STATYSTYCZNA ANALIZA DANYCH INFORMATYKA I EKONOMETRIA PROJEKT 1

BARTŁOMIEJ ORTYL

W projekcie będę badał dane dotyczące statystyk wybranych zawodników najlepszej koszykarskiej ligi świata, czyli NBA. Statystyki te pochodzą z sezonu zasadniczego 2018/2019.

Do analizy tych danych w projekcie posłużą takie metody jak analiza skupień, skalowanie wielowymiarowe czy porządkowanie liniowe.

Na samym początku przeprowadzę opis oraz wstępną analizę wykorzystywanych danych za pomocą podstawowych statystyk, wykresów czy korelacji.

OPIS I WSTĘPNA ANALIZA DANYCH

Opis danych

	dane.MIN	dane.PTS	dane.FG.	dane.X3P.	dane.FT.
Tyler Johnson	1529	19.4	41.3	34.6	74.8
Mikal Bridges	2417	13.6	43.0	33.5	80.5
Karl-Anthony Towns	2545	35.5	51.8	40.0	83.6
Cory Joseph	2063	12.5	41.2	32.2	69.8
Zach Collins	1356	18.1	47.3	33.1	74.6
Maurice Harkless	1415	15.6	48.7	27.5	67.1
Lou Williams	1993	36.1	42.5	36.1	87.6
DeAndre Jordan	2047	17.8	64.1	0.0	70.5
Jamal Murray	2447	26.8	43.7	36.7	84.8
Justin Holiday	2607	15.8	38.6	34.8	89.6
D'Angelo Russell	2448	33.6	43.4	36.9	78.0
Jared Dudley	1220	11.3	42.3	35.1	69.6
Zach LaVine	2171	33.0	46.7	37.4	83.2
Lauri Markkanen	1682	27.8	43.0	36.1	87.2
Justin Jackson	1613	17.3	44.7	35.5	78.5
Trae Young	2503	29.7	41.8	32.4	82.9
Ante Zizic	1082	20.4	55.3	0.0	70.5
Damyean Dotson	2004	18.7	41.5	36.8	74.5
Bojan Bogdanovic	2573	27.1	49.7	42.5	80.7
Dwyane Wade	1885	27.6	43.3	33.0	70.8
Trey Lyles	1120	23.4	41.8	25.5	69.8
Austin Rivers	2028	14.6	40.6	31.8	52.6
Emmanuel Mudiay	1607	26.1	44.6	32.9	77.4
Enes Kanter	1639	26.8	54.9	29.4	78.7
CJ McCollum	2375	29.7	45.9	37.5	82.8
Giannis Antetokounmpo	2358	40.6	57.8	25.6	72.9
Josh Okogie	1757	15.6	38.6	27.9	72.8
Dewayne Dedmon	1609	20.7	49.2	38.2	81.4
Jonathon Simmons	1064	16.5	38.0	26.9	74.2
Jakob Poeltl	1273	16.0	64.5	0.0	53.3

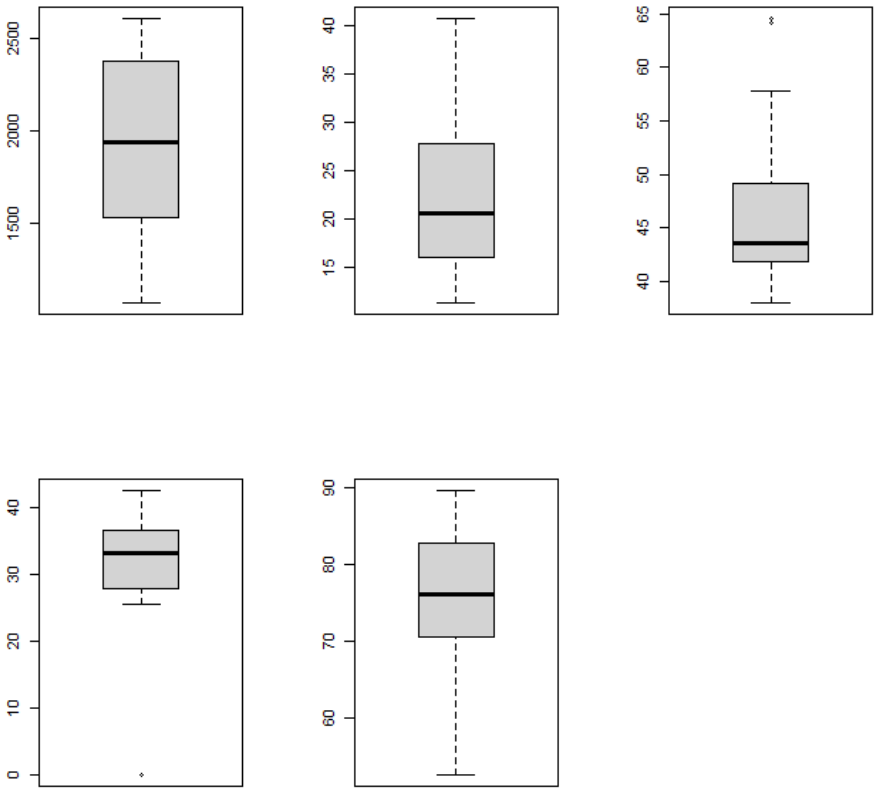
Dane, które wykorzystam w projekcie wybrałem z dwóch dostępnych na platformie UPEL. Zdecydowałem się na wybór danych dotyczących statystyk zawodników NBA. Pierwotne dane zmodyfikowałem do zmiennych widocznych na załączonym wyżej obrazku, powodem mojej decyzji jest to, że statystyki, które usunąłem z danych są według mnie najmniej istotne przy wyborze najlepszego zawodnika i najlepszej analizy tych zawodników. Usunięte kolumny to wiek zawodnika, liczba zwycięstw i liczba porażek w sezonie zasadniczym. Według mnie, wiek zawodnika nie definiuje jego umiejętności i tego jak dobrym jest graczem, dlatego zdecydowałem się zrezygnować z tej kolumny. Liczba zwycięstw i liczba porażek to statystki drużyny, sam zawodnik nie jest w stanie wygrać meczu, oczywiście można by tutaj polemizować na ten temat, jednakże na potrzeby projektu zdecydowałem, że nie ma to istotnego wpływu na ogólną prezentacje zawodnika i jego statystyk indywidualnych.

