

Zastosowanie SOM dla danych giełdowych

Łukasz Knigawka, Bazyli Reps

Dokument przedstawia zastosowanie map samoorganizujących na danych giełdowych. Wykorzystano dane giełdowe z początku maja 2020, czyli obserwacje z czasów pandemii koronawirusa. Do pobrania danych wykorzystano bibliotekę quantmod. Analizie poddano dane spółek Amazon (AMZN), Apple (AAPL), Uber (UBER), Netflix (NFLX), Spotify (SPOT).

Mapy samoorganizujące to sieci neuronów, z którymi są stowarzyszone współrzędne na prostej, płaszczyźnie lub w dowolnej n-wymiarowej przestrzeni. Uczenie tego rodzaju sieci polega na zmianach współrzędnych neuronów, tak, by dążyły one do wzorca zgodnego ze strukturą analizowanych danych. Sieci zatem „rozpinają się” wokół zbiorów danych, dopasowując do nich swoją strukturę. Sieci te klasyfikują wielowymiarowe dane wejściowe w taki sposób, by możliwa była ich reprezentacji w mniejszej ilości wymiarów - przeważnie dwóch - przy jednoczesnym jak najwierniejszym odwzorowaniu struktury wewnętrznej wektora wejściowego.

```
if (!require("quantmod")) {  
  install.packages("quantmod")  
  library(quantmod)  
}
```

```
## Loading required package: quantmod  
## Loading required package: xts  
## Loading required package: zoo  
##  
## Attaching package: 'zoo'  
## The following objects are masked from 'package:base':  
##  
##   as.Date, as.Date.numeric  
## Loading required package: TTR  
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```
if (!require("kohonen")) {  
