Raport

Jakub Wiśniewski

May 11, 2019

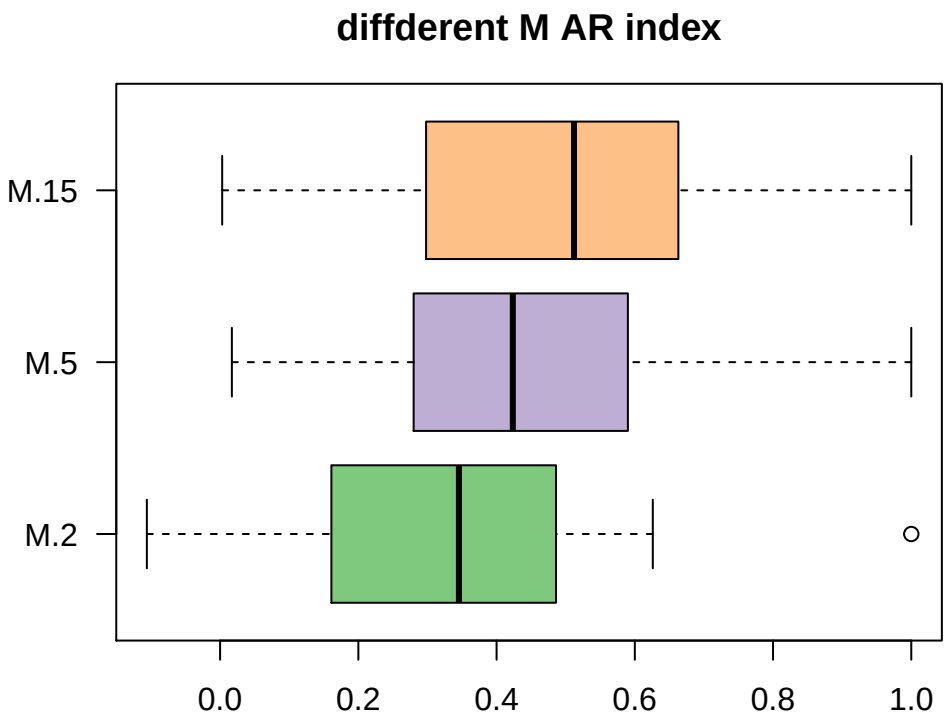
Raport ma na celu porównanie własnej implementacji algorytmu spektralnej analizy skupień, z dostępnymi gotowymi pakietami do tejże analizy. W skład analizy będą wchodziły: Spectral_Clustering - funkcja własnej implementacji, algorytm Genie autorstwa między innymi doktor Ceny i profesora Gągolewskiego, algorytmy hierarchiczne hclust oraz algorytmu PAM - Partitioning Around Medoids.

0. Wczytanie Danych

Wczytamy dane i biblioteki

```
ARall <- read.csv("all_AR")
FMall <- read.csv("FM_all")
ARstandaryzowane <- read.csv("AR_standaryzowane")
FMstandaryzowane <- read.csv("FM_standaryzowane")
rozneM <- read.csv("AR_rozneM")
```

1. Analiza własnego algorytmu przy zmianie parametru M dla indeks AR.



Ich mediany wynoszą kolejno:

```
##    M = 2  M = 5 M = 15
## 1 0.3455 0.4235  0.512
```

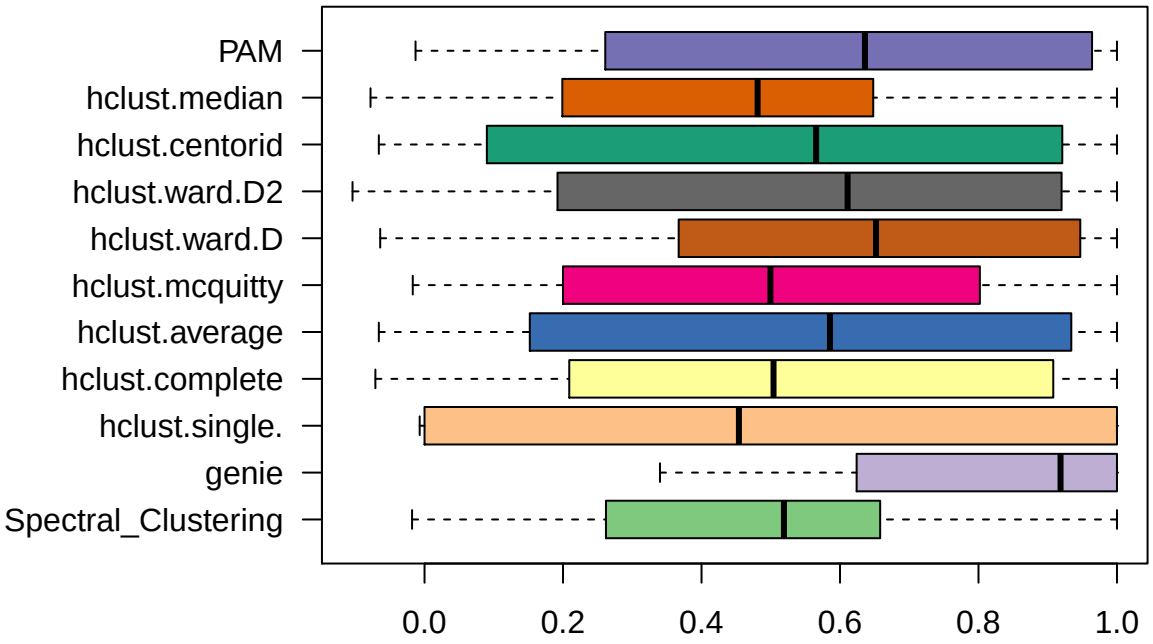
Wniosek:

