

Self-organizing map

Monika Osiak, Anna Pręgowska, Patrycja Szczepaniak, Rafał Szulejko

08 05 2020

Wstęp

Zadaniem było wykonanie analizy wybranego zbioru danych z wykorzystaniem samoorganizującej się mapy (SOM). https://clarkdatalabs.github.io/soms/SOM_NBA

```
head(NBA)
```

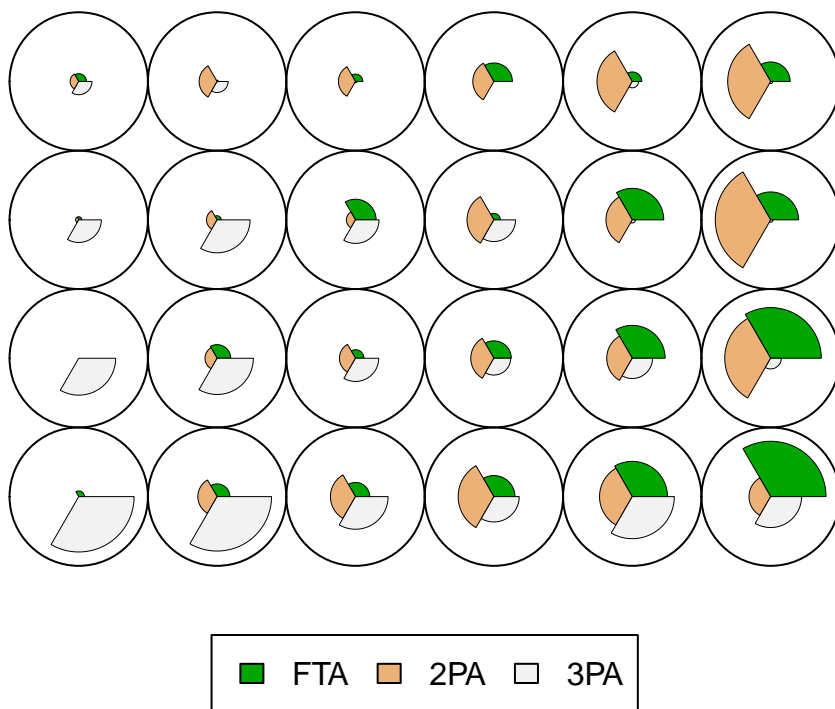
##		Player	Pos	Age	Tm	G	GS	MP	FG	FGA	FG%	3P	3PA				
## 1	1	Quincy Acy	Power Forward	25	SAC	59	29	876	4.9	8.8	0.556	0.8	2.0				
## 2	2	Jordan Adams	Shooting Guard	21	MEM	2	0	15	4.8	14.4	0.333	0.0	2.4				
## 3	4	Arron Afflalo	Shooting Guard	30	NYK	71	57	2371	5.4	12.1	0.443	1.4	3.6				
## 4	5	Alexis Ajinca	Center	27	NOP	59	17	861	6.3	13.2	0.476	0.0	0.0				
## 5	7	LaMarcus Aldridge	Power Forward	30	SAS	74	74	2261	8.5	16.6	0.513	0.0	0.3				
## 6	10	Tony Allen	Shooting Guard	34	MEM	64	57	1620	4.8	10.4	0.458	0.3	0.9				
##		3P%	2P	2PA	2P%	FT	FTA	FT%	ORB	DRB	TRB	AST	STL	BLK	TOV	PF	PTS
## 1		0.388	4.1	6.8	0.606	2.1	2.8	0.735	2.7	5.1	7.7	1.1	1.2	1.0	1.1	4.2	12.6
## 2		0.000	4.8	12.0	0.400	7.2	12.0	0.600	0.0	4.8	4.8	7.2	7.2	0.0	4.8	4.8	16.8
## 3		0.382	4.0	8.5	0.469	1.7	2.0	0.840	0.3	3.7	4.0	2.2	0.4	0.2	1.2	2.2	13.8
## 4		0.000	6.3	13.1	0.478	2.2	2.6	0.839	3.1	8.1	11.2	1.3	0.8	1.5	2.3	5.6	14.7
## 5		0.000	8.5	16.4	0.521	4.1	4.8	0.858	2.8	7.3	10.1	1.8	0.6	1.3	1.6	2.4	21.2
## 6		0.357	4.4	9.5	0.468	2.0	3.1	0.652	2.3	4.3	6.6	1.6	2.4	0.4	1.7	3.9	11.9

```
colnames(NBA)
```