  install.packages("kohonen")  
  library(kohonen)  
}
```

```
## Loading required package: kohonen
```

Sprawdźmy, jak prezentowały się przez wybrane 4 miesiące kursy wybranych spółek.

```
start <- as.Date("2020-01-01")  
end <- as.Date("2020-05-01")  
  
getSymbols("AMZN", src = "yahoo", from = start, to = end)
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will  
## use auto.assign=FALSE in 0.5-0. You will still be able to use
```

```
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
## [1] "AMZN"
getSymbols("AAPL", src = "yahoo", from = start, to = end)

## [1] "AAPL"
getSymbols("UBER", src = "yahoo", from = start, to = end)

## [1] "UBER"
getSymbols("NFLX", src = "yahoo", from = start, to = end)

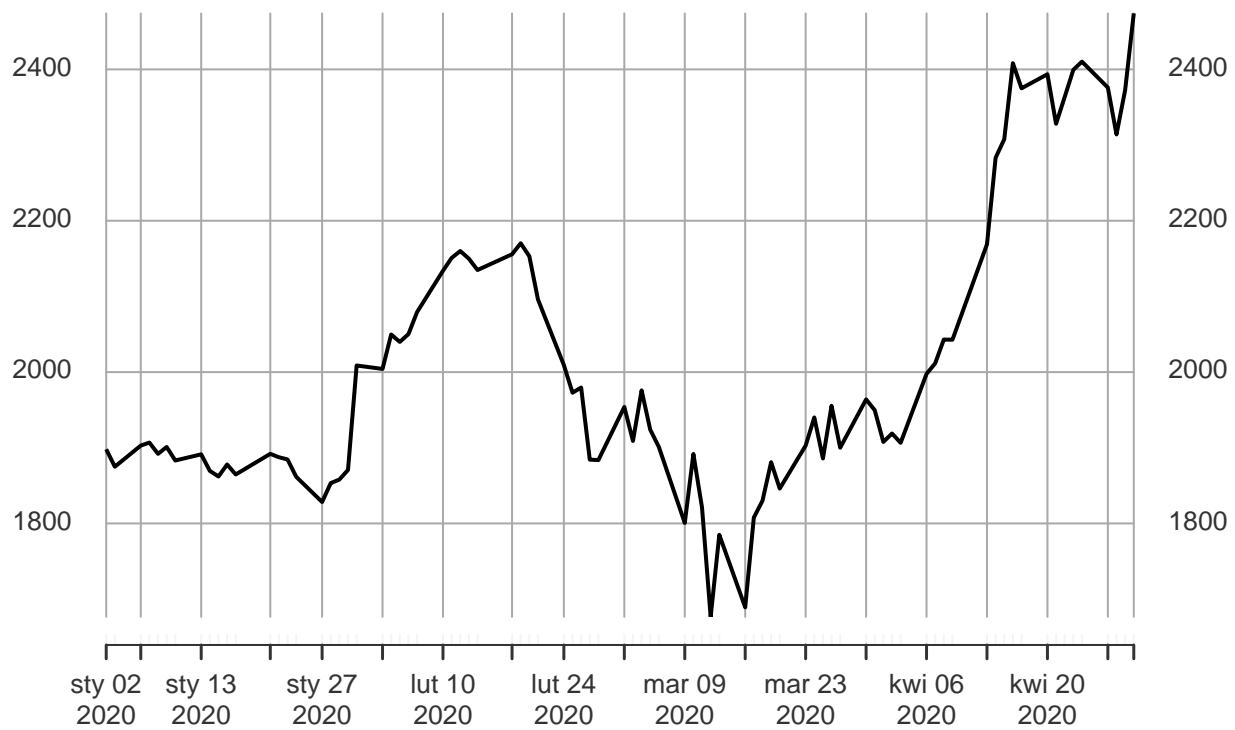
## [1] "NFLX"
getSymbols("SPOT", src = "yahoo", from = start, to = end)

## [1] "SPOT"
head(SPOT)

##           SPOT.Open SPOT.High SPOT.Low SPOT.Close SPOT.Volume SPOT.Adjusted
## 2020-01-02      151.00      152.80      149.61      151.62       662600       151.62
## 2020-01-03      149.50      153.59      149.50      152.50      1018400       152.50
## 2020-01-06      151.49      157.00      150.35      156.72      1311900       156.72
## 2020-01-07      156.70      157.85      155.01      156.02       876700       156.02
## 2020-01-08      156.33      159.48      155.34      158.78       974500       158.78
## 2020-01-09      157.84      159.98      157.11      157.74      1630600       157.74
plot(AMZN[, "AMZN.Close"], main = "AMZN")
```