Wraz ze wzrostem parametru M rośnie wydajność algorytmu.

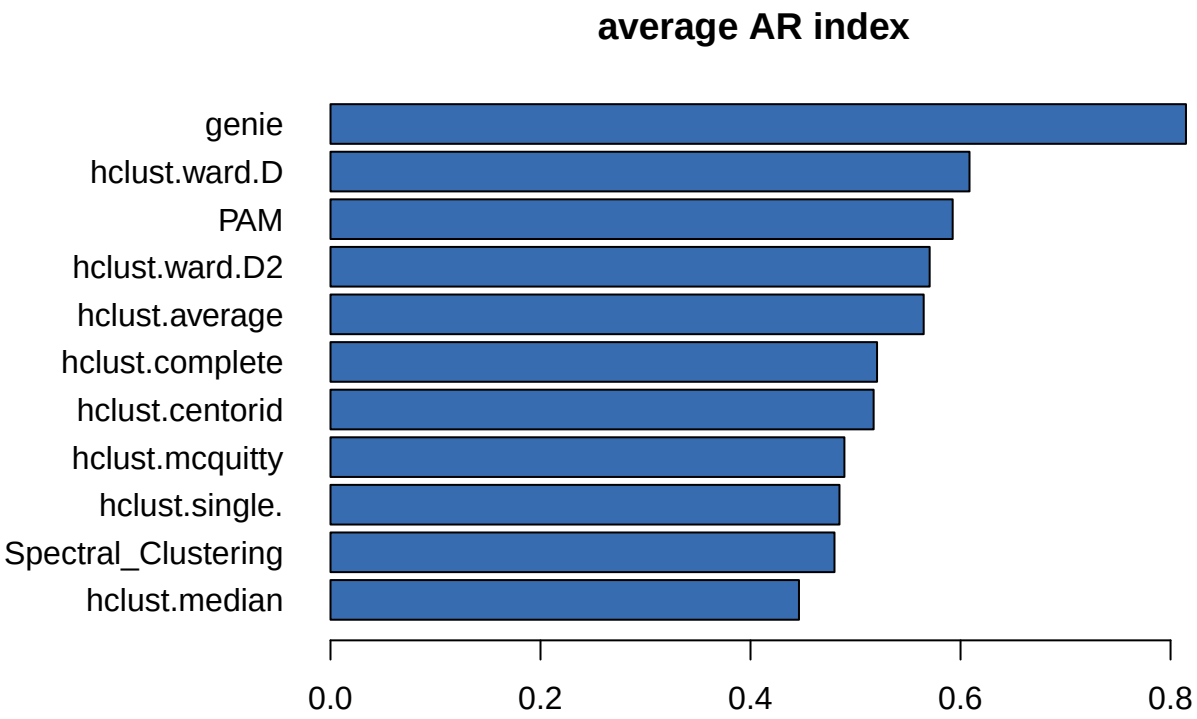
Uwaga
Następne analizy będę wykonywał dla M=10.

