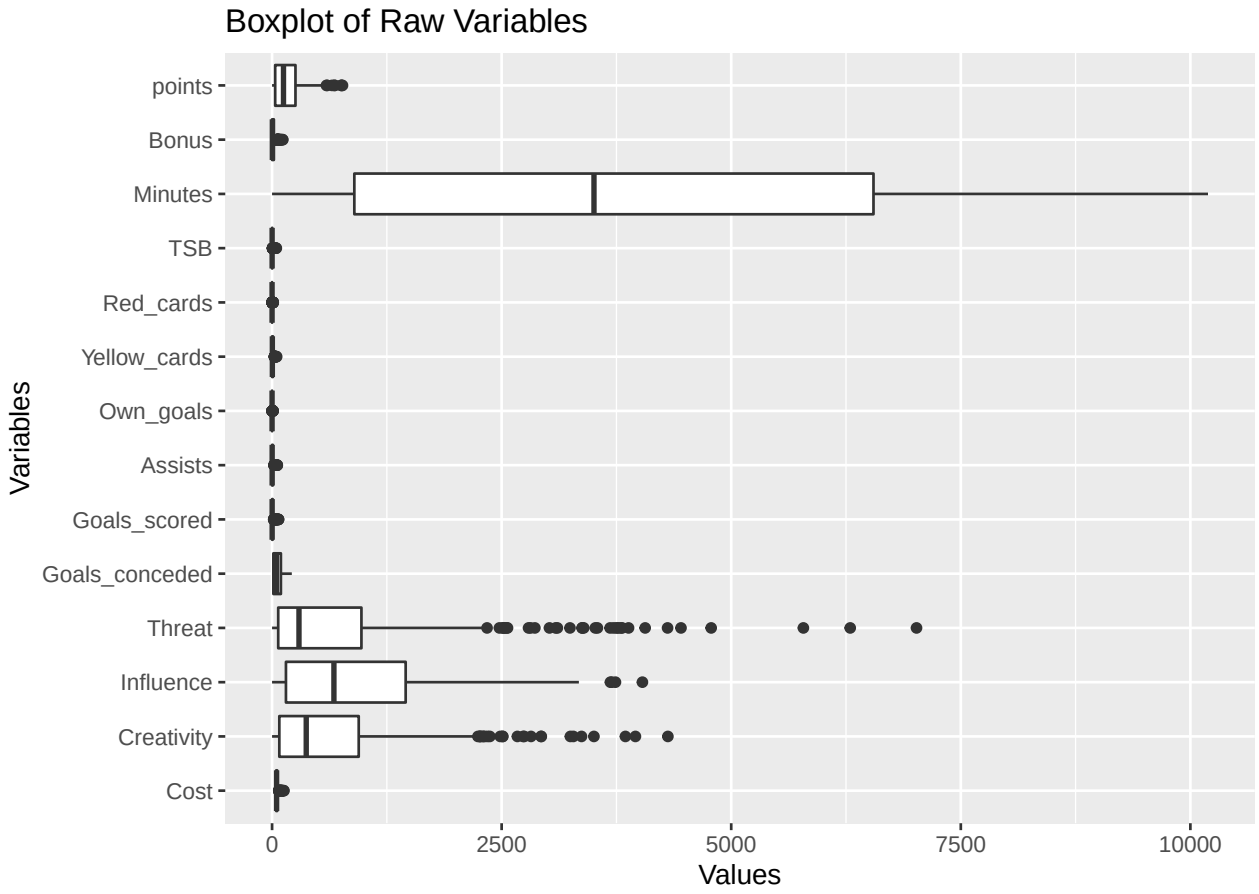
Veri setindeki `Teams` ve `Position` değişkenleri dışındaki değişkenlere baktığımızda farklı birimlerden geldikleri için değişkenlerin ortalama, standard sapma ve range (aralık) değerleri birbirinden çok farklıdır. Verileri daha iyi tanımak için yoğunluk ve boxplot grafiklerini çizdirilmiştir.

```r
# Density Plot
raw_data %>%
  melt(
    # ID variables - all the variables to keep but not split apart on
     id.vars= c('Team', 'Position'),
     # The source columns
     measure.vars=c(colnames(raw_data)[-c(1,2)]),
) %>% # set scales free since all variables in different range
  ggplot(aes(value)) +geom_density(color = 'steelblue', fill = 'steelblue') +
  facet_wrap(~variable, scales = "free") +
  labs(title = 'Density Plot of Variables') +
  theme(plot.title=element_text(color='black',hjust=0.5,size=12))
```



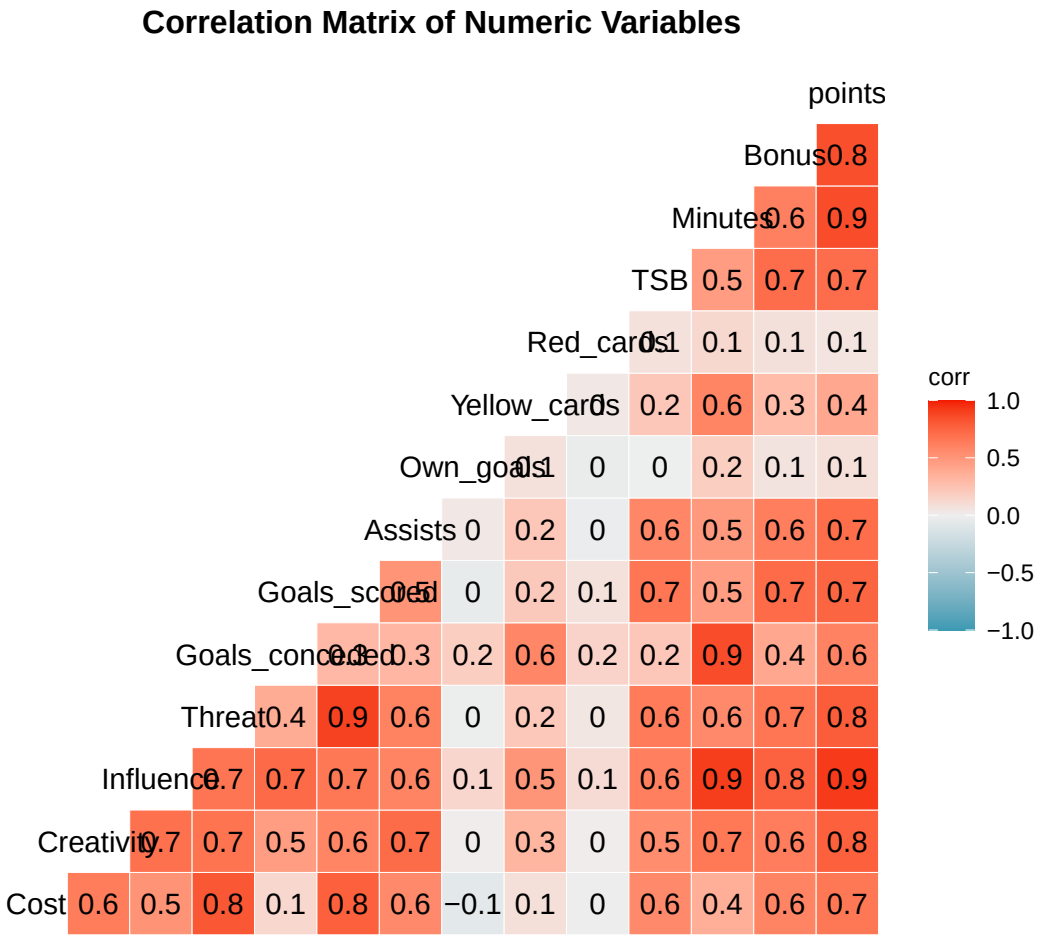
Değişkenlelerin dağılımına baktığımızda ilk göze çarpan, verilerin sağdan çarpık olduğu ve kendi içinde farklı tepelenmelere sahip olduğudur. Bu durumu oyuncuların genel olarak düşük performas değerlerine sahip olduğu ve kendi içlerinde farklı kümelenmeler yarattığı şeklinde yorumlayabiliriz. Bunun nedeni veri setimizdeki oyuncular defans (187) ve orta saha (214) ağırlıklı olmasından kaynaklanmaktadır. Veri setindeki değişkenliği daha iyi görmek ve veri setinde aykırı gözlemlerin bulunup bulunmadığını tespit etmek için boxplot grafiğini çizdirelim.

```
ggplot(stack(raw_data[, -c(1, 2)]), aes(x = ind, y = values)) +
    geom_boxplot() + coord_flip() + labs(title = "Boxplot of Raw Variables",
    x = "Variables", y = "Values")
```