Podstawowe statystyki

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
dane.MIN	1	30	1881.00	493.80	1939.00	1893.29	633.81	1064.0	2607.0	1543.0	-0.10	-1.35	90.15
dane.PTS	2	30	22.92	7.93	20.55	22.42	9.04	11.3	40.6	29.3	0.45	-0.98	1.45
dane.FG.	3	30	46.33	6.97	43.55	45.34	3.93	38.0	64.5	26.5	1.20	0.61	1.27
dane.X3P.	4	30	30.33	11.08	33.30	32.88	5.26	0.0	42.5	42.5	-1.92	2.73	2.02
dane.FT.	5	30	75.82	8.69	76.10	76.55	8.30	52.6	89.6	37.0	-0.88	0.81	1.59

Średnio zawodnik podczas sezonu zasadniczego spędza na parkiecie 1881 minut, co daje blisko 23 minuty na mecz przez 82 gry sezonu zasadniczego. Według tych statystyk, zawodnik średnio zdobywa blisko 23 punkty na mecz. Średnio zawodnik rzuca ze skutecznością 46 % z gry, 30 % za 3 punkty i 75% z linii rzutów wolnych. Dodatnia kurtoza dla zmiennych X3P, FG, FT wskazuje na koncentracje ich wartości wokół średniej.

Wykresy pudełkowe zmiennych



Z wykresów można odczytać, że wartości odstające są bardzo rzadkie i pojedyncze, dlatego nie usuwam żadnych wartości.

Współczynnik zmienności

```
"MIN"
0.262518
"PTS"
0.3460582
"FG."
0.1504572
"X3P."
0.3652111
"FT."
0.1146642
```

Współczynnik zmienności służy do badania stopnia zróżnicowania zmiennej. Wysoka wartość współczynnika zmienności świadczy o silnym zróżnicowaniu danych, niska wartość świadczy o małej zmienności cechy i jednorodności badanej populacji. Pożądana wartość współczynnika to powyżej 10%. Każda wartość spełnia założenie.

Macierz korelacji

Correlation matrix

	dane.MIN	dane.PTS	dane.FG.	dane.X3P.	dane.FT.
dane.MIN	1.00	0.45	-0.07	0.43	0.48
dane.PTS	0.45	1.00	0.18	0.27	0.48
dane.FG.	-0.07	0.18	1.00	-0.66	-0.29
dane.X3P.	0.43	0.27	-0.66	1.00	0.56
dane.FT.	0.48	0.48	-0.29	0.56	1.00

Macierz korelacji służy do przedstawiania korelacji pomiędzy zmiennymi. Korelacje te powinny być słabe pomiędzy zmiennymi objaśniającymi. Wartości bliskie 0 wskazują na słaby związek, a bliskie 1 na silny. Wartości powinny się zawierać w przedziale od (-0.9, 0.9). W tym przypadku wszystko się zgadza i mieści w przedziale. Zmienne nie są ze sobą skorelowane.

PORZĄDKOWANIE LINIOWE

W metodzie porządkowania liniowego będę porządkował obiekty od najgorszego do najlepszego według różnych kryteriów.

Metody porządkowania liniowego to metody wzorcowe i bezwzorcowe. Do przeprowadzenia porządkowania należy również określić rodzaj każdej zmiennej ze względu na jej charakter (stymulanty, destymulanty, nominanty). W moim przypadku wszystkie zmienne to stymulanty, ponieważ im większa wartość zmiennej tym lepiej, dlatego nie musze przemieniać żadnej zmiennej.