AMZN

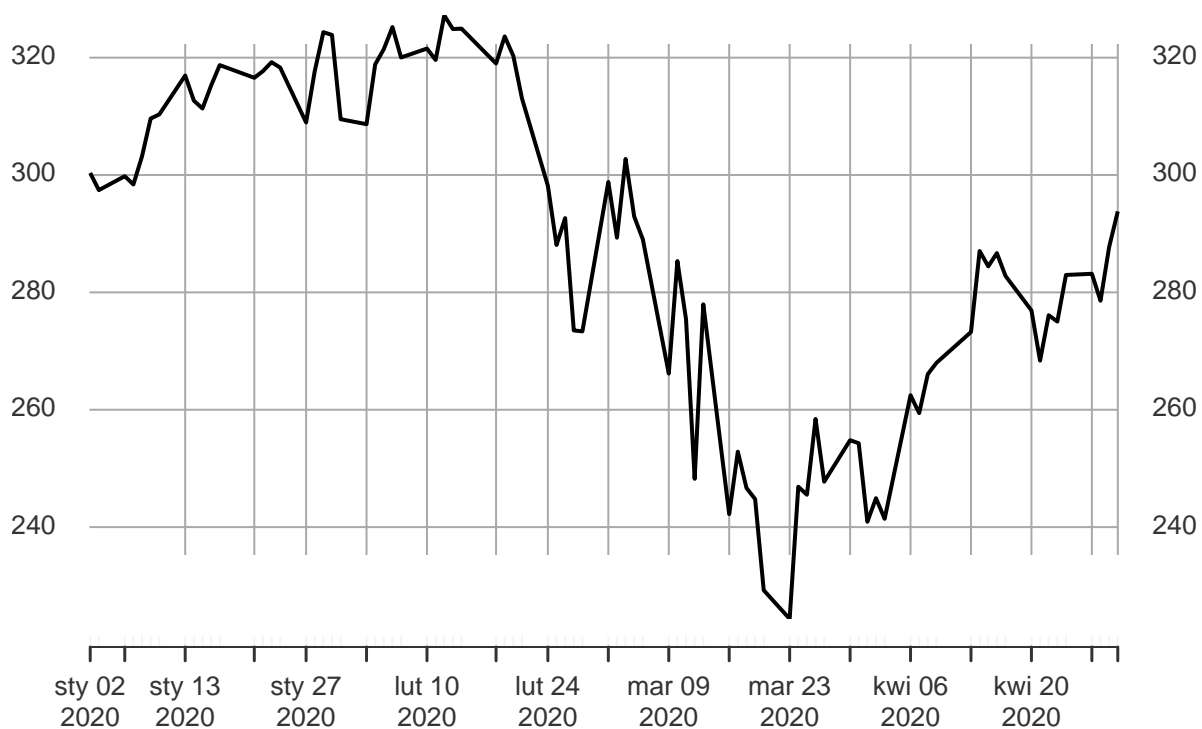
2020-01-02 / 2020-04-30



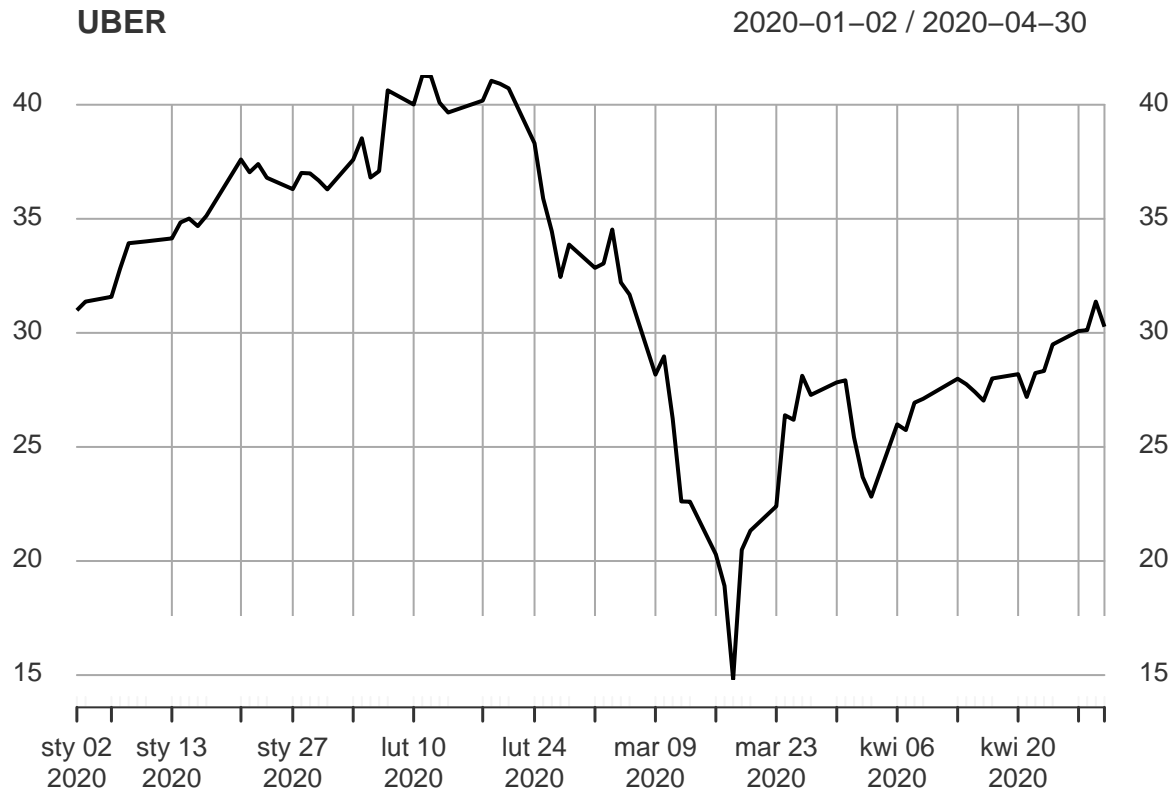
```
plot(AAPL[, "AAPL.Close"], main = "AAPL")
```