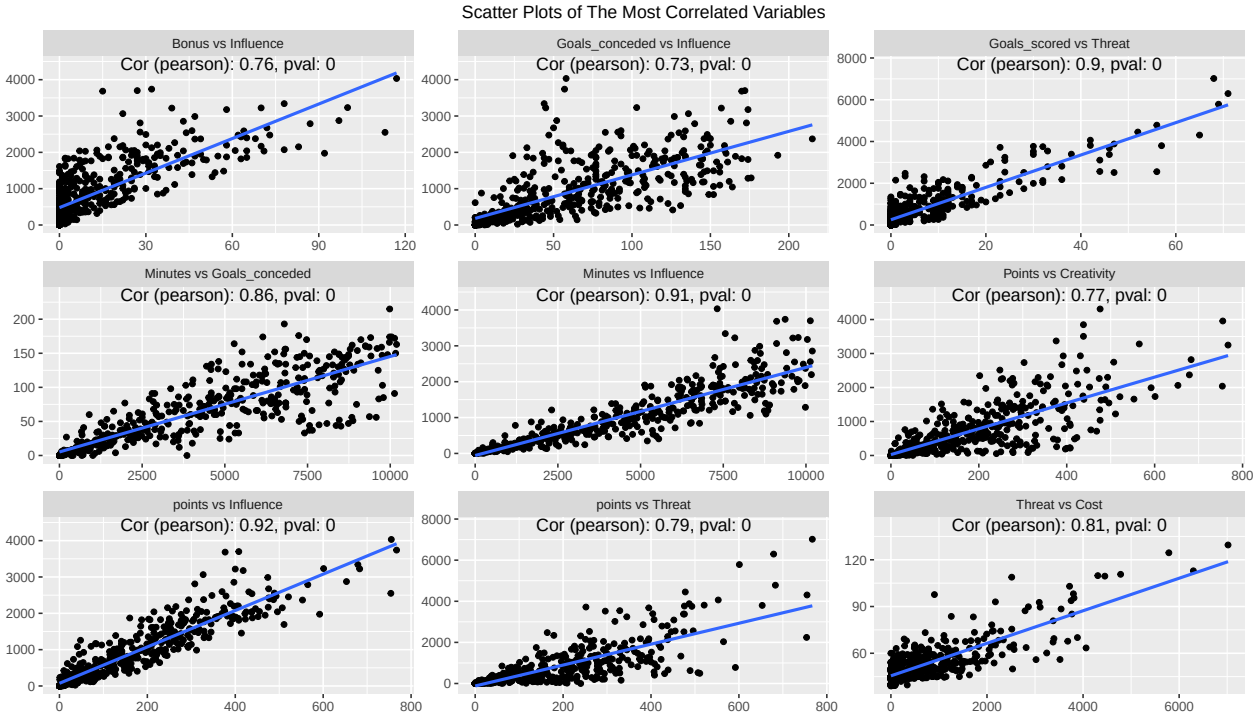
Boxplot'ı incelediğimizde veri seti içinde değişkenliği en yüksek olan değişken `Minutes` değişkenidir. `Minutes` ve `Goals Conceded` değişkeni dışındaki değişkenlerde ise aykırı değer bulunmaktadır. Sonuç olarak çarpık ve aykırı değerlerin bulunduğu bu veri setinde yapılacak kümeleme analizi, verileri en optimum şekilde kümelemeyecektir. Veri setindeki değişkenlerin birbiri ile ne derecede korele olduğunu analiz etmek için korelasyon matrisi hesaplanmıştır.

```
ggcorr(data = raw_data[, -c(1, 2)], name = "corr", label = TRUE,
    method = "complete.obs") + labs(title = "Correlation Matrix of Numeric Variables") +
    theme(plot.title = element_text(face = "bold", color = "black",
        hjust = 0.5, size = 12))
```



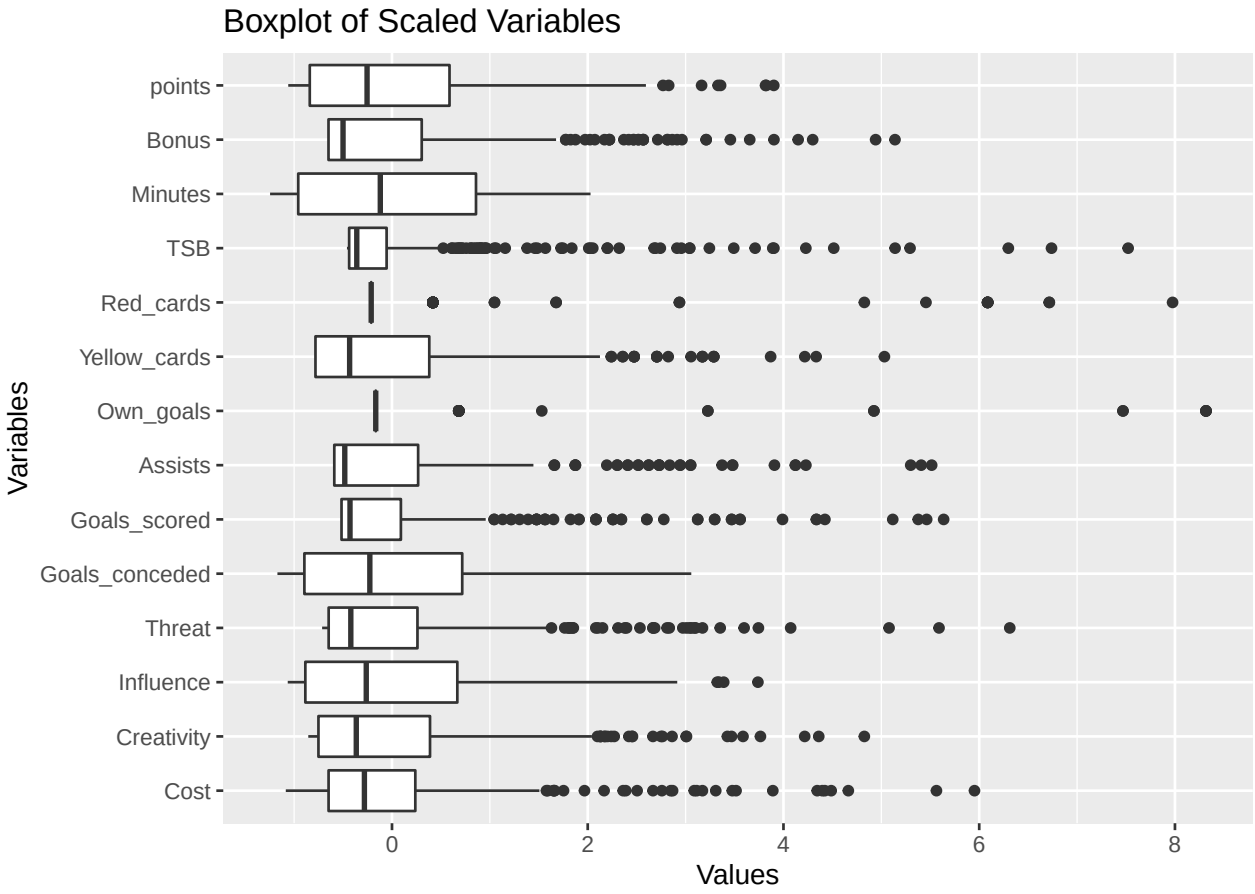
Korelasyon matrisine baktığımızda değişkenlerin birbiri ile yüksek ve güçlü derecede korelasyon içinde olduğu görülmektedir. Bu durum bize veri seti için Temel Bileşenler Analizinin yapılmasını işaret etmektedir. Veri setindeki korelasyon değeri 0.70 ve üzeri olan değişkenlerin saçılım grafiğini çizdirerek ilişkinin derecesini ve kuvvetini daha net incelenmiştir.

```
list(transmute(raw_data, x = Threat, y = Cost, dataset = "Threat vs Cost"),
    transmute(raw_data, x = points, y = Creativity, dataset = "Points vs Creativity"),
    transmute(raw_data, x = Minutes, y = Influence, dataset = "Minutes vs Influence"),
    transmute(raw_data, x = Bonus, y = Influence, dataset = "Bonus vs Influence"),
    transmute(raw_data, x = points, y = Influence, dataset = "points vs Influence"),
    transmute(raw_data, x = Goals_scored, y = Threat, dataset = "Goals_scored vs Threat"),
    transmute(raw_data, x = points, y = Threat, dataset = "points vs Threat"),
    transmute(raw_data, x = Goals_conceded, y = Influence, dataset = "Goals_conceded vs Influence"),
    transmute(raw_data, x = Minutes, y = Goals_conceded, dataset = "Minutes vs Goals_conceded")) %>%
    bind_rows() %>% ggplot(aes(x, y)) + geom_point() + geom_smooth(method = "lm",
    se = FALSE) + geom_cor(method = "pearson") + labs(title = "Scatter Plots of The Most Correlated Var
    x = "", y = "") + theme(plot.title = element_text(color = "black",
    hjust = 0.5, size = 12)) + facet_wrap(~dataset, scales = "free")
```



Saçılım grafiğinine baktığımızda seçilen yüksek koreasyolu değerlerin birbiri ile doğrusal ve güçlü bir ilişki içindedir ayrıca koreasyon değeri için hesaplanan `p` değeri 0.05'den küçük olması nedeniyle anlamlıdır. Sonuç olarak daha ileriki analiz için verilerin `scale` edilmesi ve Temel Bileşenler analizinin uygulanması gerekmektedir. Temel bileşenler analizine geçmeden önce veriler `scale` edilmiş ve bu işlemin ardından verilerin nasıl göründüğünü görmek için boxplot çizdirilmiştir.

```
scaled_data <- scale(raw_data[, -c(1, 2)], center = TRUE, scale = TRUE)
scaled_data <- data.frame(raw_data[, c(1, 2)], scaled_data)
ggplot(stack(scaled_data[, -c(1, 2)]), aes(x = ind, y = values)) +
    geom_boxplot() + coord_flip() + labs(title = "Boxplot of Scaled Variables",
    x = "Variables", y = "Values")
```



Verileri `Scale` ettikten sonraki elde ettiğimiz boxplot'ı incelediğimizde veri setindeki bütün değişkenlerde aykırı değişkenlerin olduğu görülmektedir. Bu durum bize kümeleme analizinde aykırı gözlemlere dayanıklı kümeleme algoritmasının seçilmesini işaret etmektedir.

S2: Temel bileşenler analizi uygulayarak, yorumlayınız.

Veri setindeki değişkenler birbiri ile yüksek derecede ilişkili olduğu için bu ilişkiyi ortadan kaldırmak için Temel Bileşenler Analizi uygulanmalıdır. Temel Bileşenler analizine geçmeden önce değişkenlere temel bileşen analizinin yapılıp yapılamayacağını test etmek için *Kaiser-Meyer-Olkin (KMO) Test* yapılmıştır ardından değişkenlere ait özdeğerleri ve bu özdeğerlerin veri setindeki değişkenliği ne derecede açıkladığını incelenmiştir.

```
# kmo statistics
kmo <- function(data) {
    library(MASS)
    X <- cor(as.matrix(data))
    iX <- ginv(X)
    S2 <- diag(diag((iX^-1)))
    AIS <- S2 %*% iX %*% S2  # anti-image covariance matrix
    IS <- X + AIS - 2 * S2  # image covariance matrix
    Dai <- sqrt(diag(diag(AIS)))
    IR <- ginv(Dai) %*% IS %*% ginv(Dai)  # image correlation matrix
    AIR <- ginv(Dai) %*% AIS %*% ginv(Dai)  # anti-image correlation matrix
    a <- apply((AIR - diag(diag(AIR)))^2, 2, sum)
    AA <- sum(a)
    b <- apply((X - diag(nrow(X)))^2, 2, sum)
```

```
    BB <- sum(b)
    MSA <- b/(b + a)  # indiv. measures of sampling adequacy
    AIR <- AIR - diag(nrow(AIR)) + diag(MSA)  # Examine the anti-image of the correlation matrix. That
    kmo <- BB/(AA + BB)  # overall KMO statistic
    # Reporting the conclusion
    if (kmo >= 0 && kmo < 0.5) {
        test <- "The KMO test yields a degree of common variance unacceptable for FA."
    } else if (kmo >= 0.5 && kmo < 0.6) {
        test <- "The KMO test yields a degree of common variance miserable."
    } else if (kmo >= 0.6 && kmo < 0.7) {
        test <- "The KMO test yields a degree of common variance mediocre."
    } else if (kmo >= 0.7 && kmo < 0.8) {
        test <- "The KMO test yields a degree of common variance middling."
    } else if (kmo >= 0.8 && kmo < 0.9) {
        test <- "The KMO test yields a degree of common variance meritorious."
    } else {
        test <- "The KMO test yields a degree of common variance marvelous."
    }

    ans <- list(overall = kmo, report = test, individual = MSA,
        AIS = AIS, AIR = AIR)
    return(ans)
}
kmo(data = scaled_data[, -c(1, 2)])$overall
```

```
## [1] 0.8038611
```