Hellwig

Pomijając przemianę zmiennych, która w moim przypadku jest nie potrzebna, przechodzę do standaryzacji zmiennych.

```
##Tworze pierwsza kolumne do nowej tabeli
odch_minuty<-sd(dane$MIN)
mutate(dane, odch = odch_minuty)
stand<-data.frame(mutate(dane, sredMIN = sr_minuty, odchMIN = odch_minuty, standMIN = (MIN - sredMIN)/odchMIN))
stand
## tworze nowa tabele ze zestandaryzowanymi danymi
standaryzacja<-data.frame(stand$standMIN, standPTS = ((dane$PTS - mean(dane$PTS))/sd(dane$PTS)),
                          standFG = ((dane$FG - mean(dane$FG))/sd(dane$FG)),
                          standX3P. = ((dane$X3P. - mean(dane$X3P.))/sd(dane$X3P.)),
                          standFT. = ((dane$FT. - mean(dane$FT.))/sd(dane$FT.)))
```

	stand.standMIN	standPTS	standFG	standX3P.	standFT.
Tyler Johnson	-0.712844486	-0.44414708	-0.72116746	0.38548858	-0.1177027
Mikal Bridges	1.085467741	-1.17528797	-0.47727130	0.28618239	0.5379053
Karl-Anthony Towns	1.344683918	1.58539918	0.78524998	0.87299170	0.8944641
Cory Joseph	0.368573002	-1.31395262	-0.73551430	0.16882052	-0.6927975
Zach Collins	-1.063191351	-0.60802348	0.13964251	0.25007104	-0.1407065
Maurice Harkless	-0.943708894	-0.92317042	0.34049817	-0.25548775	-1.0033487
Lou Williams	0.226814155	1.66103444	-0.54900547	0.52090611	1.3545400
DeAndre Jordan	0.336170979	-0.64584112	2.54991042	-2.73814251	-0.6122843
Jamal Murray	1.146221532	0.48868785	-0.37684348	0.57507312	1.0324869
Justin Holiday	1.470241753	-0.89795866	-1.10853195	0.40354425	1.5845779
D'Angelo Russell	1.148246659	1.34588751	-0.41988397	0.59312879	0.2503579
Jared Dudley	-1.338608539	-1.46522314	-0.57769913	0.43062775	-0.7158013
Zach LaVine	0.587286651	1.27025224	0.05356151	0.63826797	0.8484565
Lauri Markkanen	-0.403000150	0.61474662	-0.47727130	0.52090611	1.3085324
Justin Jackson	-0.542733870	-0.70887050	-0.23337515	0.46673910	0.3078674
Trae Young	1.259628610	0.85425829	-0.64943330	0.18687620	0.8139509
Ante Zizic	-1.618075979	-0.31808831	1.28738913	-2.73814251	-0.6122843
Damyean Dotson	0.249090545	-0.53238822	-0.69247380	0.58410096	-0.1522084
Bojan Bogdanovic	1.401387456	0.52650548	0.48396649	1.09868758	0.5609091
Dwyane Wade	0.008100506	0.58953486	-0.43423081	0.24104321	-0.5777786
Trey Lyles	-1.541121177	0.06008802	-0.64943330	-0.43604446	-0.6927975
Austin Rivers	0.297693578	-1.04922919	-0.82159529	0.13270918	-2.6711237
Emmanuel Mudiay	-0.554884629	0.40044670	-0.24772198	0.23201537	0.1813466
Enes Kanter	-0.490080584	0.48868785	1.23000180	-0.08395887	0.3308712
CJ McCollum	1.000412433	0.85425829	-0.06121315	0.64729581	0.8024490
Giannis Antetokounmpo	0.965985284	2.22829892	1.64605995	-0.42701662	-0.3362388
Josh Okogie	-0.251115671	-0.92317042	-1.10853195	-0.21937640	-0.3477407
Dewayne Dedmon	-0.550834376	-0.28027067	0.41223233	0.71049066	0.6414224
Jonathon Simmons	-1.654528254	-0.80971752	-1.19461295	-0.30965476	-0.1867141
Jakob Poeltl	-1.231276840	-0.87274691	2.60729775	-2.73814251	-2.5906104

Następnie tworze wzorzec, odległość od wzorca i odległość możliwie daleką.

```
wzorzec<-c(max(standaryzacja$stand.standMIN), max(standaryzacja$standPTS), max(standaryzacja$standFG),
           max(standaryzacja$standX3P), max(standaryzacja$standFT))
wzorzec
## ODleglosc od wzorca

odlegloscodwzorca<-data.frame((standaryzacja-wzorzec)^2)
odlegloscodwzorca2<-mutate(odlegloscodwzorca, odleglosc=(stand.standMIN+standPTS+standFG+standX3P.+standFT.)^(1/2))
## Odleglosx "mozliwie daleka"
srednia_odl<-mean(odlegloscodwzorca2$odleglosc)
srednia_odl
odchylenie_odl<-sd(odlegloscodwzorca2$odleglosc)
odchylenie_odl
d_zero<-srednia_odl+2*odchylenie_odl
```

Pozostaje skorzystać ze wzoru: HELLWIG= 1 – di0/d0 i w ten sposób uzyskujemy ranking najlepszych graczy na podstawie wybranych statystyk.

```
##Finalna wersja rankingu graczy
Hellwig_<-data.frame(1-(odlegloscodwzorca2$odleglosc/d_zero), row.names = names)
Hellwig_
```