##	[1]	"	"Player"	"Pos"	"Age"	"Tm"	"G"	"GS"	"MP"
##	[9]	"FG"	"FGA"	"FG%"	"3P"	"3PA"	"3P%"	"2P"	"2PA"
##	[17]	"2P%"	"FT"	"FTA"	"FT%"	"ORB"	"DRB"	"TRB"	"AST"
##	[25]	"STL"	"BLK"	"TOV"	"PF"	"PTS"			

Basic SOM

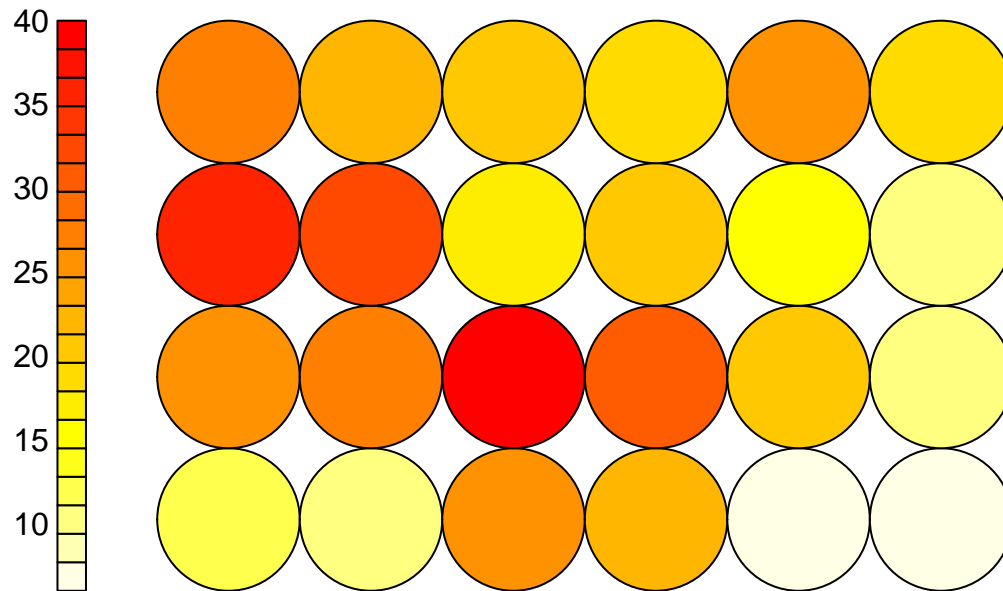
```
NBA.measures1 <- c("FTA", "2PA", "3PA")
NBA.SOM1 <- som(scale(NBA[NBA.measures1]), grid = somgrid(6, 4, "rectangular"))
plot(NBA.SOM1)
```



Heatmap SOM

```
# reverse color ramp
colors <- function(n, alpha = 1) {
  rev(heat.colors(n, alpha))
}
plot(NBA.SOM1, type = "counts", palette.name = colors, heatkey = TRUE)
```