Elde edilen *Kaiser-Meyer-Olkin (KMO) Test* sonucuna göre (0.8038611) veri setimiz temel bileşenler analizi için uygundur.

```
scaled_df_corr_matrix <- cor(scaled_data[, -c(1, 2)], method = "pearson",
    use = "complete.obs")
scaled_df_eigen <- eigen(x = scaled_df_corr_matrix)
print(scaled_df_eigen[1])
```

```
## $values
##  [1] 7.51355227 1.87078775 1.04420202 0.91806270 0.62034755 0.56318775
##  [7] 0.47141291 0.26662810 0.25427460 0.19712484 0.13421007 0.08965748
## [13] 0.04489954 0.01165241
```

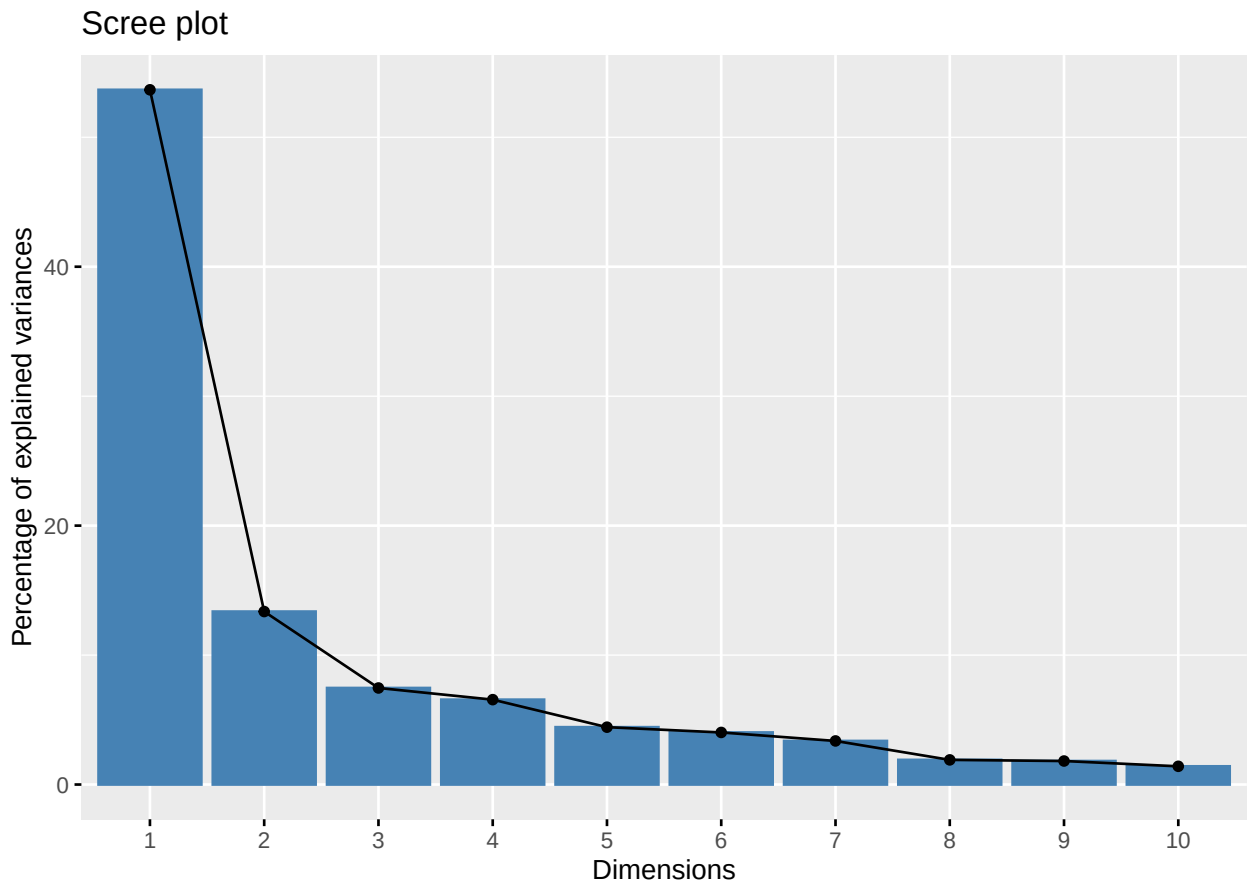
Temel Bileşenler analizini özdeğer vektörleri üzerinden yapacağımız zaman öz değerlerin birden büyük olanlarının seçilmesi önerilmektedir. Buna göre veri setimizde bu şartı sağlayan üç adet öz vektörün olduğu görülmektedir. Bu analizi toplam açıklanan varyans üzerinden teğit edecek olursak:

```
scaled_df_var <- scaled_df_eigen$values/sum(scaled_df_eigen$values)
scaled_data_cumsum_var <- cumsum(scaled_df_var)
tibble(.rows = 1:14, eigenValue = scaled_df_eigen$values, Var = scaled_df_var,
    cumsumVar = scaled_data_cumsum_var)
```

Açıklanan varyans üzerinden gittiğimizde ise ilk temel bileşenin veri setindeki değişkenliğin %54'nü açıkladığı görülmektedir. Kümülatif açıklanan varyansa göz attığımızda ise altıncı temel bileşenden itibaren açıklanan varyansın çok az bir oranla arttığı görülmektedir. Ayrıca açıklanan varyansın en az %75'i olması gerektiği şartı üçüncü temel bileşenden itibaren sağlanmaktadır ve bu sonuç özdeğer vektörleri ile elde ettiğimiz sonuçla örtüşmektedir.

```
scaled_df_pca <- prcomp(x = scaled_data[, -c(1, 2)])
fviz_screeplot(scaled_df_pca, ggtheme = theme_gray())
```

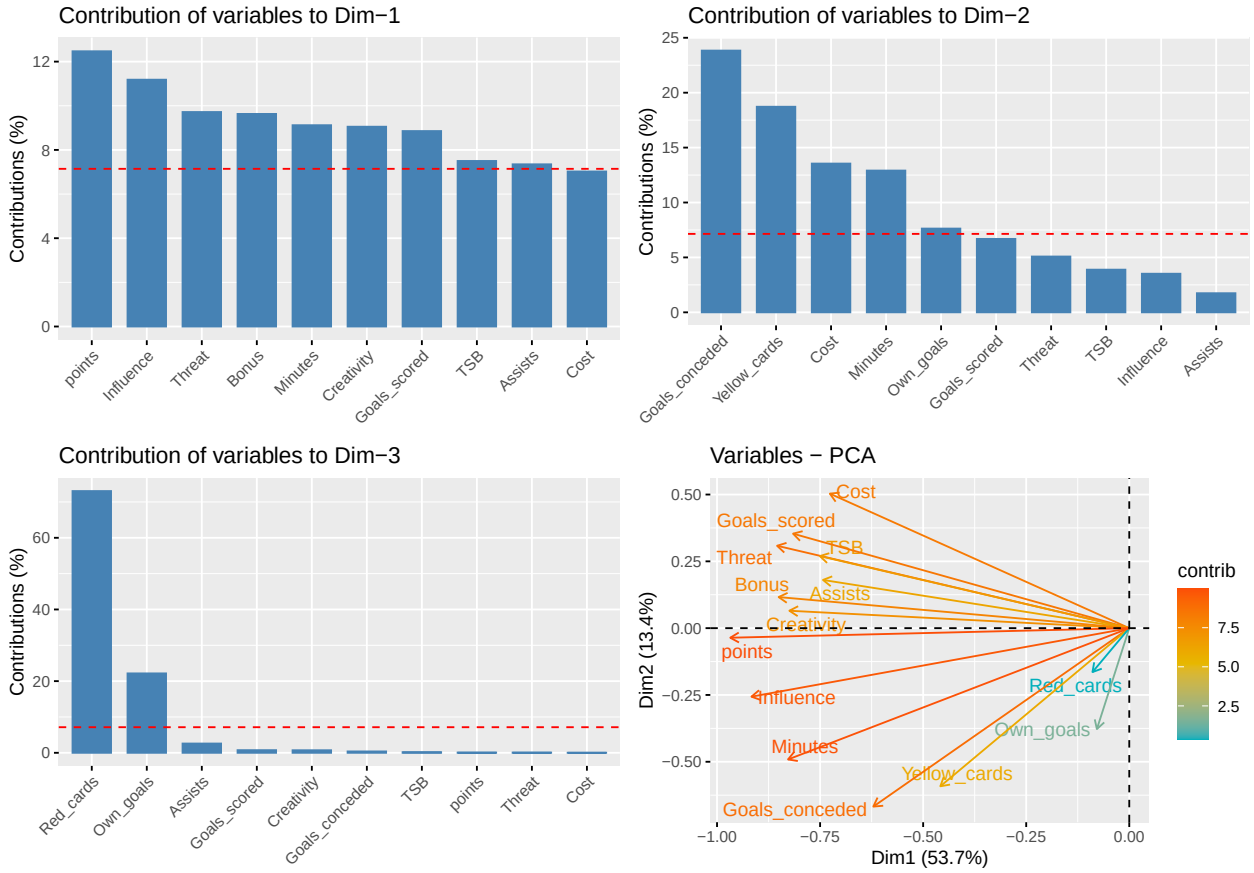
	eigenValue	Var	cumsumVar
1	7.51	0.54	0.54
2	1.87	0.13	0.67
3	1.04	0.07	0.74
4	0.92	0.07	0.81
5	0.62	0.04	0.85
6	0.56	0.04	0.90
7	0.47	0.03	0.93
8	0.27	0.02	0.95
9	0.25	0.02	0.97
10	0.20	0.01	0.98
11	0.13	0.01	0.99
12	0.09	0.01	1.00
13	0.04	0.00	1.00
14	0.01	0.00	1.00