	PLAYER	Hellwig
1	Ante Zizic	0.07607624
2	DeAndre Jordan	0.07788543
3	Austin Rivers	0.09935318
4	Jakob Poeltl	0.10833738
5	Jared Dudley	0.15299684
6	Josh Okogie	0.21291786
7	Damyean Dotson	0.23115956
8	Emmanuel Mudiay	0.26706221
9	Dewayne Dedmon	0.31029238
10	Mikal Bridges	0.35398007
11	Trey Lyles	0.39086393
12	Maurice Harkless	0.41606776
13	Jonathon Simmons	0.44070750
14	Zach LaVine	0.45132878
15	Zach Collins	0.46129246
16	Tyler Johnson	0.48690119
17	Justin Jackson	0.50168181
18	Lou Williams	0.50559879
19	Justin Holiday	0.51989635
20	Dwyane Wade	0.53289601
21	Cory Joseph	0.53690090
22	Karl-Anthony Towns	0.56984145
23	Giannis Antetokounmpo	0.65346793
24	Trae Young	0.67021089
25	D'Angelo Russell	0.69585264
26	Lauri Markkanen	0.71155399
27	CJ McCollum	0.71912794
28	Enes Kanter	0.72392287
29	Jamal Murray	0.79022322
30	Bojan Bogdanovic	0.87004516

Po przeprowadzeniu wszystkich potrzebnych obliczeń ranking stworzony za pomocą metody Hellwiga prezentuje się następująco. Najlepszym zawodnikiem według wybranych przeze mnie statystyk został Bojan Bogdanovic, co jest dość zaskakujące.

ANALIZA SKUPIEŃ

Jest to metoda eksploracyjna oznacza to, że algorytm wykrywa pewne istniejące w danych struktury, ale ich nie wyjaśnia. Można ją nazwać również metodą uczenia maszynowego bez nadzoru, służy do grupowania obiektów w taki sposób, że grupuje je tak, by elementy w grupach były jak najbardziej podobne do siebie.

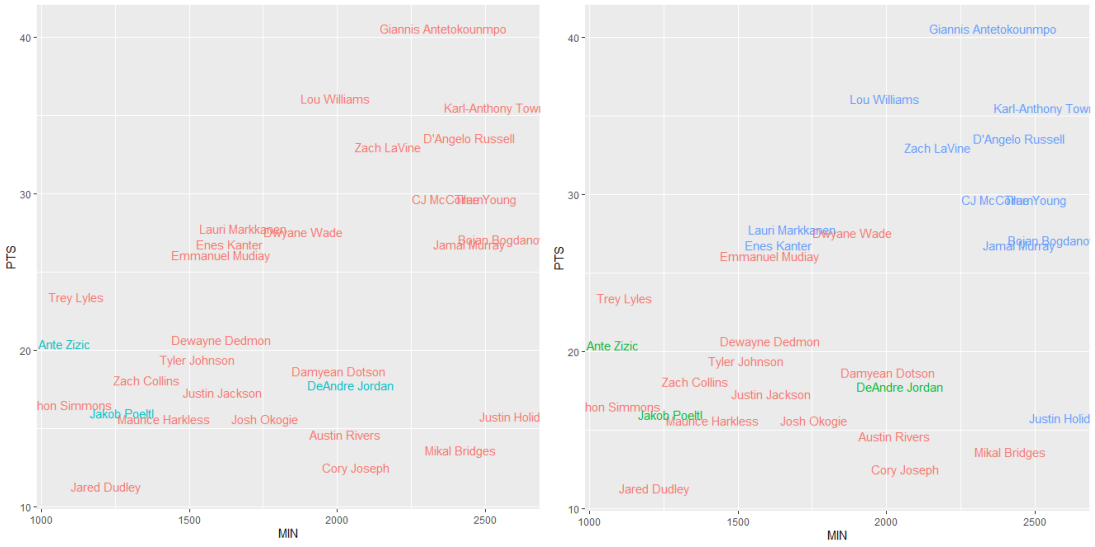
Istnieją dwa rodzaje klastrowania w analizie skupień, jest to grupowanie podziałowe(metoda k-średnich) oraz grupowanie hierarchiczne(metoda Warda).