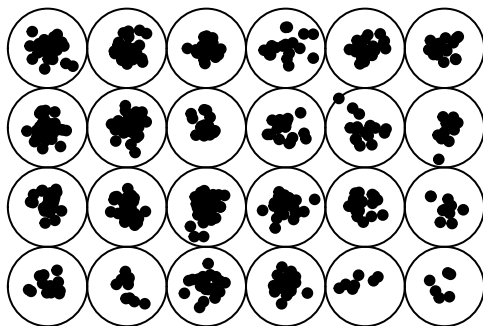
Counts plot



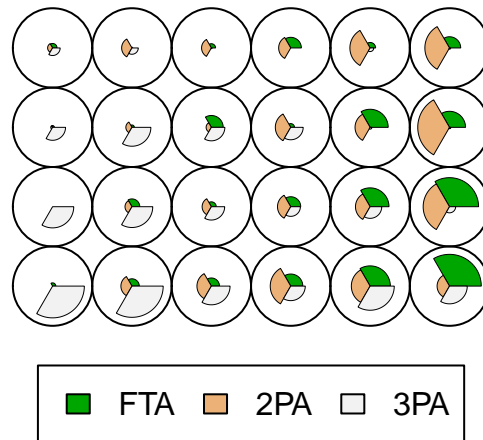
Plotting points

```
par(mfrow = c(1, 2))
plot(NBA.SOM1, type = "mapping", pchs = 20, main = "Mapping Type SOM")
plot(NBA.SOM1, main = "Default SOM Plot")
```