Temel bileşen Analizinden sonra elde ettiğimiz screeplotta üçüncü temel bileşenden itibaren bir dirsek oluştuğu için yukarıda yaptığımız analizi destekler niteliktedir. Değişkenlerin temel bileşenlere yaptığı katkıyı ve birbiri ile ilişkilerini görselleştirecek olursak:

```
# Contributions of variables to PC1
pca_p1 <- fviz_contrib(scaled_df_pca, choice = "var", axes = 1,
    top = 10, ggtheme = theme_gray())
```

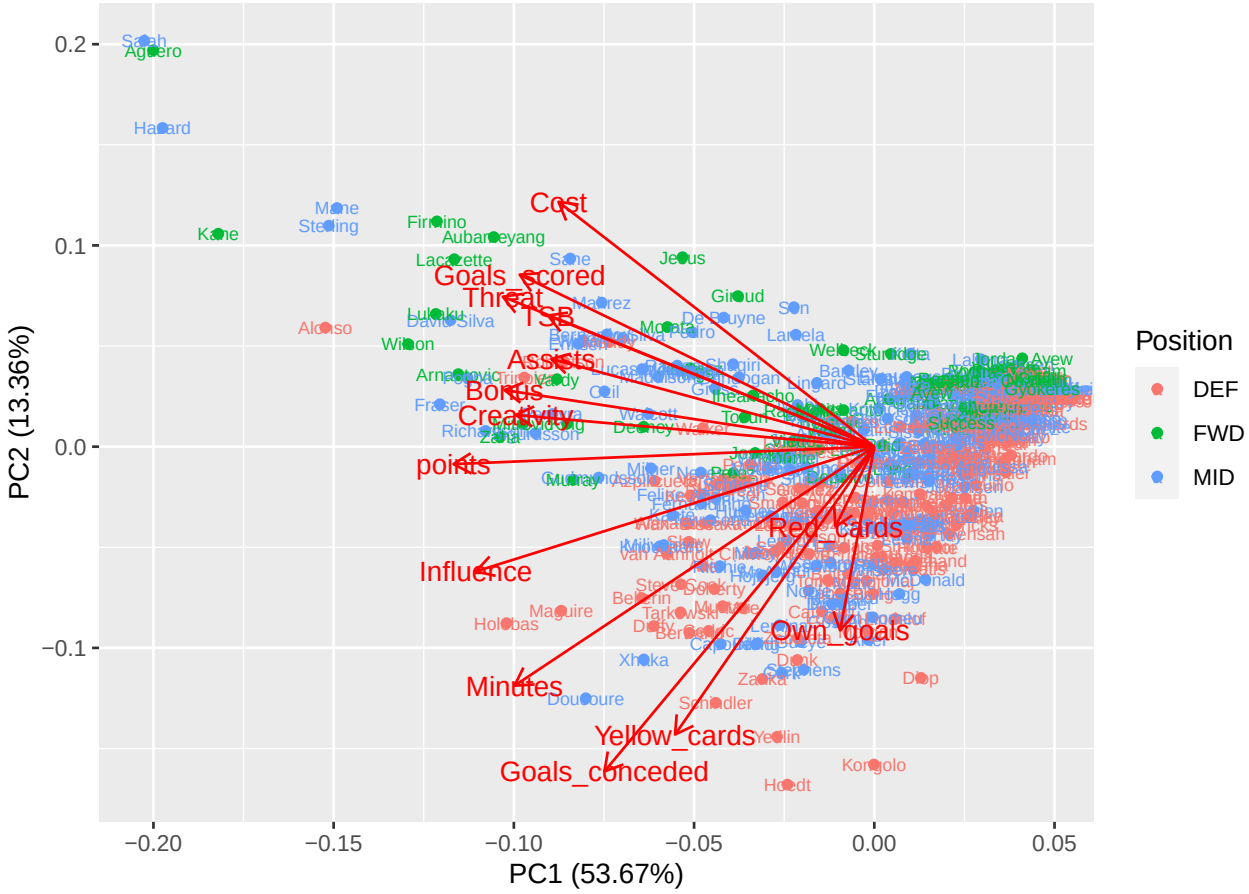
```r
# Contributions of variables to PC2
pca_p2 <- fviz_contrib(scaled_df_pca, choice = "var", axes = 2,
    top = 10, ggtheme = theme_gray())
# Contributions of variables to PC3
pca_p3 <- fviz_contrib(scaled_df_pca, choice = "var", axes = 3,
    top = 10, ggtheme = theme_gray())
pca_p4 <- fviz_pca_var(X = scaled_df_pca, col.var = "contrib",
    repel = TRUE, gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
    ggtheme = theme_gray()) + ggtitle("Variables - PCA")
grid.arrange(pca_p1, pca_p2, pca_p3, pca_p4, nrow = 2)
```



İlk grafikten de görüleceği üzere birinci temel bileşen, verideki değişkenliği ençok açıklayan bileşendir. Birinci temel bileşende `cost` değişkeni dışında başta `points` ve `influence` olmak üzere bütün değişkenlerin katkısı varken ikinci temel bileşende başta `Goals_conceded` ve `yellow_cards`olmak üzere `cost` ve `minutes` değişkeninin katkısı vardır. Üçüncü temel bileşene geldiğimizde ise `red_cards` ve `own_goals` değişkeni katkıda bulunmaktadır. Tüm bu değişkenleri dördüncü grafikte topluca incelediğimizde hepsinin aynı yöne bakması nedeniyle değişkenlerin cosine benzerliği çok yüksek olmasına rağmen `cost` ile `red cards` ve `cost` ile `own goals`'ın birbiri ile olan açısı 90° olduğu için birbirleriyle ilişkisizdir. `TSB` ve `Threat` değişkeninin üst üste olması bir oyuncunun seçilme yüzdesi arttıkça karşı kaleyi tehtit etmesinin de arttığının işaretçisidir. Sonuç olarak temel bileşenlerden elde ettiğimiz sonuç, en başta incelediğimiz korelasyon matrisi ile doğrudan alakalıdır. Elde ettiğimiz bu üç bileşeni yeni bir değişken olarak tanımlayacak olursak:

- ilk bileşen, forvet oyuncuları ve ofansif orta sağa oyuncularının saha içi etkinliğini açıklayan bileşendir.
- ikinci bileşen, orta sağa ve defansif orta sağa oyuncularının saha içi etkinliğini açıklayan bileşendir.
- üçüncü bileşen ise defans oyuncularının saha içi etkinliğini açıklayan değişkendir.

```
autoplot(scaled_df_pca, data = scaled_data, colour = "Position",
    loadings = TRUE, label = TRUE, label.size = 2.5, loadings.label = TRUE,
    loadings.label.size = 4)
```

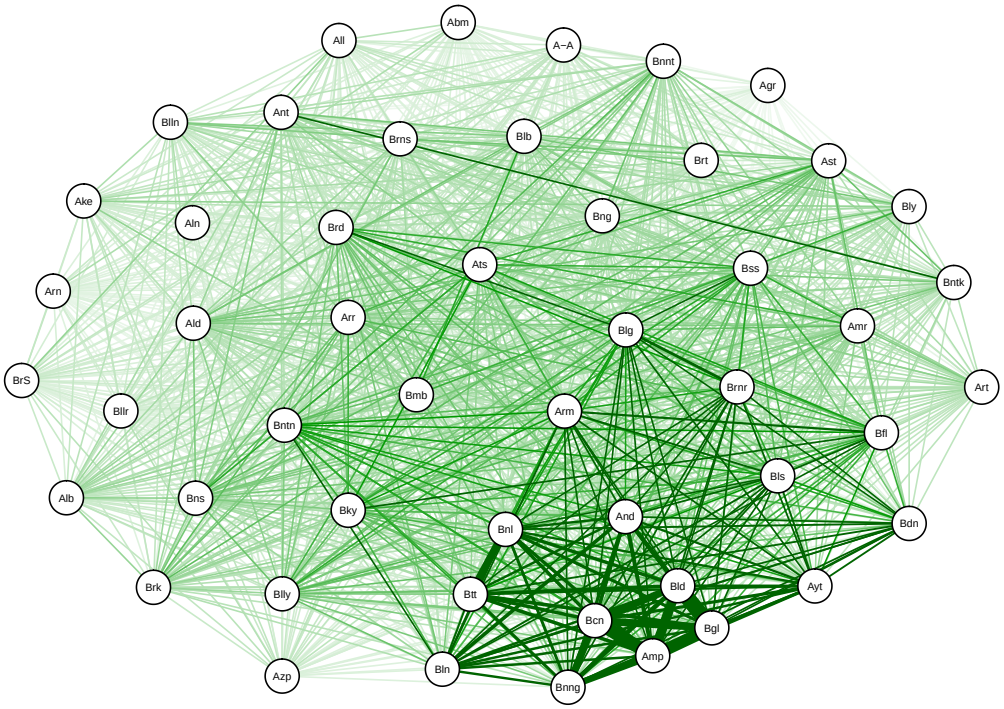


Grafiği incelediğimizde oyuncuların büyük çoğunluğunun orjin etrafında toplanması ilk soruda elde ettiğimiz yoğunluk grafiği ile örtüşmektedir. Oyunculara ait değişkenlerin sağdan çarpık olması, ve orta saha ve defans oyuncularının ağırlıklı olması nedeniyle oyuncuların çoğunluğunun orjin etrafında toplanmasına neden olmuştur. Ayrıca oyuncuların oynadığı mevkiye göre görselliştirilen bu grafiğe göre oynalınan mevkinin oyuncuları kümelemede kullanılabileceği ve oyuncuların temel bileşenler etrafında kümelenmesi hakkında bize fikir vermektedir. Bunlara ek olarak bütün oyuncular orjin etrafında toplanmasına rağmen boxplot'dan da hatırlayacağımız aykırı değerler alan oyuncular özellikle ikinci temel bileşende toplanmışlardır. Grafiğin sol üst köşesine yığılan *Salah*, *Agriero*, *Hazard*, *Mane* gibi oyuncular forvet olması nedeniyle başta satın alma maliyetleri, sahadayken atılan gol sayısı ve oyunda geçirdikleri süre çok fazla olduğu için ikinci temel bileşenin pozitif bölgesinde yer almaktadır. Ayrıca bu oyuncular karşı takımın defansına yaptıkları baskı çok fazla olduğu için aldıkları sarı kart çok fazladır. Orta saha oyuncularının büyük çoğunluğunun hem birinci hemde ikinci temel bileşenin pozitif olduğu yerde yer almaları, bu oyuncuların hem forvet'e hemde defans'a destek vermelerinden kaynaklanmalıdır. Bu nedenle bu oyuncular sarı kart, saha içi etkinlik ve oyunda geçirilen süre değerleri yüksektir. Son olarak defans oyuncularının üçüncü bileşene katkı yapan kırmızı kart ve kendi kalesine gol atma değişkenlerinin etrafında yer alması, bu oyuncuların karşı takım oyuncularını durdurmak için baskı yapmaları ve kaleciden sonra kaleyi koruyan oyuncular olmasından kaynaklanmaktadır.