AAPL

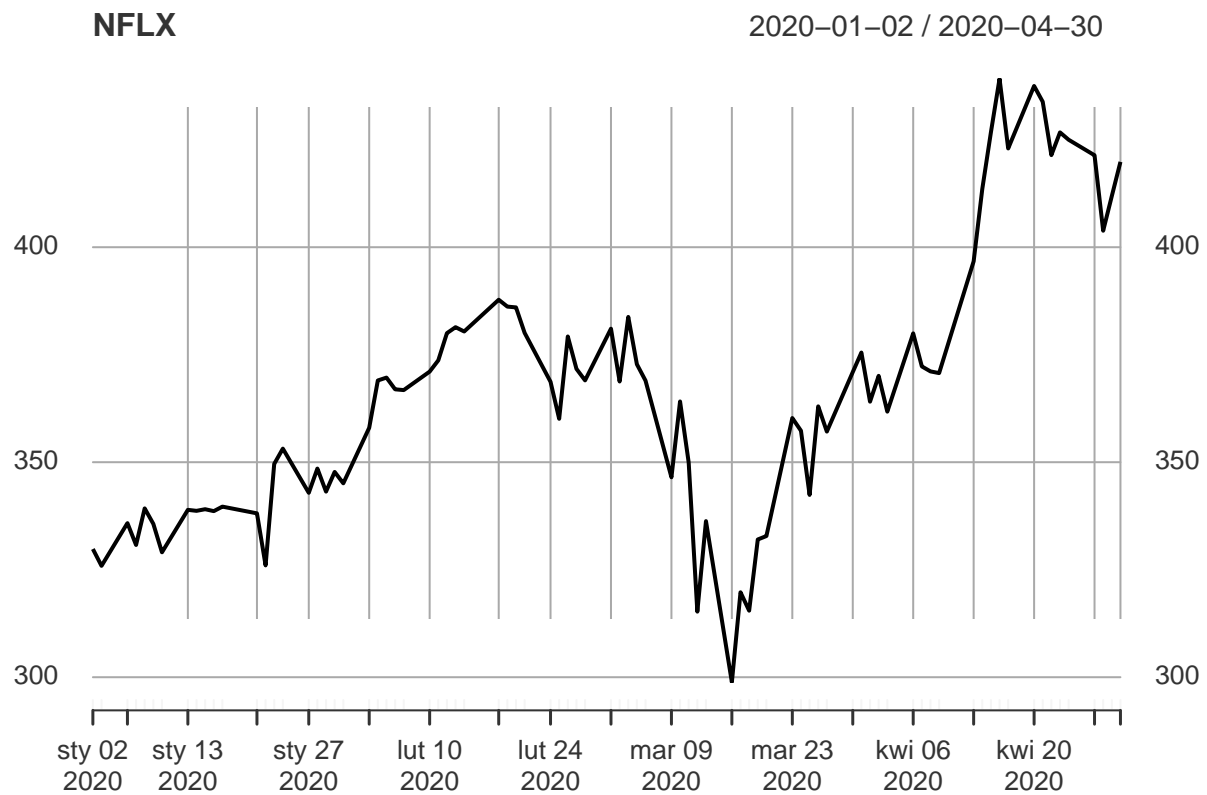
2020-01-02 / 2020-04-30



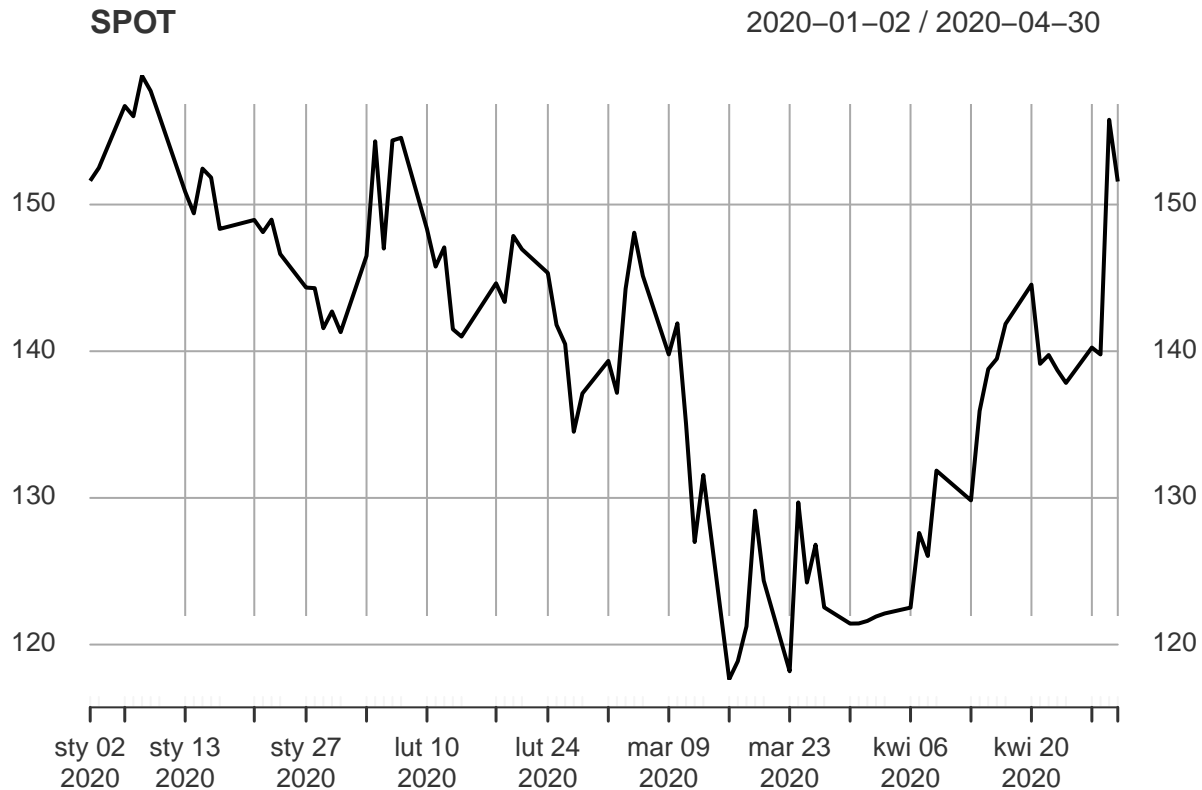
```
plot(UBER[, "UBER.Close"], main = "UBER")
```



```
plot(NFLX[, "NFLX.Close"], main = "NFLX")
```



```
plot(SPOT[, "SPOT.Close"], main = "SPOT")
```