2. Analiza wyników według indeksów AR

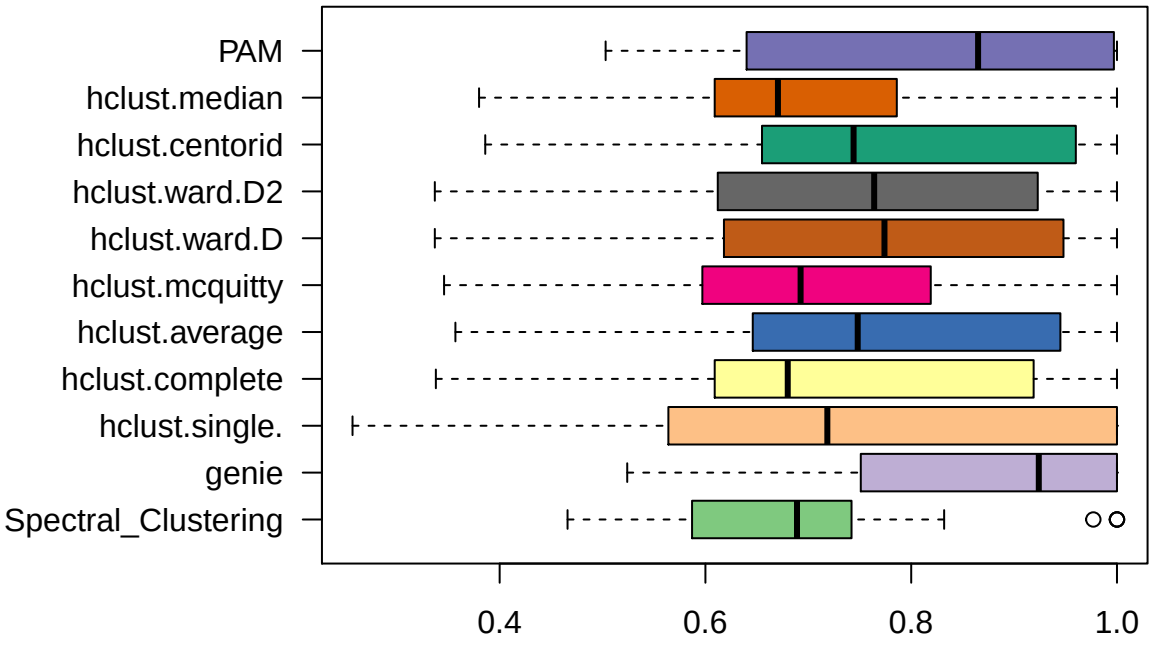
Tak prezentują się wyniki analizy dla indeksów AR



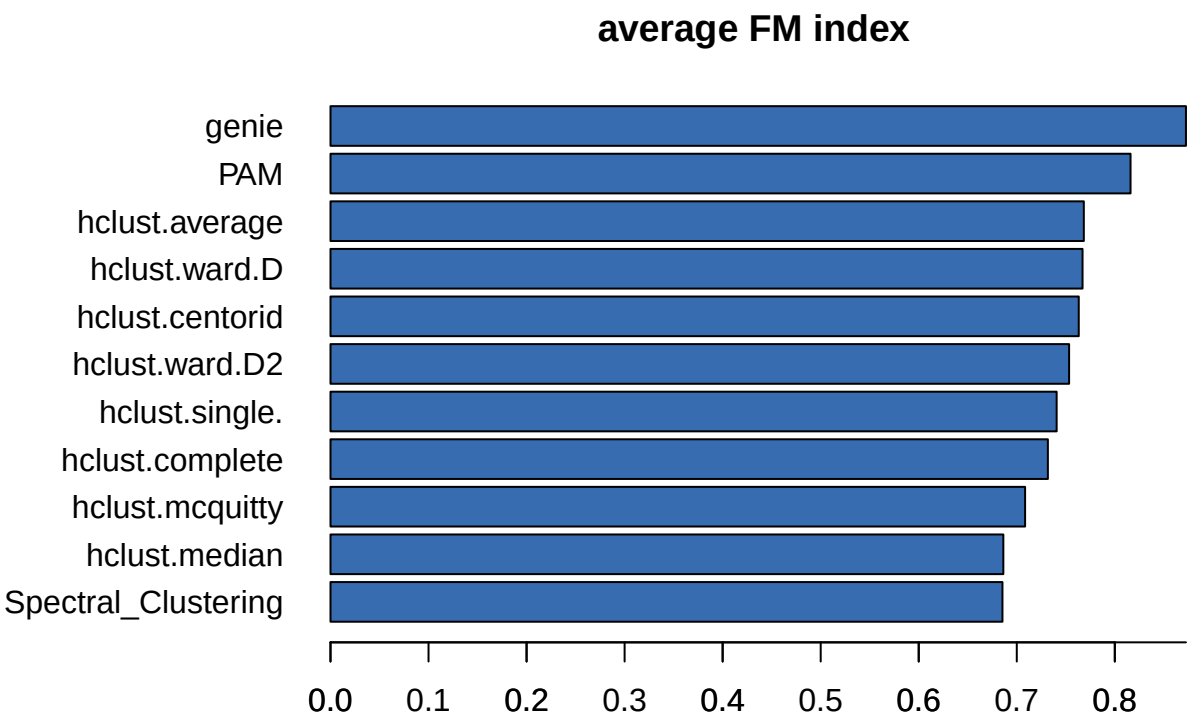
Jak możemy zauważyć najlepszym algorytmem w tym zestawieniu jest Genie. Sprawdźmy jednak jaka jest średnia trafność przydzielanych podzbiorów według indeksu AR.