S3: Kümeleme analizinde hangi yöntemi seçtiğinizi ve kaç küme belirlediğinizi gerekçeleriniz ile açıklayınız. Elde ettiğiniz sonuçları yorumlayınız.

Kümeleme analizine geçmeden önce gözlem noktalarının birbirine olan uzaklığı hesaplanmalıdır. Veri kümemizdeki bütün değişkenlerde aykırı değer olduğu için bu aykırılıktan en az etkilenecek olan manhattan distance ölçüsü kullanılmalıdır. Ayrıca hemen belirtilmelidir ki eğer veriler scale edilmişse bu ölçütler arasında çok ufak farklılıklar olacaktır. Son olarak veri matrisinin büyüklüğü nedeniyle uzaklık büyük verilerde daha etkili olan uzaklık ağı ile gösterilmelidir.

```
dist_m <- as.matrix(dist(scaled_data[1:50, -c(1, 2)], method = "manhattan"))
dist_mi <- 1/dist_m  # one over, as qgraph takes similarity matrices as input
qgraph(dist_mi, layout = "spring", vsize = 3)
```



Veri kümesine ait en uygun algoritmayı bulmak için yukarıda da belirttiğimiz gibi veride aykırı değer var mı yok mu tartışmasının yanısıra literatürde bazı ölçütlerin hesaplanması yoluyla da veri kümesi için en uygun yöntem bulunmaktadır. Internal ve Stabilitiy ölçütleri bunlara örnektir. Internal ölçüsü connectivity, Silhouette Width, ve Dunn Index'ini dahil ederek bunları en optimum yapan algoritmayı seçerken Internal ölçütünün özel bir türü olan Stability ölçütü regression'dan hatırlayacağımız leave one out yöntemi gibi her bir değişkeni teker teker çıkararak en optimal sonucu veren algoritmayı seçmektedir. Bu iki ölçütü kullanarak veri setimiz için en uygun yöntemi bulmaya çalışalım.

```
# Choosing the right algorithm with internal measures
scaled_opt_algorithm_internal <- clValid(scaled_data[, -c(1,
    2)], nClust = 2:10, clMethods = c("hierarchical", "kmeans",
    "pam", "clara"), validation = "internal", method = "ward")
optimalScores(scaled_opt_algorithm_internal)
```

```
##                    Score       Method Clusters
## Connectivity 53.09126984 hierarchical        2
## Dunn          0.05690324       kmeans        7
## Silhouette    0.42650505       kmeans        2
```

```
# Choosing the right algorithm with stabilitiy measures
scaled_opt_algorithm_stability <- clValid(scaled_data[, -c(1,
    2)], nClust = 2:10, clMethods = c("hierarchical", "kmeans",
    "pam", "clara"), validation = "stability", method = "ward")
optimalScores(scaled_opt_algorithm_stability)
```

```
##          Score Method Clusters
## APN 0.02724868    pam        2
## AD  2.36233252 kmeans       10
## ADM 0.12533567    pam        2
## FOM 0.66278149 kmeans       10
```

Elde edilen sonuçlara göre Internal ölçütü hiyerarşik kümeleme için 2 küme önernirken kmeans için 2 ve 7 küme önermektedir. Stabilitiy ölçütü ise kmeans için 10 küme önerirken pam için 2 küme önermektedir. Sonuç olarak her iki ölçütün k-means'ı önermesi nedeniyle aday algoritmamız k-means'dir.

Veri kümemizin kümelemeye eğilimli olup olmadğını test etmek iiçin *hopkinks istatistiği* kullanılmıştır.

```
set.seed(123)
hopkins(data = scaled_data[, -c(1, 2)], nrow(scaled_data[, -c(1,
    2)]) - 1)
```

```
## $H
## [1] 0.1263217
```

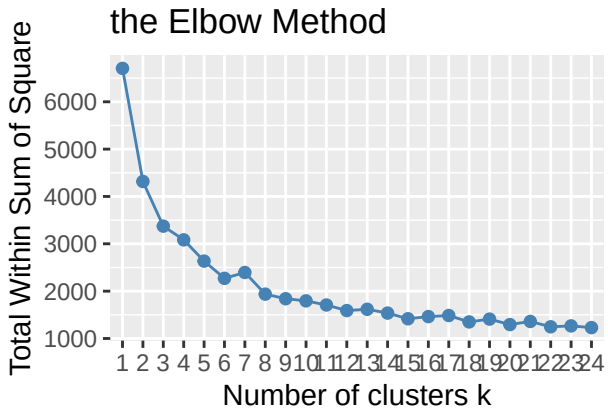
elde edilen p değeri $p > \alpha$ olduğu için verilerimiz kümeleme eğiliminde değil şeklinde yorum yapabiliriz. Bunun temel nedeni verilerimizin pozitif yönde çarpık olması ve veri setimizde orta saha ve defans oyuncularının ağırlıklı olmasından kaynaklanmaktadır.

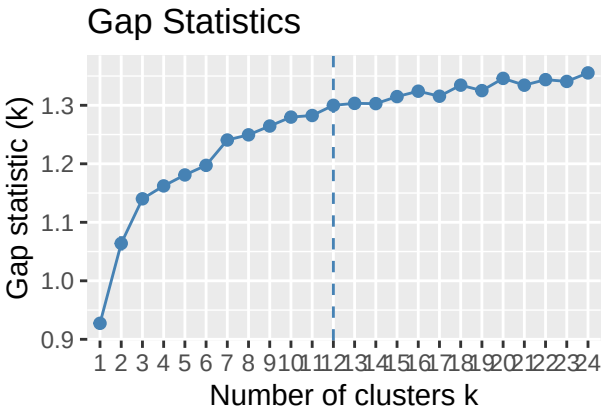
K-means Yöntemi

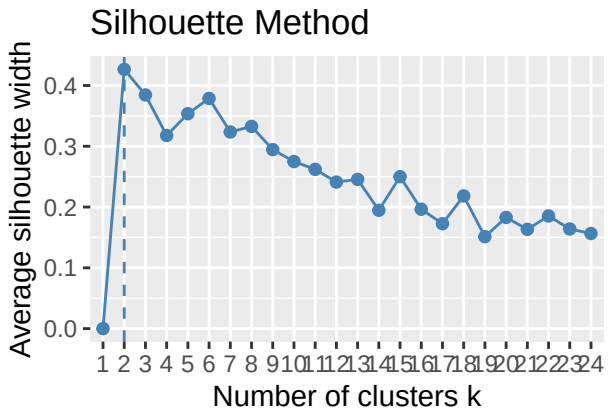
k-means kümeleme için en optimal k sayısının belirlenmesi için wss, gap istatistiği, Silhouette ve nbclust fonksiyonunda yer alan istatistikler kullanılmıştır.

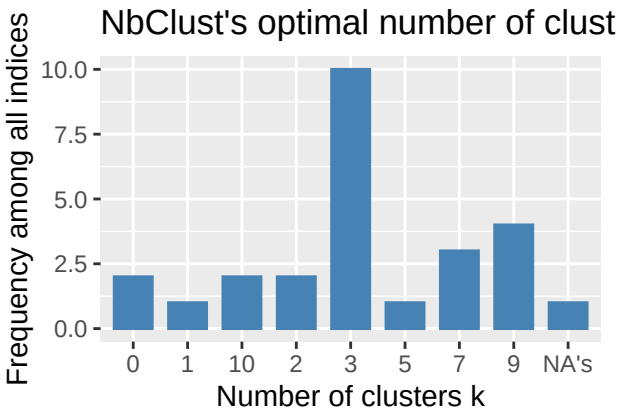
```
set.seed(31)
# function to compute total within-cluster sum of squares
km_elbow <- fviz_nbclust(scaled_data[, -c(1, 2)], kmeans, method = "wss",
    k.max = 24) + ggtitle("the Elbow Method") + theme_gray()
# Gap Statistics
km_gap <- fviz_nbclust(scaled_data[, -c(1, 2)], kmeans, method = "gap_stat",
    k.max = 24) + ggtitle("Gap Statistics") + theme_gray()
# The Silhouette Method
km_silhouette <- fviz_nbclust(scaled_data[, -c(1, 2)], kmeans,
    method = "silhouette", k.max = 24) + ggtitle("Silhouette Method") +
    theme_gray()
# NbCluster method
scaled_nbclust <- NbClust(scaled_data[, -c(1, 2)], distance = "manhattan",
    min.nc = 2, max.nc = 10, method = "ward.D2", index = "all")
```

```
km_nbclust <- fviz_nbclust(scaled_nbclust) + theme_gray() + ggtitle("NbClust's optimal number of cluste
grid.arrange(km_elbow, km_gap, km_silhouette, km_nbclust, nrow = 2)
```