Mapping Type SOM



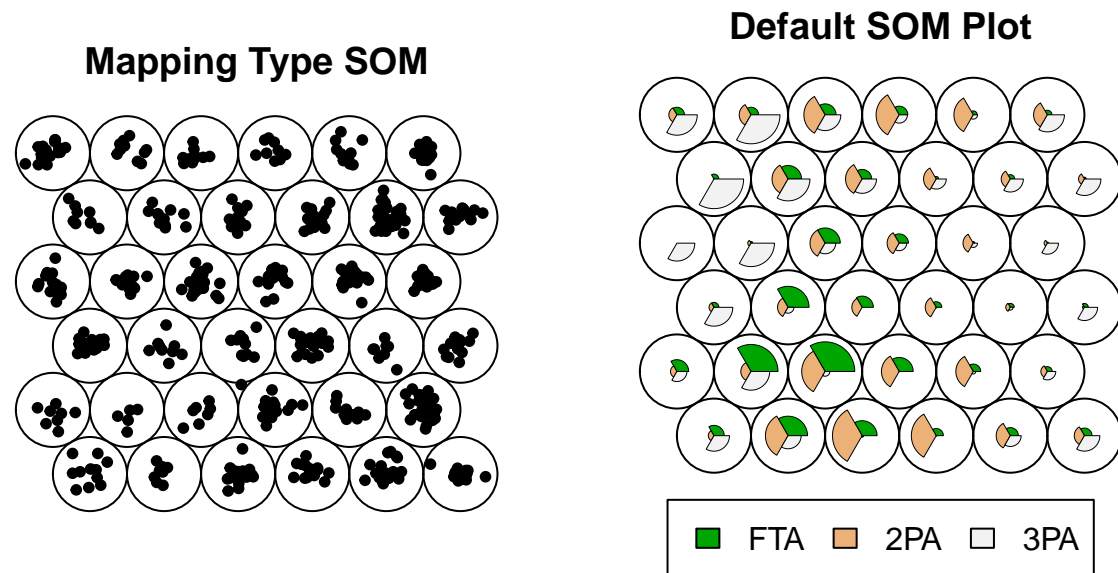
Default SOM Plot



Toroidal SOMs

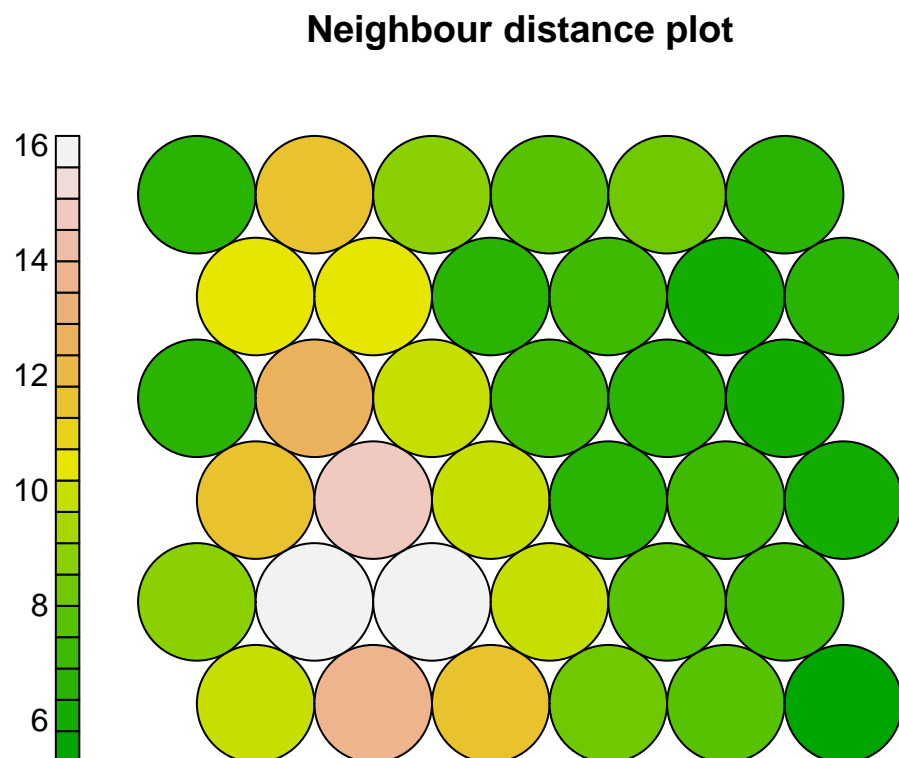
```
NBA.SOM2 <- som(scale(NBA[NBA.measures1]), grid = somgrid(6, 6, "hexagonal"), toroidal = TRUE)
#NBA.SOM2 <- som(scale(NBA[NBA.measures1]), grid = somgrid(6, 6, "hexagonal"))
par(mfrow = c(1, 2))
```

```
plot(NBA.SOM2, type = "mapping", pchs = 20, main = "Mapping Type SOM")
plot(NBA.SOM2, main = "Default SOM Plot")
```



Mapping Distance

```
plot(NBA.SOM2, type = "dist.neighbours", palette.name = terrain.colors)
```



Supervised SOMs

```
NBA.measures2 <- c("FTA", "FT", "2PA", "2P", "3PA", "3P", "AST", "ORB", "DRB",  
  "TRB", "STL", "BLK", "TOV")
```

The xyf() Function

```
training_indices <- sample(nrow(NBA), 200)  
NBA.training <- scale(NBA[training_indices, NBA.measures2])  
NBA.testing <- scale(NBA[-training_indices, NBA.measures2], center = attr(NBA.training,  
  "scaled:center"), scale = attr(NBA.training, "scaled:scale"))
```

```
NBA.SOM3 <- xyf(NBA.training, classvec2classmat(NBA$Pos[training_indices]), grid = somgrid(13, 13, "hex")  
#NBA.SOM3 <- xyf(NBA.training, classvec2classmat(NBA$Pos[training_indices]), grid = somgrid(13, 13, "hex"))
```

```
# !!! w poniższej linijce wywala błąd "data type not allowed"!!!  
# po znalezieniu błędu odkomentować obie linijki  
# PS: nie wiem jaki był błąd, bo ładnie mi wszystko przeszło :p ale stosowałam się dokładnie do tutoria
```

```
pos.prediction <- predict(NBA.SOM3, newdata = NBA.testing)  
table(NBA[-training_indices, "Pos"], pos.prediction$prediction)
```

```
##  
##           Center Point Guard Power Forward Shooting Guard Small Forward  
## Center           29           0           5           1           4  
## Point Guard      0           57           1           5           4  
## Power Forward    16           2          26           9           8  
## Shooting Guard   1           11           7          37          24  
## Small Forward    0           5           7          13          31
```