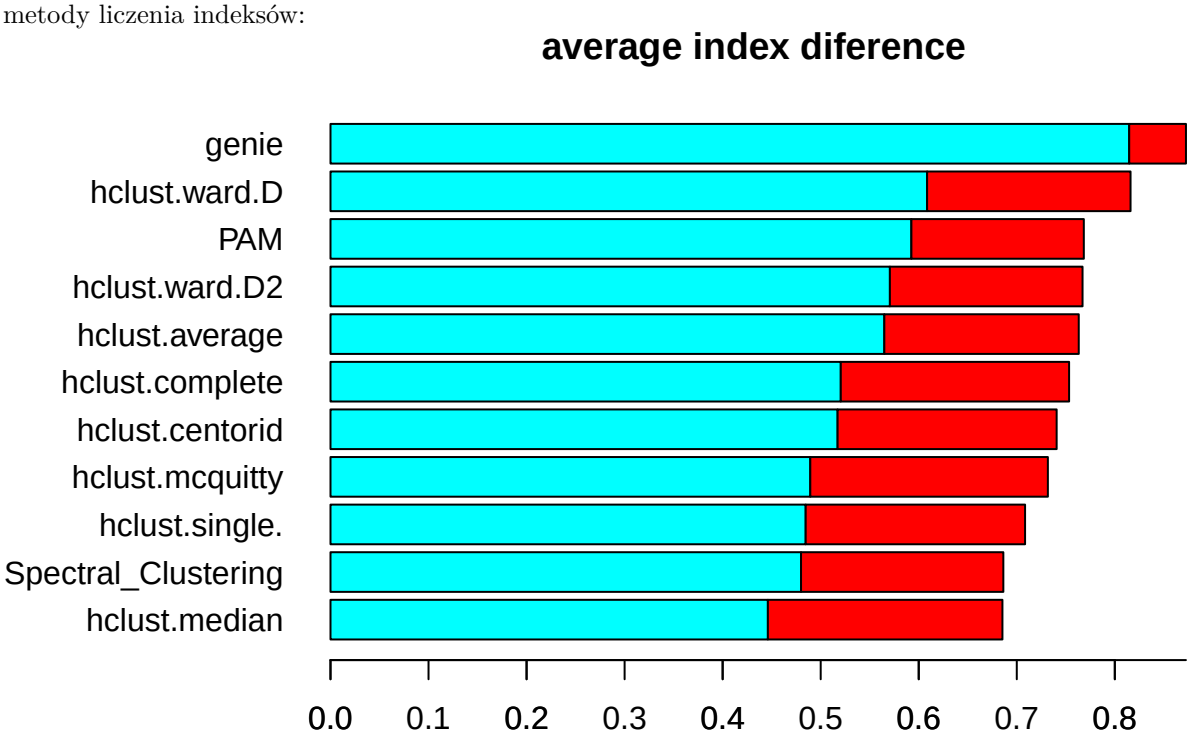
Największą średnią trafnością może pochwalić się Genie - 0.81, kolejny hclust.ward.D ma już 0.60. Algorytm własnej implementacji wypada przy nich gorzej - 0.47. Sprawdźmy teraz jak wyglądają te same dane dla indeksów FM.



Sprawdźmy teraz średnie



Sytuacja się powtórzyła, tym razem jednak wszystkie wartości się zwiększyły. Rekordzistą został Genie- 0,87, kolejny PAM- 0,81, spectral_clustering - 0,68. Zobaczmy jak bardzo trafność zwiększyła się po zmienieniu metody liczenia indeksów:



Co ciekawe największe i zarazem podobne (bo około 0,2) przyrosty osiągnęły wszystkie algorytmy, oprócz Genie, u której przyrost wyniósł 0,05.