Metoda k-średnich

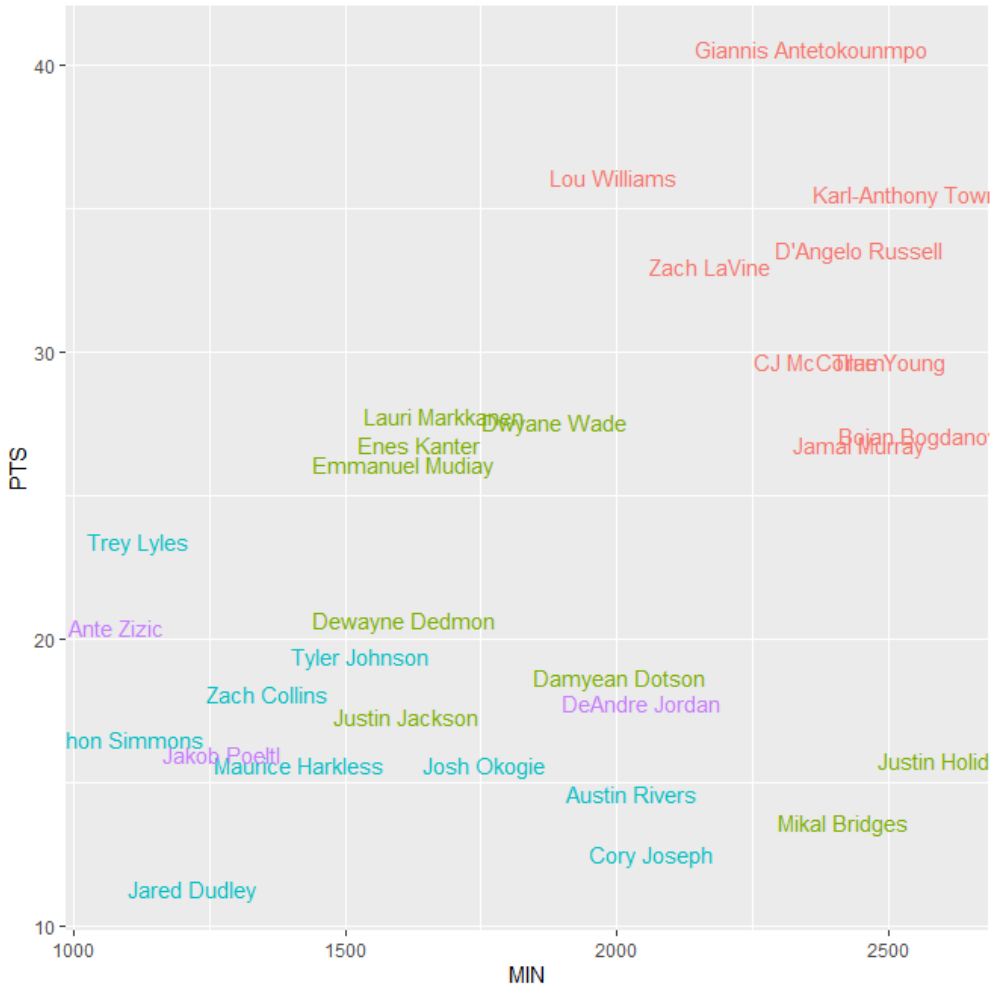
Metoda k-średnich jest metodą należącą do grupy algorytmów analizy skupień tj. analizy polegającej na szukaniu i wyodrębnianiu grup obiektów podobnych (skupień) . Reprezentuje ona grupę algorytmów niehierarchicznych. Główną różnicą pomiędzy niehierarchicznymi i hierarchicznymi algorytmami jest konieczność wcześniejszego podania ilości skupień.
Przy pomocy metody k-średnich zostanie utworzonych k różnych możliwie odmiennych skupień. Algorytm ten polega na przenoszeniu obiektów ze skupienia do skupienia tak długo aż zostaną zoptymalizowane zmienności wewnątrz skupień oraz pomiędzy skupieniami. Oczywistym jest, iż podobieństwo w skupieniu powinno być jak największe, zaś osobne skupienia powinny się maksymalnie od siebie różnić.

```
# 2 grupy
library(stats)
grup2<-kmeans(x=dane_st,centers=2,nstart=20)
sort(grup2$cluster)
dane1$grup2<-as.factor(grup2$cluster)
describeBy(dane1[,-7],group=dane1$grup2)

# 3 grupy
grup3<-kmeans(x=dane_st,centers=3,nstart=20)
sort(grup3$cluster)
dane1$grup3<-as.factor(grup3$cluster)
describeBy(dane1[,-c(7,8)],group=dane1$grup3)


# 4 grupy
grup4<-kmeans(x=dane_st,centers=4,nstart=20)
sort(grup4$cluster)
dane1$grup4<-as.factor(grup4$cluster)
describeBy(dane1[,-c(7,8)],group=dane1$grup4)
```

Podzieliłem zawodników na 4 grupy, aby wyniki były jak najbardziej optymalne.



Wykres zależności punktów zawodnika od liczby spędzonych minut na parkiecie przy podziale danych na 4 skupienia.



Po wykresie można wywnioskować, że im więcej zawodnik spędza minut na parkiecie tym zdobywa więcej punktów, co jest oczywiste, jednakże słaby zawodnik nie dostanie tyle minut na parkiecie co najlepszy zawodnik, co oczywiście tłumaczy, że w czerwonej grupie znajdują się praktycznie same gwiazdy NBA.

METODA WARDA

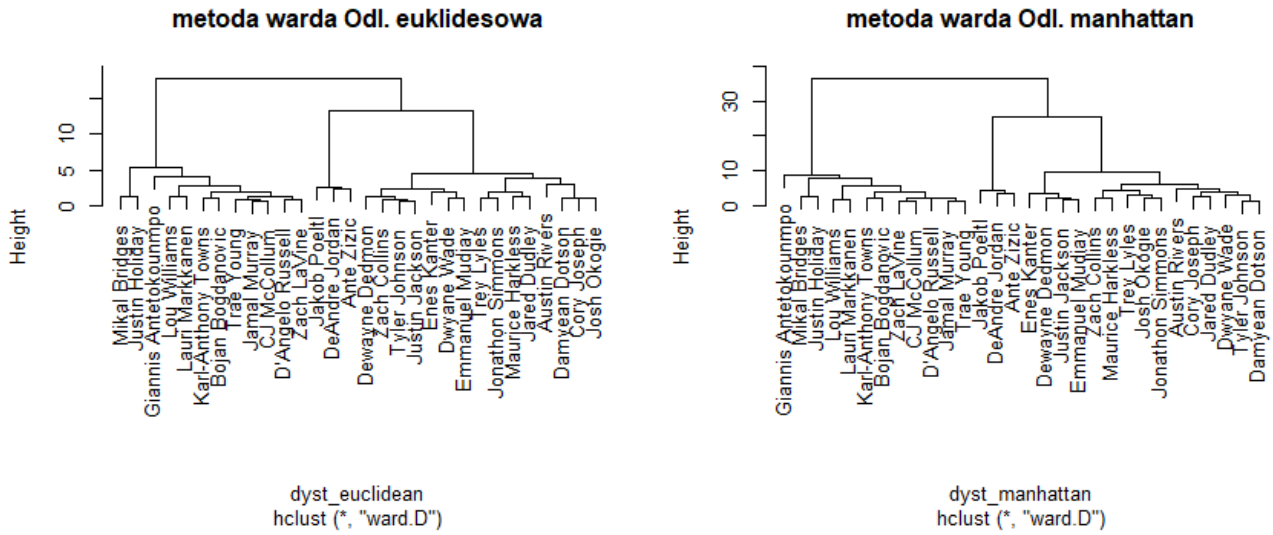
Metoda Warda to jedna z aglomeracyjnych metod grupowania, którą spośród pozostałych wyróżnia wykorzystanie podejścia analizy wariancji do oszacowania odległości między skupieniami. Zmierza ona do minimalizacji sumy kwadratów odchyleń dowolnych dwóch skupień, które mogą zostać uformowane na każdym etapie. Traktowana jest jako bardzo efektywna, chociaż zmierza do tworzenia skupień o małej wielkości. Daje pełną kontrolę nad wynikową liczbą grup oraz przedstawia najbardziej naturalne skupiska elementów.