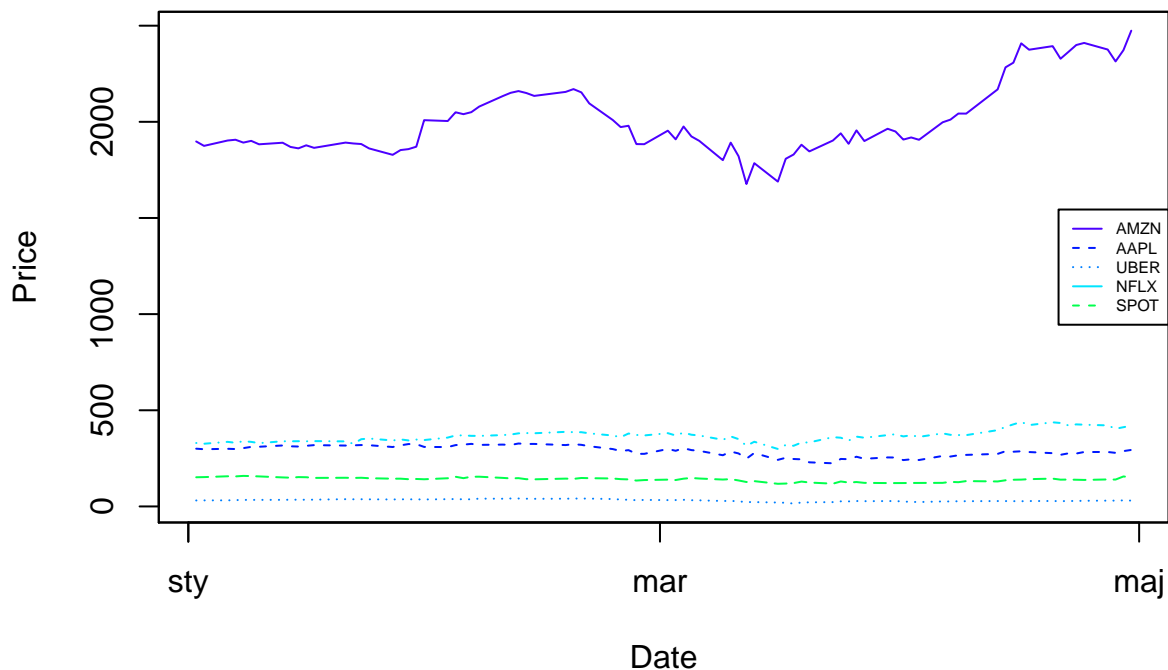
Możemy zaobserwować, że wszystkie wybrane spółki znacznie odczuły marcowe załamanie giełdowe. Niektóre z nich (Netflix, Amazon) zanotowały wkrótce wzrost wartości do poziomu wyższego niż przed załamaniem, podczas gdy inne nie wróciły na dawny poziom.