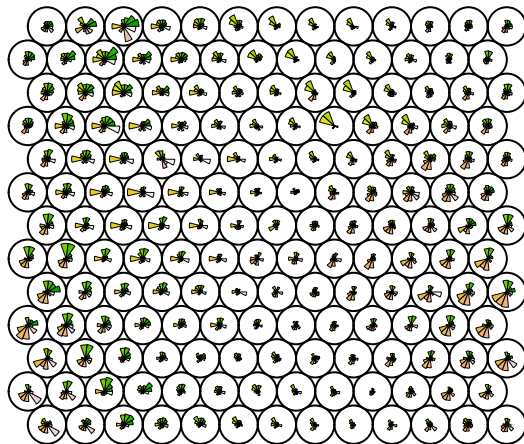
Visualizing predictions: “Codes” SOMs

```
NBA.SOM4 <- xyf(scale(NBA[, NBA.measures2]), classvec2classmat(NBA[, "Pos"]), grid = somgrid(13, 13, "hex")  
#NBA.SOM4 <- xyf(scale(NBA[, NBA.measures2]), classvec2classmat(NBA[, "Pos"]), grid = somgrid(13, 13, "hex"))
```

```
par(mfrow = c(1, 2))
```

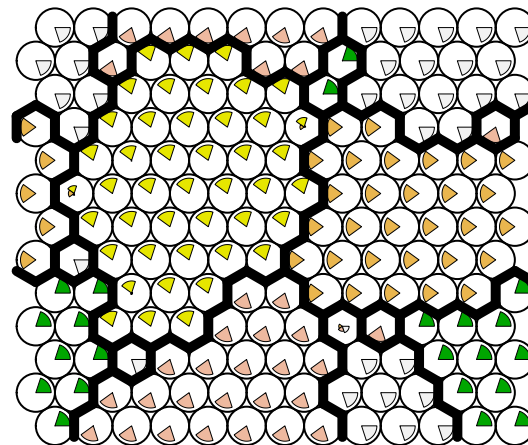
```
plot(NBA.SOM4, type = "codes", main = c("Codes X", "Codes Y"))  
NBA.SOM4.hc <- cutree(hclust(dist(NBA.SOM4$codes$Y)), 5)  
add.cluster.boundaries(NBA.SOM4, NBA.SOM4.hc)
```

Codes X



■ FTA	■ 3P	■ STL
■ FT	■ AST	■ BLK
■ 2PA	■ ORB	■ TOV
■ 2P	■ DRB	
■ 3PA	■ TRB	

Codes Y



■ Center	■ Power Forward	■ Small Forward
■ Point Guard	■ Shooting Guard	

```
# !!! błąd w poniższej linijce !!!
# odkomentować obie po rozwiązaniu
#NBA.SOM4.hc <- cutree(hclust(dist(NBA.SOM4$codes$Y)), 5)
#add.cluster.boundaries(NBA.SOM4, NBA.SOM4.hc)
# PS: tu też przeszło (gdy przeniosłam te dwie linijki bezpośrednio pod plot, bo ewidentnie dotyczą rys
```

```
bg.pallet <- c("red", "blue", "yellow", "purple", "green")
```