Grupowanie polega na odległości między obiektami, gdy odległość jest mała to obiekty są do siebie podobne.

odległość euklidesowa: $d(x,y) = \sqrt{\sum(xi - yi)}\,2$

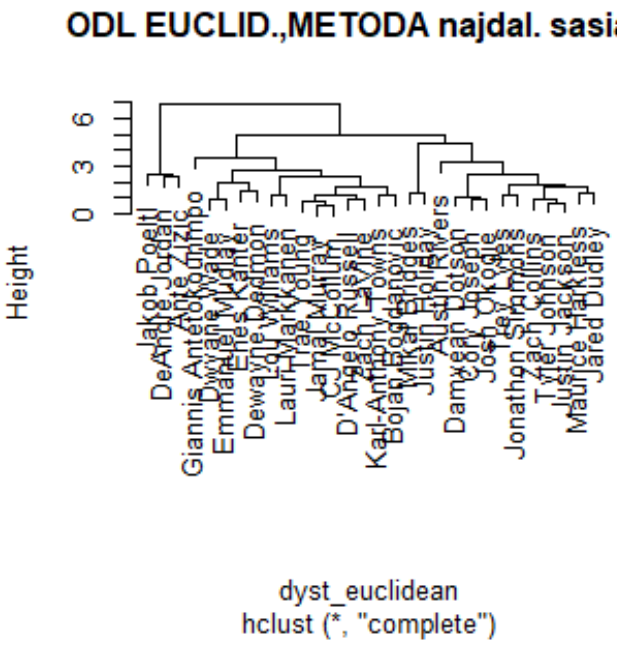
odległość miejska (nazywana również Manhattan): d(x,y)=∑|xi−yi $d(x,y) = \sum|xi - yi|$

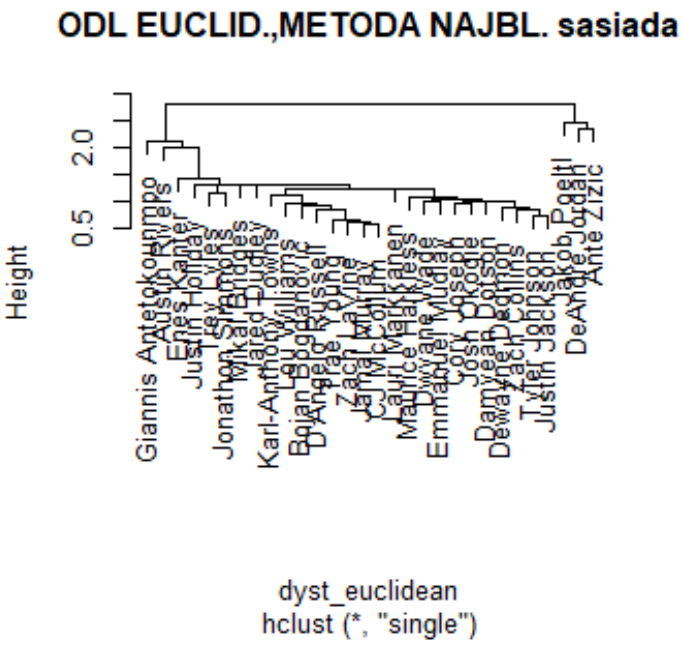
odległość Minkowskiego: $d(x,y) = \left(\sum|x_i - yi|^m \left(\frac{1}{n}\right)\right)$