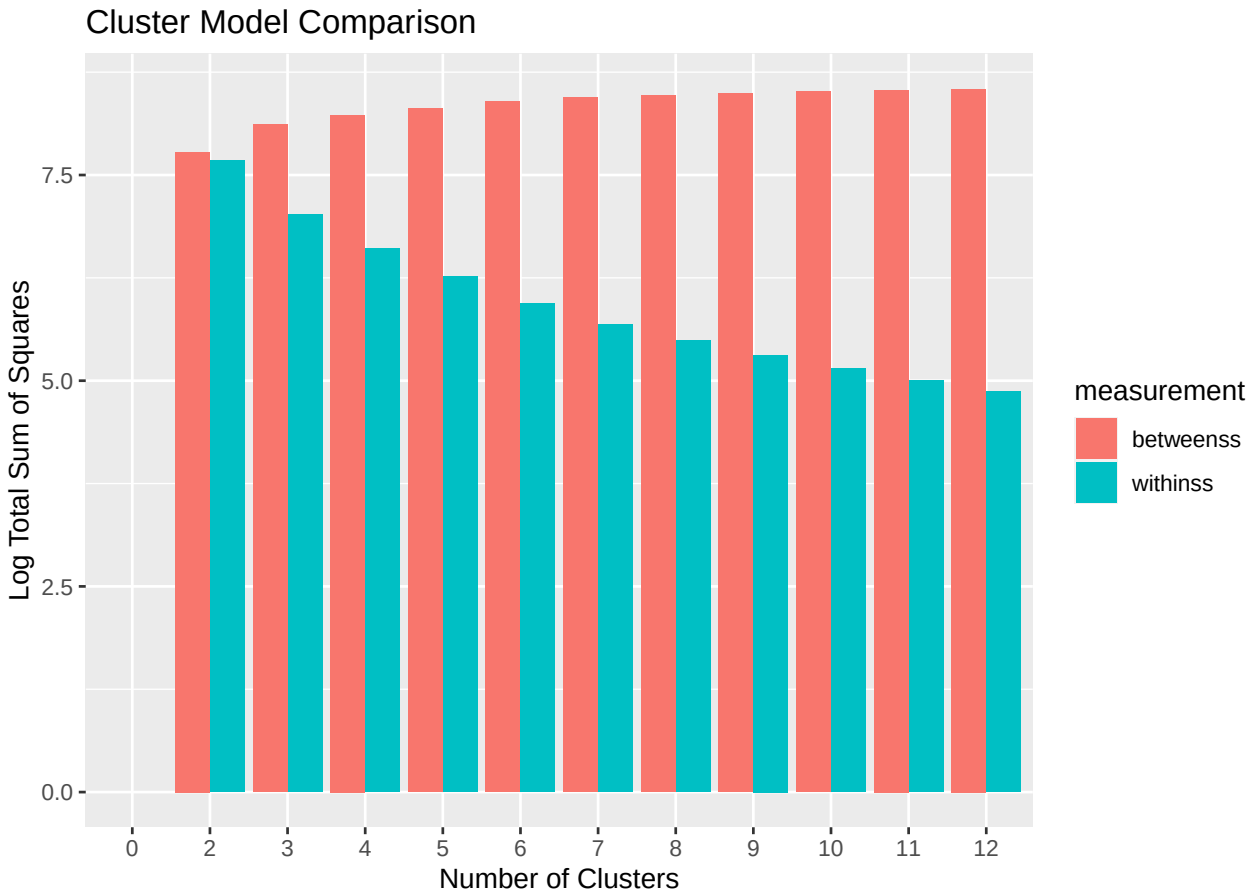
wss, gap istatistiği, Silhouette ve nbclust fonksiyonunda yer alan istatistiklerden elde ettiğimiz sonuçlara göre wss ye göre dirsek k = 3 de sağlanmıştır. Gap istatistiği 12 küme önerirken nbclust fonksiyonunda yer alan istatistikler çoğunlukla üç küme önermiştir. Oyuncuların oynadığı mevkileri göz önüne aldığımızda üç küme en optimum küme olarak gözükmektedir.

```
k2 <- kmeans(scaled_data[, -c(1, 2)], centers = 2, nstart = 25)
k3 <- kmeans(scaled_data[, -c(1, 2)], centers = 3, nstart = 25)
k4 <- kmeans(scaled_data[, -c(1, 2)], centers = 4, nstart = 25)
k5 <- kmeans(scaled_data[, -c(1, 2)], centers = 5, nstart = 25)
k6 <- kmeans(scaled_data[, -c(1, 2)], centers = 6, nstart = 25)
k7 <- kmeans(scaled_data[, -c(1, 2)], centers = 7, nstart = 25)
k8 <- kmeans(scaled_data[, -c(1, 2)], centers = 8, nstart = 25)
k9 <- kmeans(scaled_data[, -c(1, 2)], centers = 9, nstart = 25)
k10 <- kmeans(scaled_data[, -c(1, 2)], centers = 10, nstart = 25)
k11 <- kmeans(scaled_data[, -c(1, 2)], centers = 11, nstart = 25)
k12 <- kmeans(scaled_data[, -c(1, 2)], centers = 12, nstart = 25)
```

Hesaplanan her istatistiğin önerdiği küme sayısı dikkate alınarak kümeleme oluşturmuş ve bu kümelerin withhinss ve betweenss değerleri aşağıdaki gibi karşılaştırılmıştır.

```
ssc <- data.frame(kmeans = c(2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
    12), withinss = c(mean(k2$withinss), mean(k3$withinss), mean(k4$withinss),
    mean(k5$withinss), mean(k6$withinss), mean(k7$withinss),
    mean(k8$withinss), mean(k9$withinss), mean(k10$withinss),
    mean(k11$withinss), mean(k12$withinss)), betweenss = c(k2$betweenss,
    k3$betweenss, k4$betweenss, k5$betweenss, k6$betweenss, k7$betweenss,
    k8$betweenss, k9$betweenss, k10$betweenss, k11$betweenss,
    k12$betweenss))
```

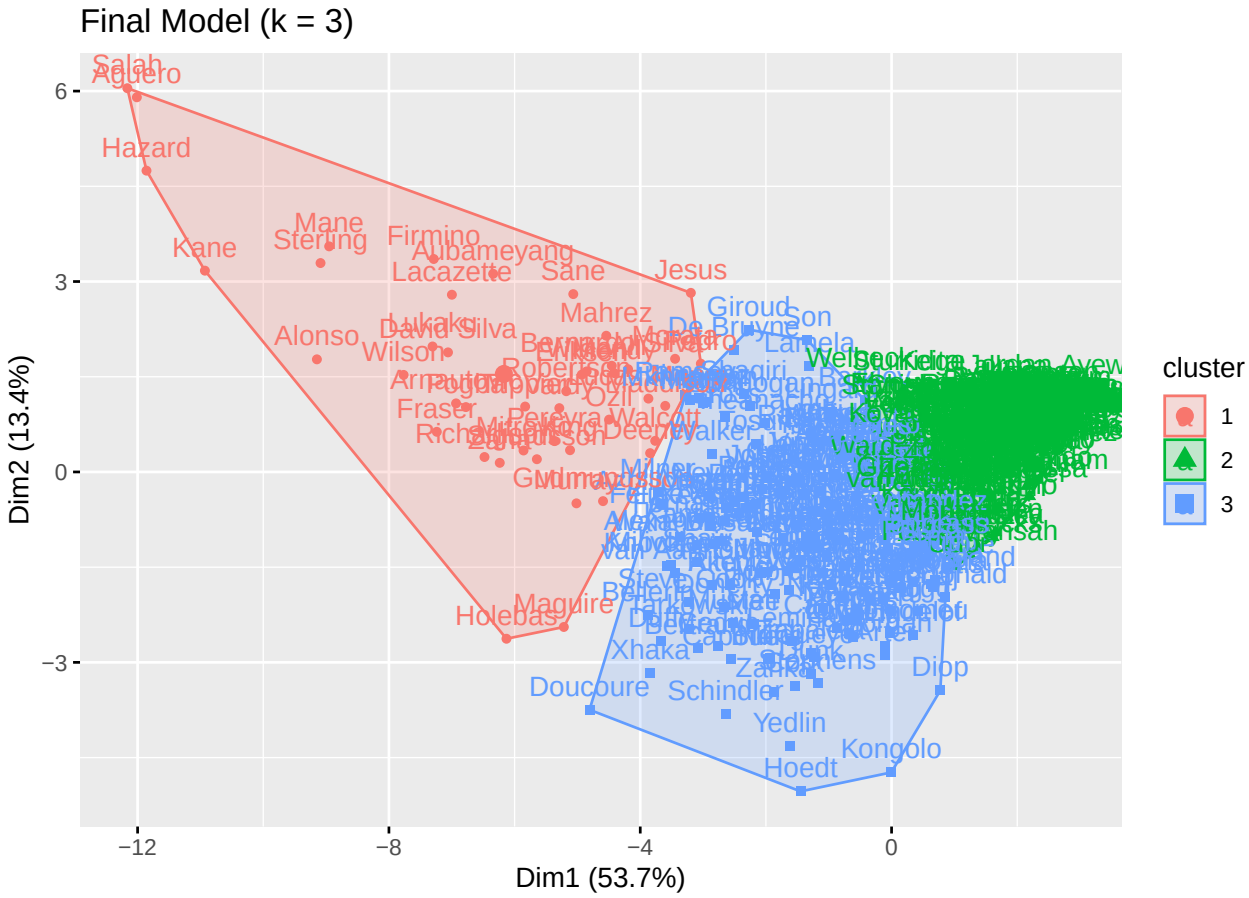
```
ssc %<>% gather(., key = "measurement", value = value, -kmeans)
ssc %>% ggplot(., aes(x = kmeans, y = log(value), fill = measurement)) +
    geom_bar(stat = "identity", position = "dodge") + ggtitle("Cluster Model Comparison") +
    xlab("Number of Clusters") + ylab("Log Total Sum of Squares") +
    scale_x_discrete(name = "Number of Clusters", limits = c("0",
        "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"))
```



Logaritmik artışın hesaplandığı withhinss ve betweenss değerlerine göre küme sayısı arttıkça withhinss değeri önemli bir oranda azalmaktadır ancak bu durum betweenss için geçerli değildir. Genel olarak oyuncuları ufak kümelere bölmek homejenliği arttıracağı için withhinss değeri düşmektedir ancak kümeler arası uzaklık olan betweenss değeri üçüncü kümeden itibaren ciddi bir artış göstermemektedir. Elde ettiğimiz bu sonuçtan da haraketle, veri setimiz için en optimal küme sayısının üç olacağı görülmektedir.