Sprawdźmy jeszcze na wspólnym wykresie jak prezentowały się ceny akcji wybranych spółek.

```
stocks <- as.xts(data.frame(AMZN = AMZN[, "AMZN.Close"], AAPL = AAPL[, "AAPL.Close"], UBER = UBER[, "UBER.Close"], NFLX = NFLX[, "NFLX.Close"], SPOT = SPOT[, "SPOT.Close"]),
head(stocks)
```

```
##          AMZN.Close AAPL.Close UBER.Close NFLX.Close SPOT.Close
## 2020-01-02    1898.01    300.35    30.99    329.81    151.62
## 2020-01-03    1874.97    297.43    31.37    325.90    152.50
## 2020-01-06    1902.88    299.80    31.58    335.83    156.72
## 2020-01-07    1906.86    298.39    32.81    330.75    156.02
## 2020-01-08    1891.97    303.19    33.93    339.26    158.78
## 2020-01-09    1901.05    309.63    33.97    335.66    157.74
```

```
plot(as.zoo(stocks), screens = 1, lty = 1:5, xlab = "Date", ylab = "Price", col = topo.colors(10))
legend("right", c("AMZN", "AAPL", "UBER", "NFLX", "SPOT"), lty = 1:3, cex = 0.5, col = topo.colors(10))
```



```
head(AMZN, 10)
```

##		AMZN.Open	AMZN.High	AMZN.Low	AMZN.Close	AMZN.Volume	AMZN.Adjusted
##	2020-01-02	1875.00	1898.01	1864.15	1898.01	4029000	1898.01
##	2020-01-03	1864.50	1886.20	1864.50	1874.97	3764400	1874.97
##	2020-01-06	1860.00	1903.69	1860.00	1902.88	4061800	1902.88
##	2020-01-07	1904.50	1913.89	1892.04	1906.86	4044900	1906.86
##	2020-01-08	1898.04	1911.00	1886.44	1891.97	3508000	1891.97
##	2020-01-09	1909.89	1917.82	1895.80	1901.05	3167300	1901.05
##	2020-01-10	1905.37	1906.94	1880.00	1883.16	2853700	1883.16
##	2020-01-13	1891.31	1898.00	1880.80	1891.30	2780800	1891.30
##	2020-01-14	1885.88	1887.11	1858.55	1869.44	3440900	1869.44
##	2020-01-15	1872.25	1878.86	1855.09	1862.02	2896600	1862.02

Przeskalowano dane przed wykorzystaniem ich do analizy map samoorganizujących. Do wykonania analizy wykorzystano bibliotekę kohonen.

```
AMZN.measures <- c("Open, High, Low, Close, Volume, Adjusted")
AMZN.sc = scale(AMZN[, 1:6])
head(AMZN.sc)
```