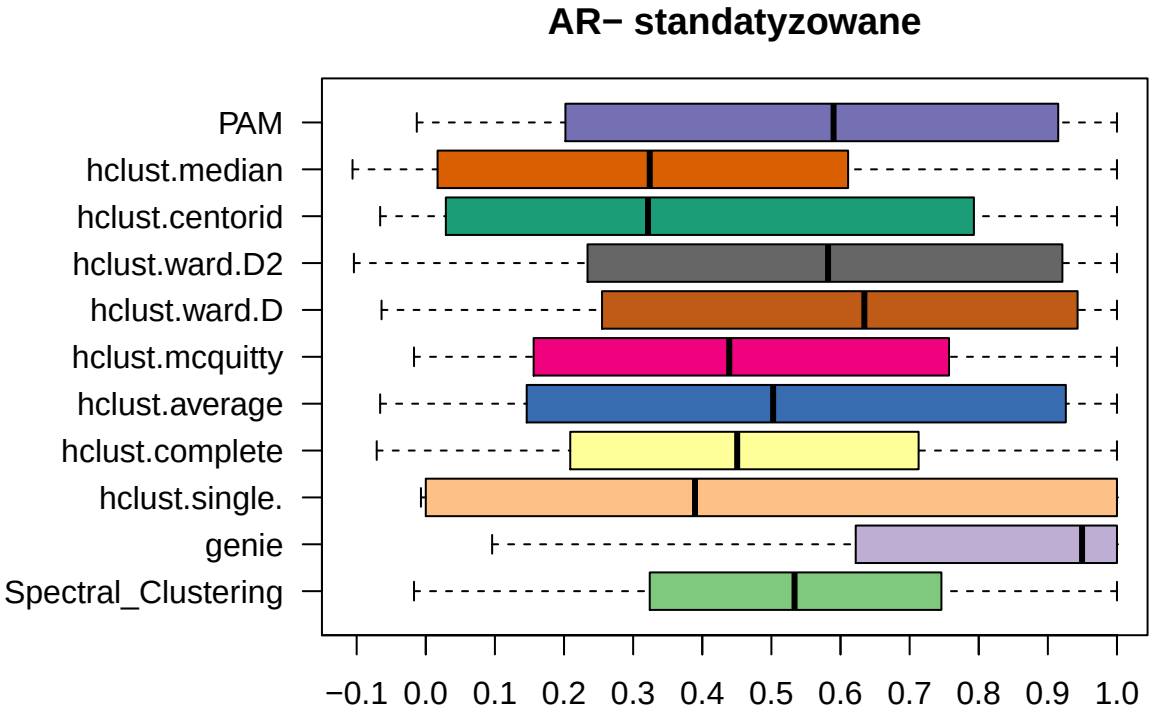
3. Standaryzowane Zmienne

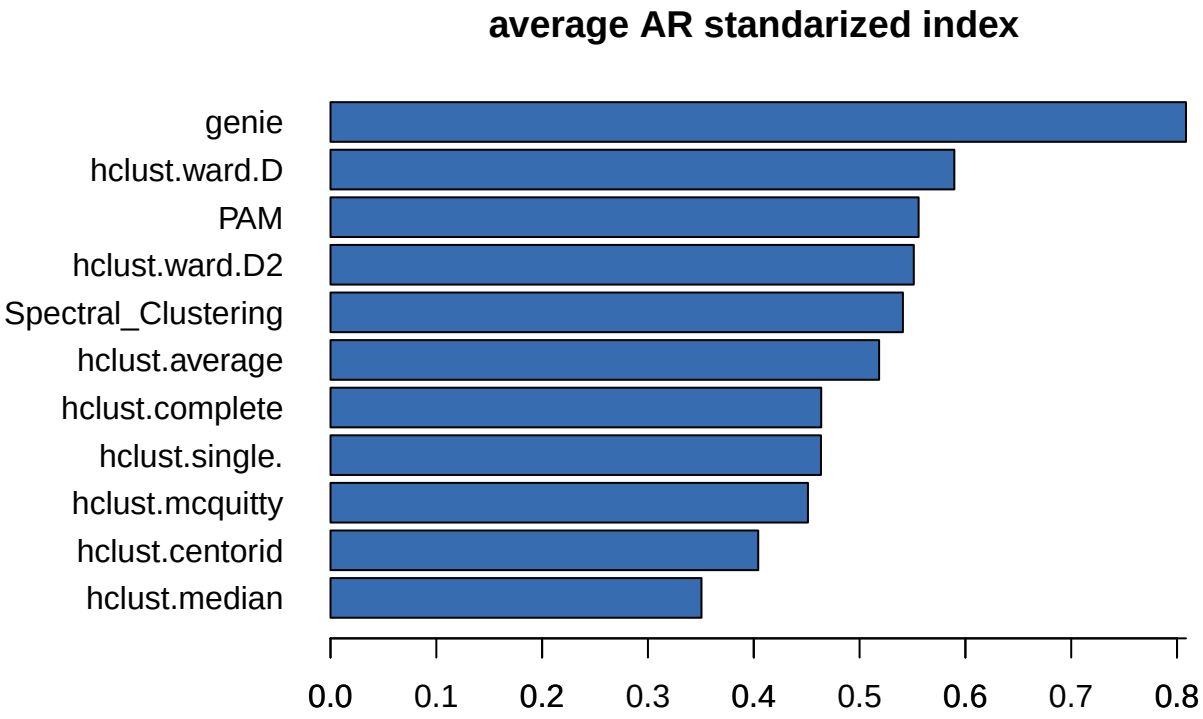
Zobaczmy, czy standaryzacja zmiennych pomaga, czy przeszkadza algorytmom w znalezieniu optymalnych podziałów zbioru.