S4: Finalde belirlediğiniz kümeler için ayrıntılı yorumlar yapınız.

```
k3 <- kmeans(scaled_data[, -c(1, 2)], centers = 3, nstart = 25)
fviz_cluster(k3, data = scaled_data[, -c(1, 2)]) + ggtitle("Final Model (k = 3)")
```



Final model olarak belirlediğimiz $k = 3$ kümesine ilk bakışta yukarıda elde ettiğimiz temel bileşenler analizi sonucuyla örtüştüğü görülmektedir. Kırmızı ile taranan oyuncular forvet oyuncularıdır ve bu kümenin seyrek olması, veri kümemizdeki forvet oyuncularının diğerlerine oranda daha az olmalarıyla (veri setinde 79 adet forvet oyuncusu vardır.) örtüşmektedir. Ayrıca birinci kümedeki *Salah*, *Agiero*, *Hazard* ve *Kane*'in aynı kümede bulunan oyunculardan bir hayli uzak olması, bu oyuncuların fiyatlarının ve attıkları gol sayısının aynı kümedeki diğer oyunculardan daha fazla olmasından dolayı kaynaklanmaktadır. Mavi kümeyi incelediğimizde ise burada da defans oyuncularınınn yer aldığı görülmektedir. Kırmızı kümeye göre daha yoğun olan bu kümede (veri setinde 187 adet defans oyuncusu vardır.) yer alan *Xhaka*, *Yedlin* ve *Kongolo*'nun diğer oyunculardan uzak olması yine bu oyuncuların gösterdiği performansların diğerlerinden fazla olmasından kaynaklanmaktadır. Yeşil kümeye geldiğimizde ise bu kümenin diğer iki kümeden daha yoğun olduğu görülmektedir (214 adet ortasaha oyuncusu vardır). Genel olarak veri setimizde yer alan orta saha oyuncularının performansları birbiri ile hemen hemen yakın olması nedeniyle burada ilk iki kümedeki gibi aykırı oyuncu bulunmamaktadır.

```
jaccard_df <- data.frame(actual = as.numeric(as.factor(scaled_data$Position)),
    predicted = k3$cluster)
jaccard <- function(df, margin) {
    if (margin == 1 | margin == 2) {
        M_00 <- apply(df, margin, sum) == 0
        M_11 <- apply(df, margin, sum) == 2
        if (margin == 1) {
            df <- df[!M_00, ]
            JSim <- sum(M_11)/nrow(df)
        } else {
            df <- df[, !M_00]
            JSim <- sum(M_11)/length(df)
        }
```

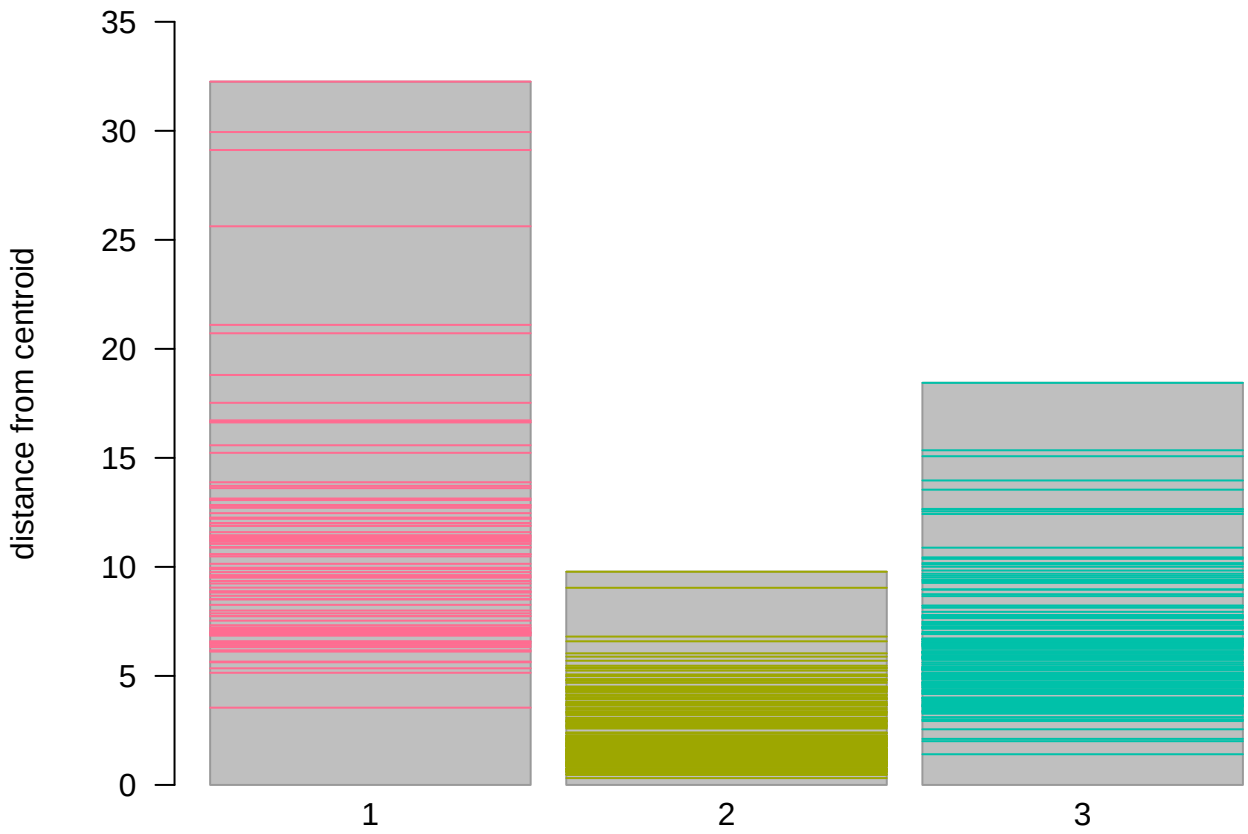
```
        JDist <- 1 - JSim
        return(c(JSim = JSim, JDist = JDist))
    } else break
}

jaccard(jaccard_df, margin = 1)
```

```
##    JSim  JDist
## 0.0125 0.9875
```

Elde ettiğimiz bu sonucun gerçek mevki değerleri ile ne derecede uyuşup uyuşmadığı jaccard ölçüsü ile test ettiğimizde uyuşmanın çok az olduğu görülmektedir. Bunun önemli nedenlerinden biri veri kümesindeki bazı oyuncuların mevkilerinin yanlış işeretlenmesinden kaynaklanmaktadır. Örneğin *Salah*, *Hazard*, *Mane* gibi oyuncular forvet olmasına rağmen orta saha oyuncusu olarak etiketlenmiştir. Ölçünün düşük çıkmasının diğer nedeni ise veri kümesinde aykırı gözlemler nedeniyle k-means algoritmasının iyi kümeleyememesinden kaynaklanmaktadır. Örneğin *Maguire*, *Holebas* gibi oyuncular defans olmasına rağmen bu oyuncuların performanslarının diğer defans oyuncularından daha iyi olması nedeniyle forvet olarak kümelenmiştir.

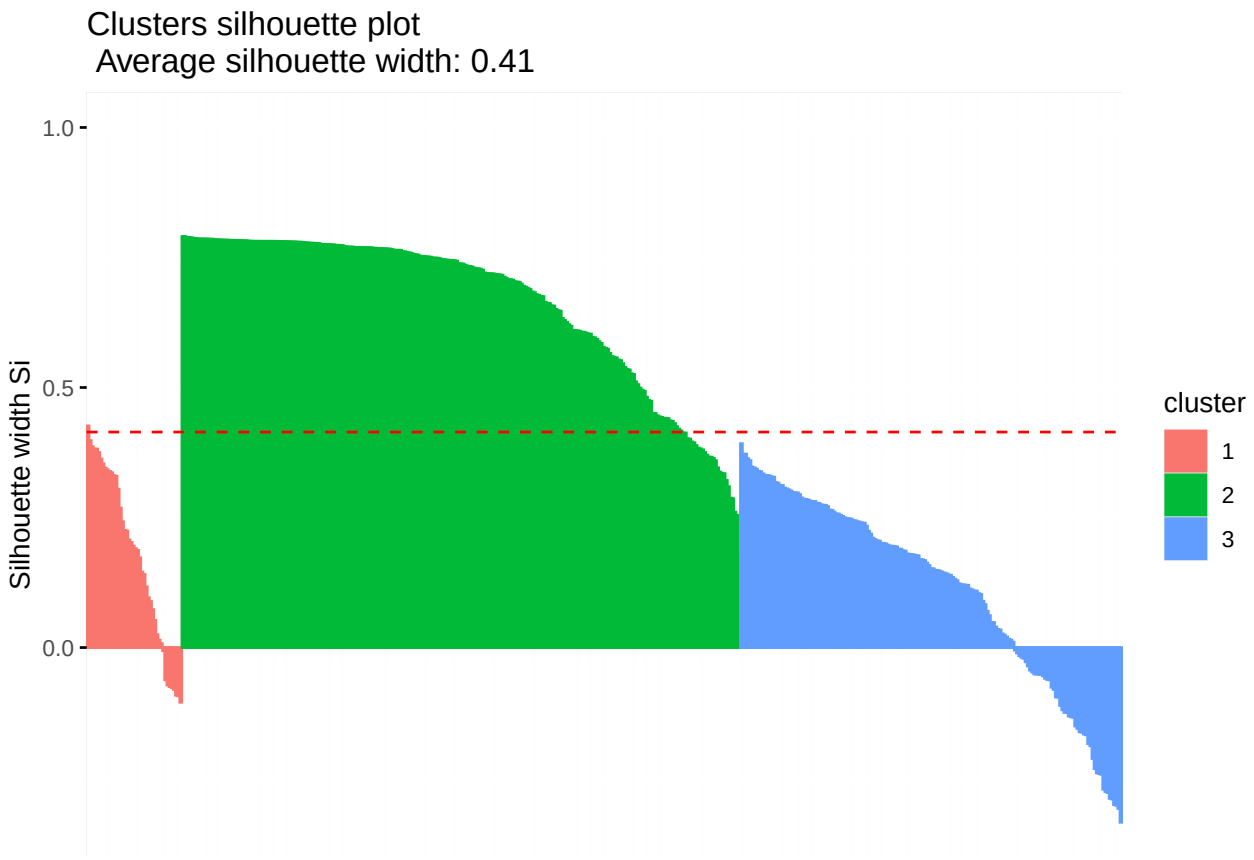
```
d1 <- cclust(scaled_data[, -c(1, 2)], 3, dist = "manhattan")
stripes(d1)
```



Küme içindeki elemanların hesaplanan centrioid'e uzaklığını incelediğimizde birinci (forvet) ve ikinci (defans) kümelerindeki aykırı oyuncuların k-means üzerinde yarattıkları etki net bir şekilde gözükmektedir. Oyuncuların performans olarak birbirine çok yakın olduğu orta saha mevkiindeki (üçüncü küme) değişim, beklendiği gibi diğerlerine oranla daha azdır.