##		AMZN.Open	AMZN.High	AMZN.Low	AMZN.Close	AMZN.Volume
##	2020-01-02	-0.6835316	-0.7331929	-0.6099482	-0.5950270	-0.6180512
##	2020-01-03	-0.7388887	-0.7954912	-0.6080767	-0.7175760	-0.7214586
##	2020-01-06	-0.7626132	-0.7032311	-0.6321404	-0.5691237	-0.6052328
##	2020-01-07	-0.5280043	-0.6494254	-0.4608066	-0.5479543	-0.6118374
##	2020-01-08	-0.5620620	-0.6646703	-0.4907531	-0.6271537	-0.8216613
##	2020-01-09	-0.4995876	-0.6286949	-0.4407000	-0.5788571	-0.9548090

##		AMZN.Adjusted
##	2020-01-02	-0.5950270
##	2020-01-03	-0.7175760
##	2020-01-06	-0.5691237
##	2020-01-07	-0.5479543
##	2020-01-08	-0.6271537

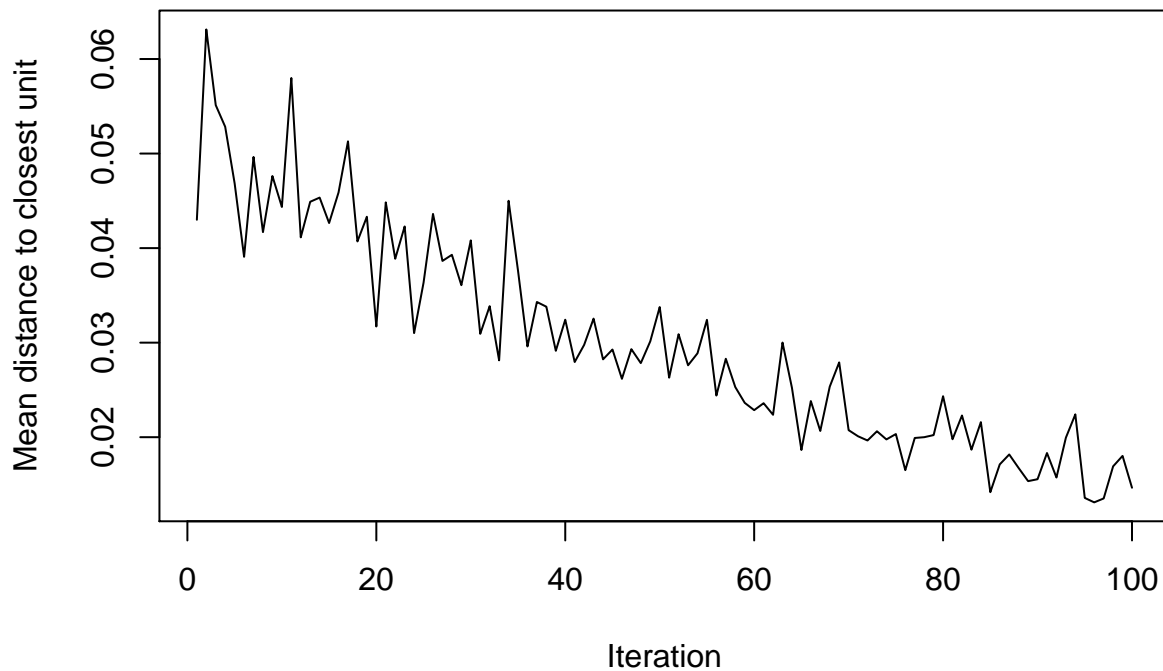
```
## 2020-01-09      -0.5788571
```

```
AMZN.grid = somgrid(xdim = 7, ydim=7, topo="hexagonal")
AMZN.som = som(AMZN.sc, grid=AMZN.grid, rlen=100, alpha=c(0.05,0.01))
```

Sprawdźmy jak przebiegał proces treningowy. Wraz z następowaniem kolejnych iteracji procesu uczenia, dystans między neuronami zmniejszał się. Możliwe że model jest nieco niedotrenowany, gdyż widzimy, że odległość maleje wciąż przy setnej iteracji.

```
plot(AMZN.som, type="changes")
```