Standaryzowałem zmienne za pomocą wzoru:

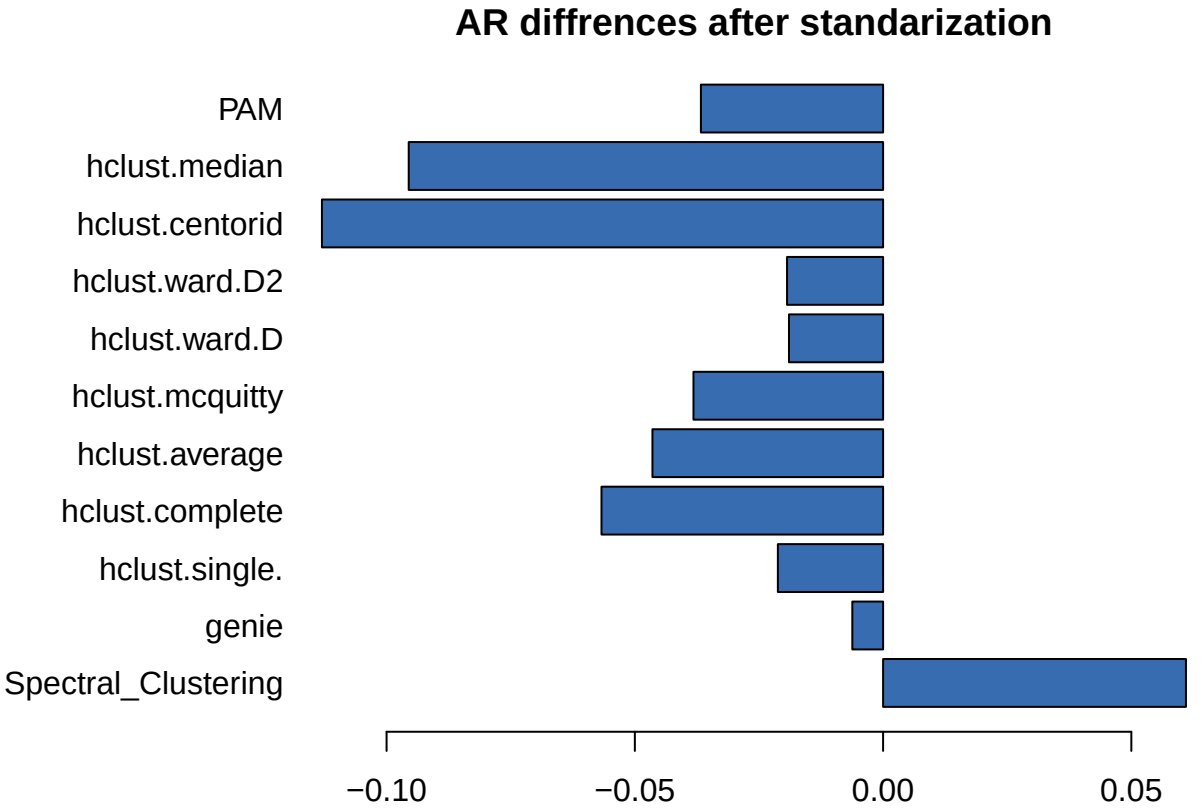
$$Z = \frac{X - \mu}{\sigma} \tag{1}$$

Sprawdźmy jak wyglądają teraz indeksy FM i AR

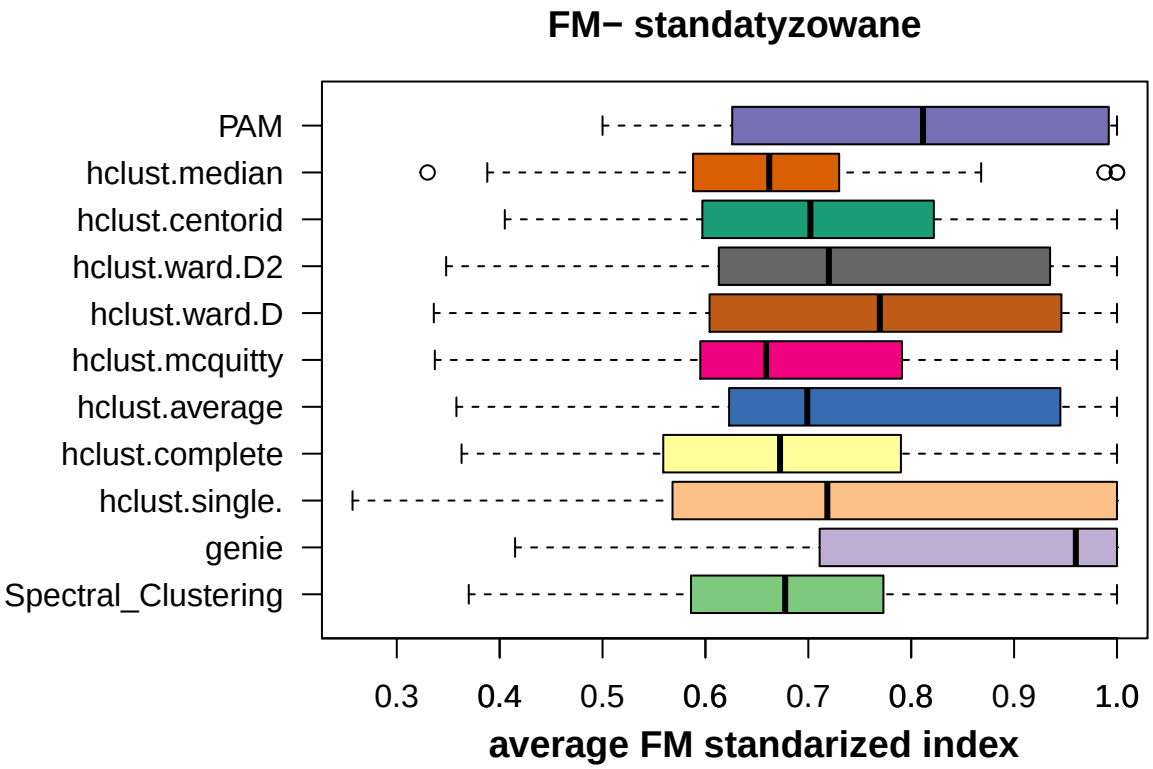


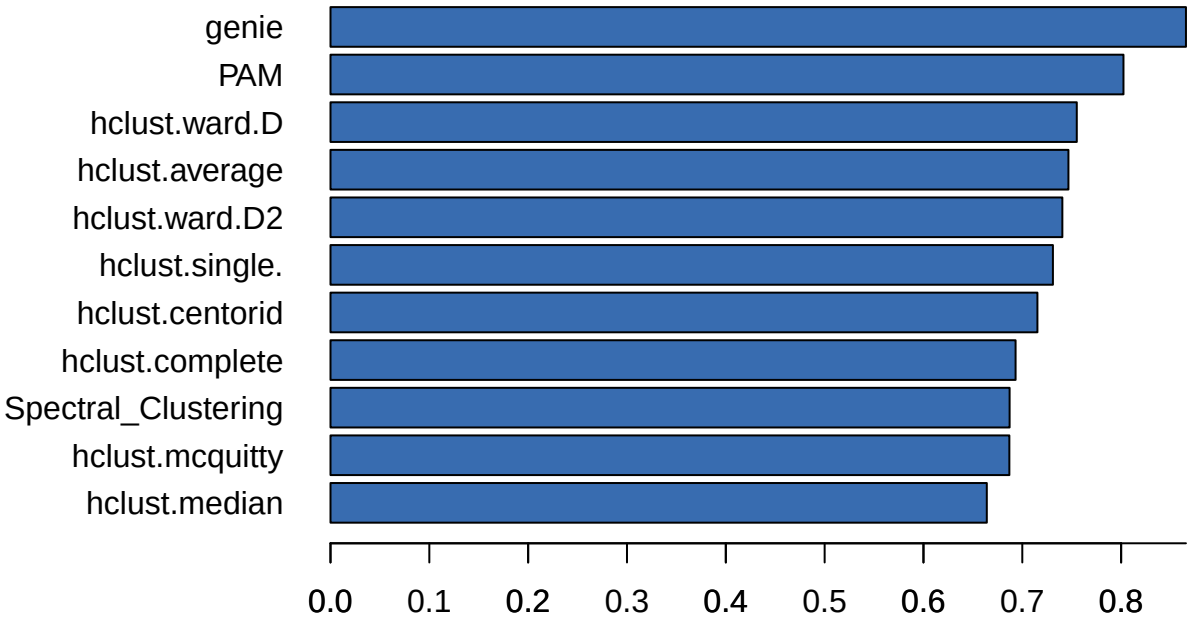


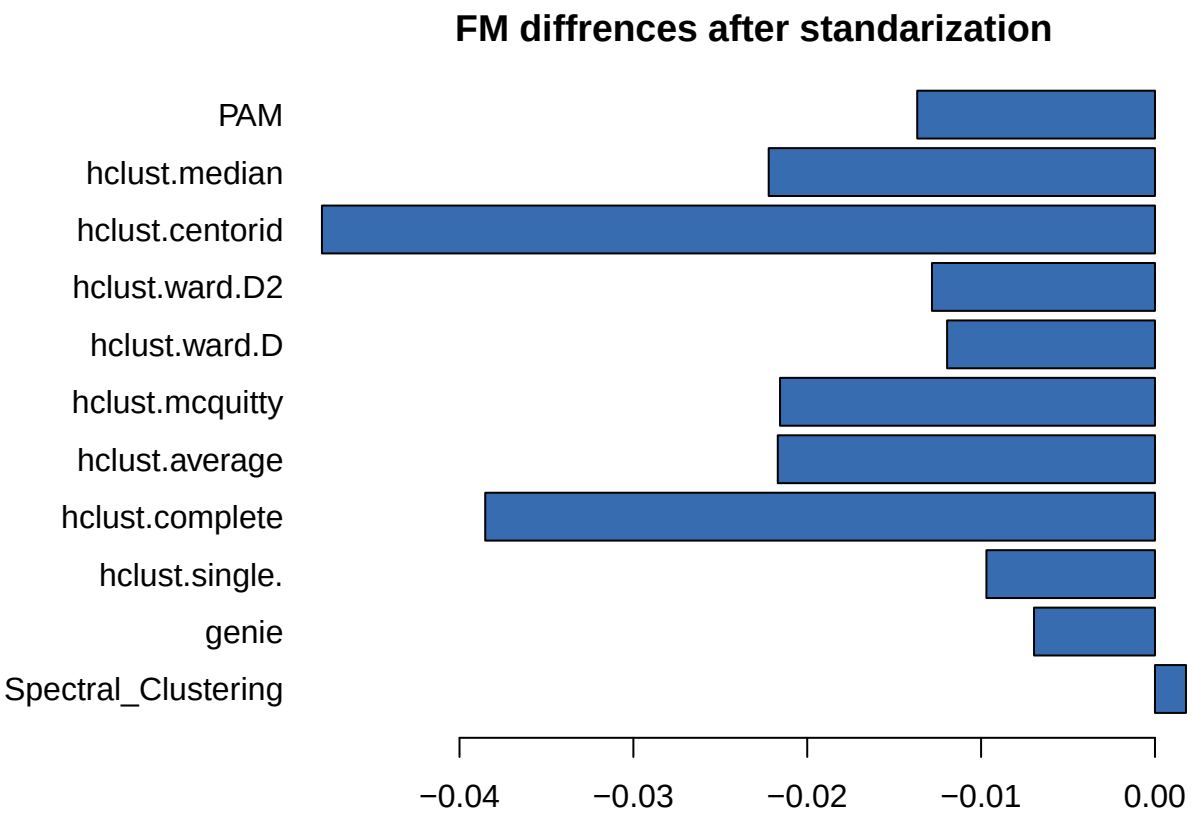
Teraz sprawdźmy, którym algorytmom standaryzacja pomogła, a których trafność została zmiejszona.



Teraz zróbmy to samo dla indeksów FM


FM− standatyzowane
PAM
hclust.median
hclust.centorid
hclust.ward.D2
hclust.ward.D
hclust.mcquitty
hclust.average
hclust.complete
hclust.single.
genie
Spectral_Clustering
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0
average FM standarized index


genie
PAM
hclust.ward.D
hclust.average
hclust.ward.D2
hclust.single.
hclust.centorid
hclust.complete
Spectral_Clustering
hclust.mcquitty
hclust.median
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8