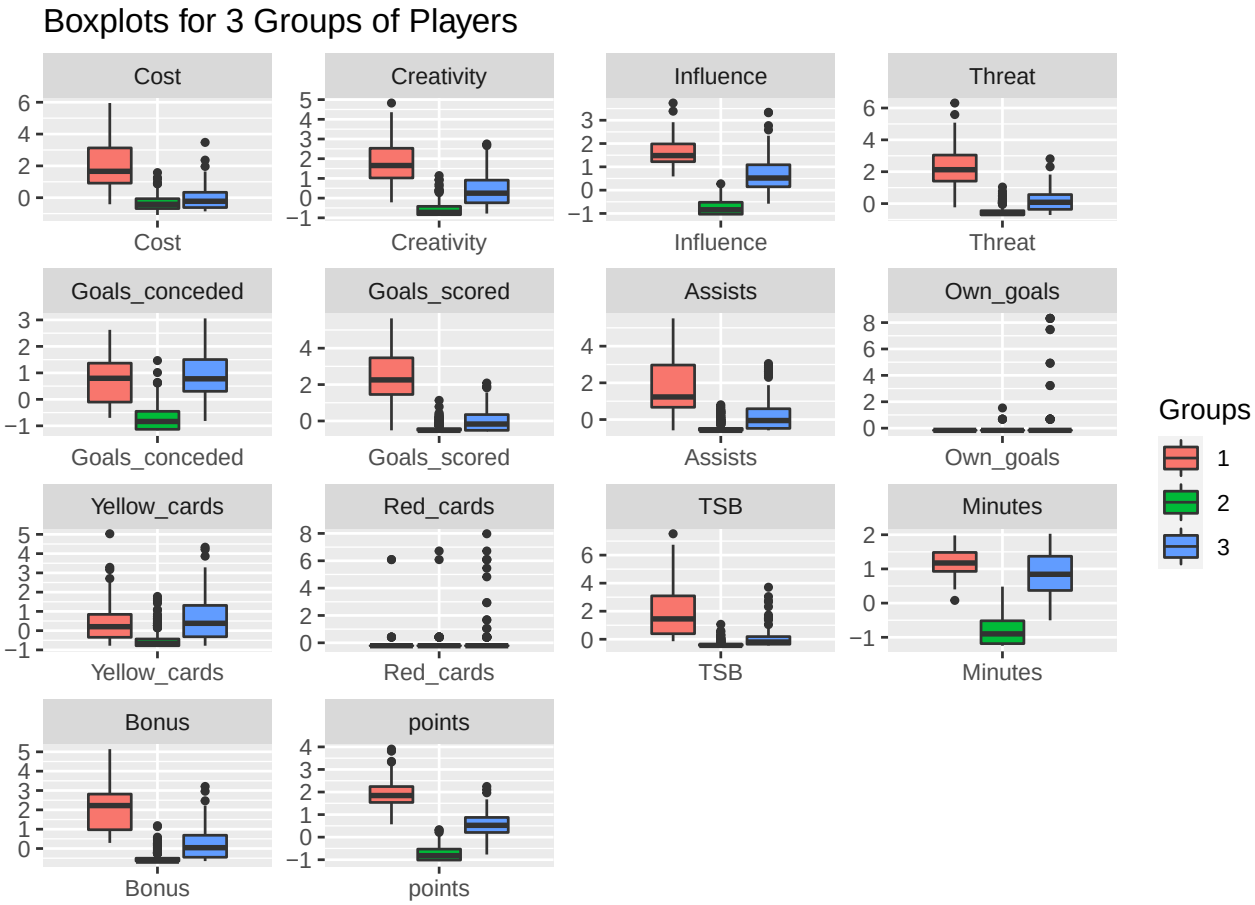
```
km.sil <- silhouette(k3$cluster, dist(scaled_data[, -c(1, 2)],
    method = "manhattan"))
```

```
fviz_silhouette(km.sil, print.summary = FALSE)
```



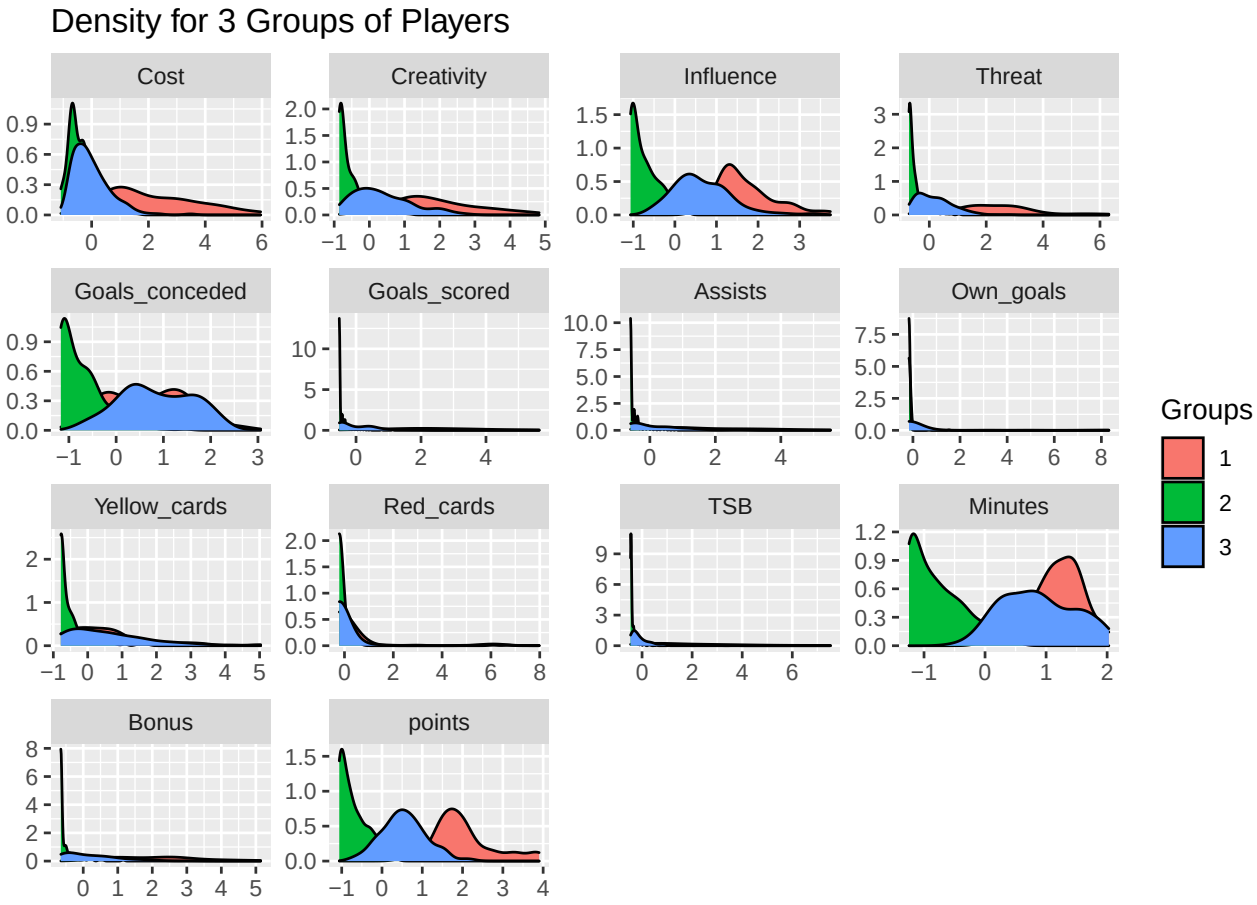
Kümeleme algoritmasının kümeleme performansını değerlendirmek için hesapladığımız bir diğer yöntem olan *silhoutte* istatistiğine göre ($0-1$ arasında değer alır ve 1'e yakın olması daha başarılı olduğu anlamına gelir.) hesapladığımız kümenin başarısı çok yüksek değildir.

```
scaled_data.c <- cbind(scaled_data[, -c(1, 2)], k3$cluster)
colnames(scaled_data.c)[15] <- c("Group")
df.m <- melt(scaled_data.c, id.var = "Group")
df.m$Group <- as.character(df.m$Group)
ggplot(data = df.m, aes(x = variable, y = value)) + geom_boxplot(aes(fill = Group),
    outlier.size = 1) + facet_wrap(~variable, scales = "free") +
    xlab(label = NULL) + ylab(label = NULL) + ggtitle("Boxplots for 3 Groups of Players") +
    guides(fill = guide_legend(title = "Groups"))
```



Elde ettiğimiz kümeleri değişkenler bazında inceleyecek olursak ilk göze çarpan, birinci kümenin yani forvet oyuncularının bulunduğu kümenin diğer kümelere göre daha değişken olduğu görülmektedir. Küme içindeki elemanların hesaplanan centrioid'e uzaklığını incelediğimiz grafikle örtüşen bu durumun nedeni, daha öncede açıkladığımız gibi *Salah*, *Agiero*, *Hazard* ve *Kane* gibi oyuncuların sergiledikleri performans bu kümenin değişkenliğini arttırmaktadır. Kırmızı kart, sarı kart ve kendi kalesine gol atma değişkenlerindeki aykırı değerlerin orta saha ve defans oyuncularının bulunduğu kümede yer alması, temel bileşenler analizinde elde ettiğimiz sonuçla örtüşmektedir. `TSB` değişkenindeki aykırı değerlerin orta saha ve defans oyuncularının bulunduğu kümede yığılmasının nedeni ise bu mevkilerde yer alan *mesut özil*, *Yedlin* gibi oyuncuların kendi mevkilerinde performanslarının çok yüksek olmasından kaynaklanmaktadır.

```
ggplot(data = df.m, aes(value, fill = Group)) + geom_density() +
    facet_wrap(~variable, scales = "free") + xlab(label = NULL) +
    ylab(label = NULL) + ggtitle("Density for 3 Groups of Players") +
    guides(fill = guide_legend(title = "Groups"))
```



Elde ettiğimiz kümelerin değişken bazında yoğunluk grafiğine baktığımızda elde ettiğimiz grafik ile keşifsel veri analizi kısmında elde ettiğimiz yoğunluk grafiğiyle birebir örtüştüğü gözükmektedir. İlk grafikte, değişkenlerin yoğunluk grafiğinde genellikle üç tepe bulunmaktaydı ve bu tepelerin aslında oyuncuların mevkileriyle ilişkili olduğu açıkça ortaya çıkmaktadır. Ayrıca değişkenlerin pozitif yönde çarpık olmasının nedeninin defans oyuncularından kaynaklandığı net bir şekilde görülmektedir.