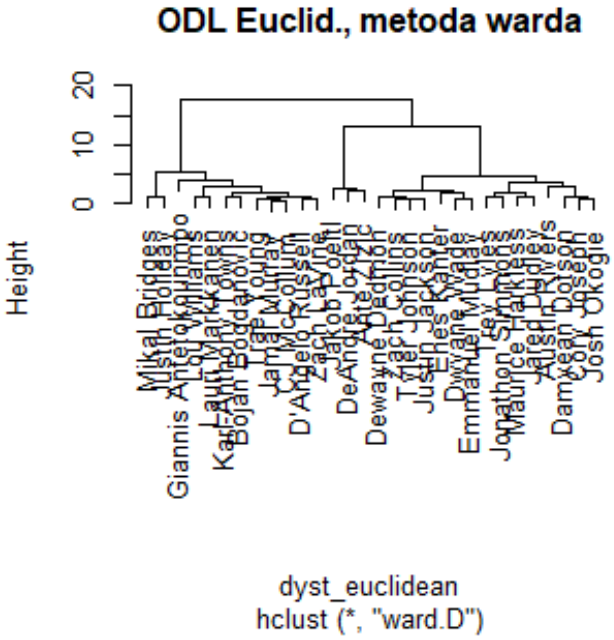


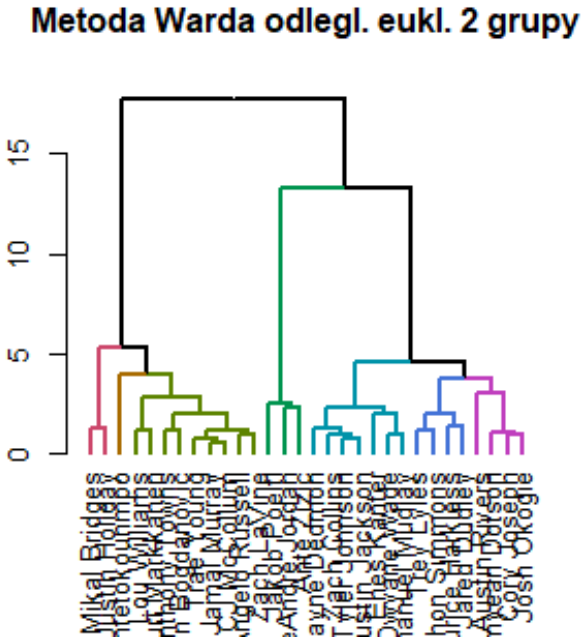
Zestawienie dendogramów w zależności od metody obliczania odległości.

Dendogramy nie różnią się znacząco ze względu na rodzaj odległości, co oznacza że grupowanie przebiega w podobny sposób.









Dendogramy w zależności od metody grupowania

Otrzymane dendogramy są od siebie różne, można zatem stwierdzić że wybór metody będzie mieć w wpływ na kształt grupy.

SKALOWANIE WIELOWYMIAROWE

Skalowanie wielowymiarowe jest eksploracyjną metodą SAD, która pozwala na wizualizację obiektów n-wymiarowych w przestrzeni m-wymiarowej (m<n). To oznacza, że skalowanie wielowymiarowe jest jedną z technik redukcji wielowymiarowości. Technicznie polega na znalezieniu funkcji, która przekształca odległości rzeczywiste na skalowane przy najmniejszej stracie informacji.

W skalowaniu wielowymiarowym mamy do czynienia z dwoma głównymi rodzajami, czyli skalowanie metryczne(Kruskala, Simmonsa) i skalowanie niemetryczne.

Skalowanie Kruskala

Rodzaj skalowania metrycznego. Polega na przekształceniu odległości z zastosowaniem funkcji zniekształcenia, którą chcemy minimalizować. W naszym przypadku będziemy wykorzystywać funkcje STRESS, wyznaczanej z następującego wzoru.

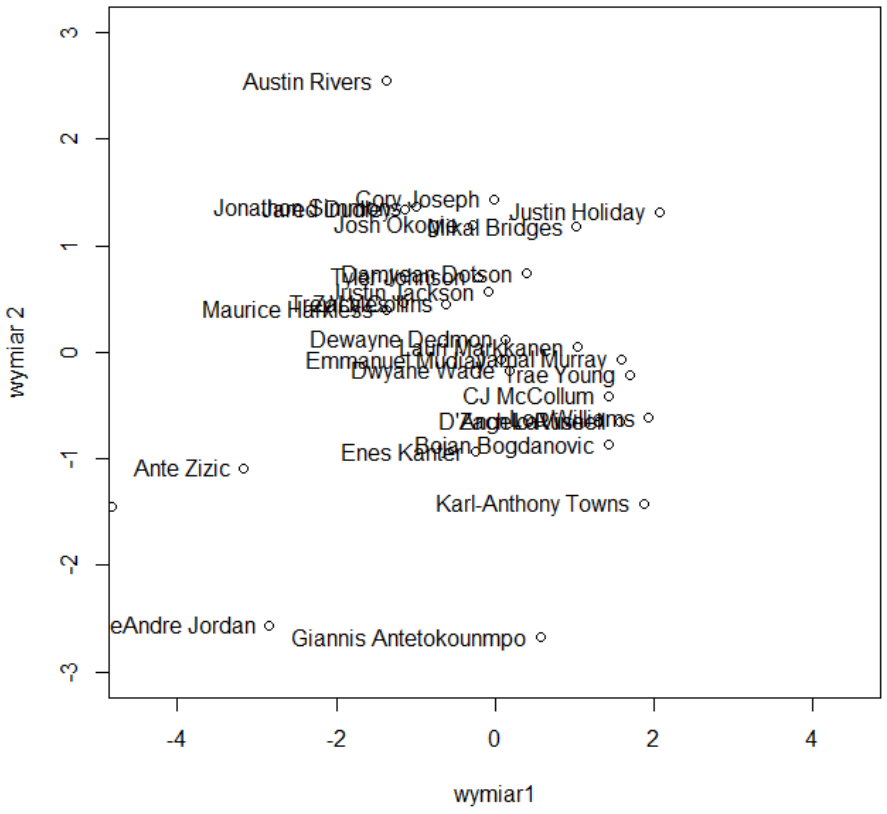
$$\mathrm{STRESS} = \sqrt{\frac{\sum_{ij}\left(f\left(d_{ij}\right) - \tilde{d}_{ij}\right)^2}{f\left(d_{ij}\right)^2}}$$