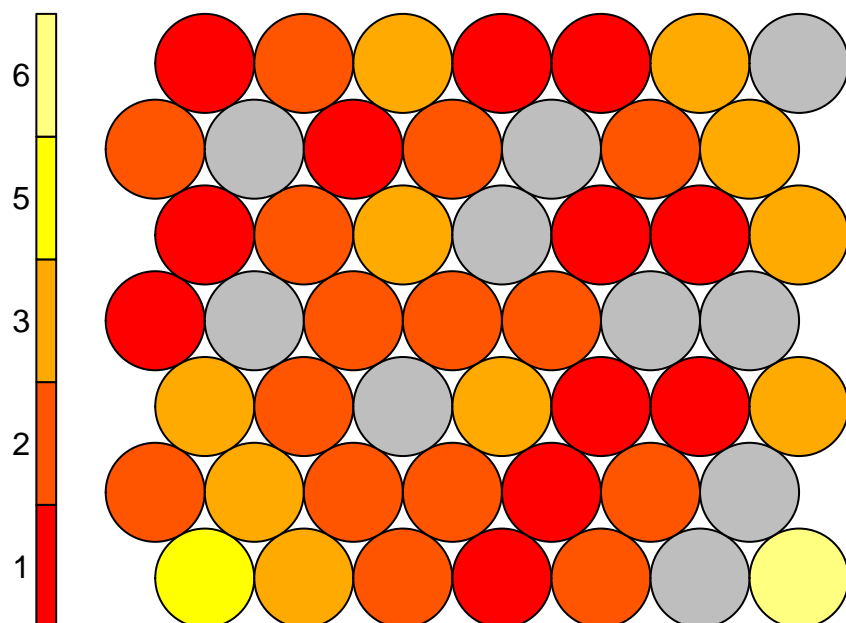
Training progress



Pakiet kohonen umożliwia przedstawienie wykresu na którym widać, ile obserwacji trafiło do której komórki.

```
plot(AMZN.som, type="count")
```

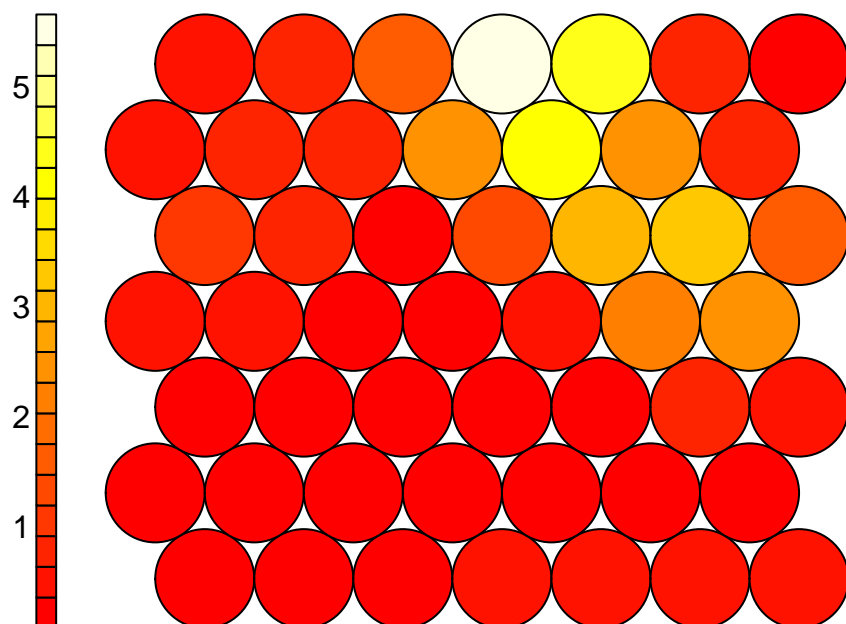
Counts plot



Poniższy wykres przedstawia odległości komórek od najbliższych sąsiadów. Taka wizualizacja nosi także nazwę U-macierzy.

```
plot(AMZN.som, type="dist.neighbours")
```

Neighbour distance plot



Poniższy wykres przedstawia charakterystykę obserwacji znajdujących się w danych komórkach.


```
plot(AMZN.som, type="codes")
```

Codes plot