Znów po standaryzacji, tylko wynik naszego algorytmu spektralnego się powiększa.

4. Szybkość algorytmów

```
## Unit: milliseconds
##         expr         min          lq        mean      median          uq
##     spectral 1251.495917 1270.810189 1307.108044 1287.069688 1291.659640
##        Genie    4.955170    5.019373    6.252114    5.218060    6.248680
##    hc.single    3.948986    3.983609    5.173807    4.374989    5.075345
##  hc.complete    6.617234    7.167717    7.974499    7.556868    8.770310
##       hc.avg    5.376052    5.534483    6.223088    6.062127    6.651135
##  hc.mcquitty    5.427354    5.952509    6.272026    6.020204    6.631422
##    hc.ward.D    6.118772    6.137330   11.550650    6.853050    9.371406
##   hc.ward.D2    5.743666    5.770003    6.520549    6.278619    6.690957
##  hc.centroid    5.027972    5.517624    9.826811    5.815671    6.704806
##    hc.median    5.112033    5.523627    6.254765    5.816654    6.878958
##          PAM   36.197702   36.819502   42.942119   37.800776   39.950931
##          max neval
##  1520.228445    10
##    12.544300    10
##    10.500101    10
##    10.591481    10
##     7.414000    10
##     7.763814    10
##    49.054103    10
##     9.424746    10
##    43.465056    10
```

```
##     8.541413    10
##    84.002850    10
```

5. Wnioski

Mój algorytm w porównaniu do innych algorytmów jest jednym z gorszych i wolniejszych. Prawdopodobnie mogłbym bardziej poeksperymentować z funkcją k-średnich, sprawdzić jak zmiana jej parametrów wpływa na wydajność. Jako jedynemu algorytmowi z przetestowanych standaryzacja zmiennych pomaga w widocznym stopniu, nawet o 0,06 w przypadku indeksów AR. Niekwestionowanym liderem jest jednak algorytm Genie, który zawsze pod względem wydajności przewyższa konkurencję, lecz co ciekawe standaryzacja zmiennych wpływa negatywnie na jego "celność".