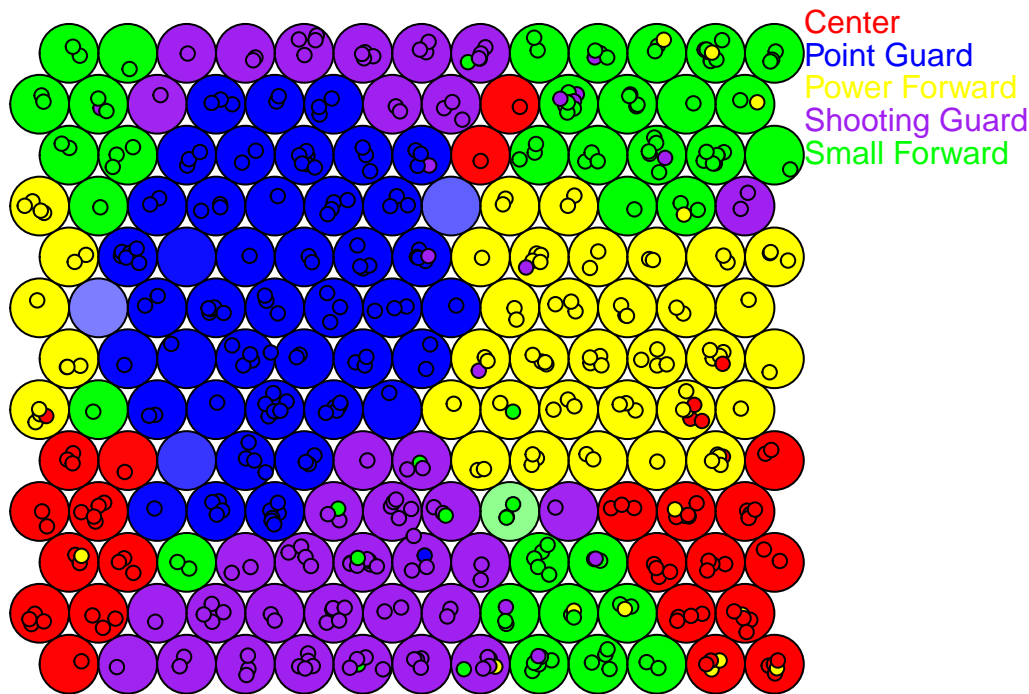
```
position.predictions <- classmat2classvec(predict(NBA.SOM4)$unit.predictions)
base.color.vector <- bg.pallet[match(position.predictions, levels(NBA$Pos))]
```

```
bgcols <- c()
max.conf <- apply(NBA.SOM4$codes$Y, 1, max)
for (i in 1:length(base.color.vector)) {
  bgcols[i] <- adjustcolor(base.color.vector[i], max.conf[i])
}
```

```
par(mar = c(0, 0, 0, 4), xpd = TRUE)
plot(NBA.SOM4, type = "mapping", pchs = 21, col = "black", bg = bg.pallet[match(NBA$Pos,
  levels(NBA$Pos))], bgcol = bgcols)
```

```
legend("topright", legend = levels(NBA$Pos), text.col = bg.pallet, bty = "n",
  inset = c(-0.03, 0))
```

Mapping plot



#to już prawdopodobnie jest niepotrzebne

Podsumowanie i wnioski

NA RAZIE PISZĘ TU TO CO TRZEBABY BYŁO ZROBIĆ:

#1. sprawdzenie o co chodzi z błędem “unused arguments”. Przez niego zakomentowałam niektóre linijki i pod spodem wrzuciłam ich wersje z usuniętymi problematycznymi argumentami (pytanie czy to czegoś nie zmienia?) - linia 65, 91, 104 # 2. sprawdzenie o co chodzi z błędami w liniach 95, 112 # 3. ostatnia sekcja cała zakomentowana, bo cała sypała błędami

PS: zrobione, wszystko przeszło już bez błędów

4. zastanowić się, czy to wszystko nam potrzebne i czy czegoś nie usunąć/dodać z innego źródła
5. sensownie podzielić na bloki
6. ustawić flagę include=FALSE przy blokach, z których kodu nie chcemy pokazywać
7. opisać kolejne kroki, co tam właściwie się dzieje
8. dodać wstęp (opis co my właściwie badamy, co to za zbiór, źródło do tutoriala) i wnioski

jeszcze może tu coś fajnego będzie: <https://www.shanelynn.ie/self-organising-maps-for-customer-segmentation-using-r/>