STRESS	Dopasowanie
>20%	bardzo słabe
10-20%	słabe
5-10%	średnie
2-5%	dobre
0-2%	bardzo dobre
0%	idealne

Zaczynamy od standaryzacji zmiennych, następnie tworząc macierz odległości rzeczywistych użytych elementów. Skaluje obiekty do przestrzeni dwuwymiarowej.

Obliczam współczynnik STRESS, żeby sprawdzić w jakim stopniu odległości są dopasowane.

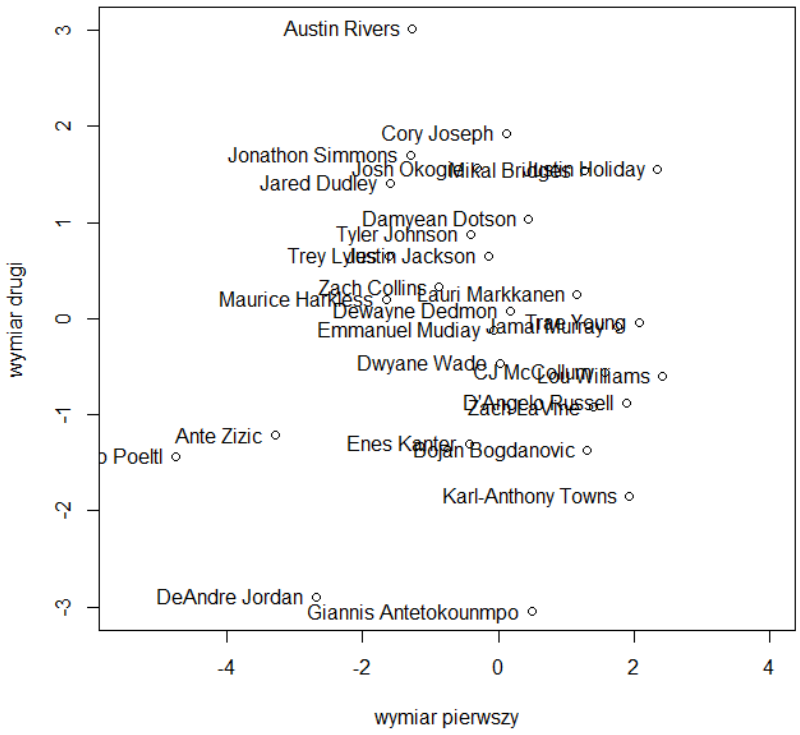
```
#metoda kruskala
mds2<-isoMDS(odl,k=2)
mds2$stress
mds2$points[,1]<--1*mds2$points[,1]
 plot(mds2$points, xlab="wymiar1", ylab="wymiar 2",xlim=c(-4.5,4.5),ylim=c(-3,3))
text(mds2$points,labels=rownames(dane3),pos=2)
```



Współczynnik STRESS wyszedł zdecydowanie za wysoki, więc od razu przechodzimu do meteody Sammona

Metoda Sammona

```
# metoda sammona
library(MASS)
??MASS::sammon

sammon1<-sammon(odl,k=2)
sammon1$points
sammon1$stress
sammon1$points[,1]<--1*sammon1$points[,1]

plot(sammon1$points, xlab="wymiar pierwszy", ylab="wymiar drugi",xlim=c(-5.5,4),ylim=c(-3,3))
text(sammon1$points,labels=rownames(dane3),pos=2)
```




```
sammon1$stress
1] 0.02492965
```

Skaluje obiekty do przestrzeni dwuwymiarowej i jak widać na załączonym wyżej obrazku współczynnik STRESS jest mniejszy 5% co oznacza dobre dopasowanie.

Można zatem wnioskować, że metoda Sammona dla moich danych sprawdziła się zdecydowanie lepiej niż Kruskala.

Klasyczne skalowanie wielowymiarowe

```
dane3<-subset(dane,select=c(MIN, PTS, FG., X3P., FT.))
dane_st3<-as.data.frame(scale(dane3))

odl<-dist(dane_st3)
odl
library(stats)
??stats::cmdscale
library(graphics)
??graphics::text

#dwu wymiarowe
sww2<-cmdscale(odl,k=2)

sww2

sww2[,1]<--1*sww2[,1]
plot(sww2, xlab="wymiar pierwszy", ylab="wymiar drugi",
     xlim=c(-4.5,4),ylim=c(-3,3))
text(sww2,labels=rownames(dane3),pos=2)

stress<-function(d1, d2)
{
  sqrt(sum((d1-d2)^2)/sum(d1^2))
}

odl

odl2<-dist(sww2)
odl2
```

```
> stress(odl,odl2)
[1] 0.2128067
>
```

Jak widać na powyższym obrazku, współczynnik STRESS dla przestrzeni dwuwymiarowej jest równy 21,3% co oznacza bardzo słabe dopasowanie.