Részvényelemzés klaszteranalízissel

Diplomamunka

Írta: Dzsubák Edina

Alkalmazott matematikus szak

Külső témavezető:

Sebestyén Géza, egyetemi adjunktus
Befektetések és Vállalati Pénzügy Tanszék
Budapesti Corvinus Egyetem, Gazdálkodástudományi Kar

Belső konzulens:

Pröhle Tamás, egyetemi tanársegéd
Valószínűségelméleti és Statisztika Tanszék
Eötvös Loránd Tudományegyetem, Természettudományi Kar

Eötvös Loránd Tudományegyetem
Természettudományi Kar
2010

Tartalomjegyzék

Köszönetnyilvánítás

Ezúton szeretném megköszönni Sebestyén Gézának az érdekes témát és a segítségét. Köszönet illeti továbbá Pröhle Tamást, aki nagyon sok hasznos tanáccsal látott el mind az R program használatában, mind a téma mélyebb megértésében.

Bevezetés

Az értékpapírok piacán a befektetők számára a potrfóliójuk kialakításához nélkülözhetetlen a részvények valalmilyen módszerekkel történő elemzése. Dolgozatomban klaszteranalízis segítségével vizsgálom a Budapesti Értéktőzsdén az elmúlt több mint tizenkét évben forgalomban lévő részvényeket.

A klaszterek segítségével láthatjuk, hogy mely részvények viselkednek hasonlóan, melyek különbözően.

A dolgozat felépítése a következő.

Az 1. Fejezetben ismertetem a klaszteranalízis elméletét. Csoportosítom a klaszterező eljárásokat. Bemutatom, azokat a klaszterező algoritmusokat, amelyeket később a részvényelemzés során használok, valamint a stabilitásvizsgálatnál használt hasonlóságmértéket. A fejezetet néhány alkalmazás felsorolásával zárom.

A 2. Fejezet egy közgazdaságtani alkalmazása a klaszeranalízisnek. Adatbázist készítek a részvények napi logaritmikus hozamából. A részvények egyéni vizsgálatához az idősorokat matematikai és pénzügyi mutatókkal jellemzem. A részvények együttes mozgását a keresztkorrelációkkal reprezentálom.

A 3. Fejezetben bemutatom a különböző klaszterező eljárások eredményeit.

A 4. Fejezetben ellenőrzöm, hogy a 3. Fejezetben lévő eredmények érvényesek-e.

Az 5. Fejezetben végül összegzem a munkámat.

1. fejezet

Klaszteranalízis

A klaszteranalízis olyan többváltozós elemzési technika, amely csoportosítással térképezi fel a megfigyelési eszközök között az azonosságokat és a különbözőségeket. A megfigyelési egységekhez rendelt változók jelentik azokat az eredeti dimenziókat, amelyek mentén a megfigyelteket csoportosítani kívánjuk, oly módon, hogy az egy csoportba tartozók minden változó mentén közel legyenek egymáshoz, és mindegyik más csoporttól, klasztertől távol essenek. A definícióból következik, hogy a klaszteranalízis kulcsfogalma a távolság. Az egyik legnehezebb probléma a klaszterelemzésnél a klaszterek számának meghatározása.

Az adatok klaszterezésének három főbb célja:

- Tömörítés: Eljárás az adatok rendszerezésére és összesítésére a klaszterprototípusokon keresztül.
- Természetes klasszifikáció: A hasonlóság mértékének azonosítása az osztályok és szervezetek között.
- Az adatok mögött rejlő struktúra feltérképezése.

A klaszterező eljárásoknak rengeteg alkalmazása van, és ezernél is több algoritmust publikáltak.

A klaszterező algoritmusokat két csoportba soroljuk: hierarchikus és partícionáló.

1.1. Hierarchikus Klaszterezés

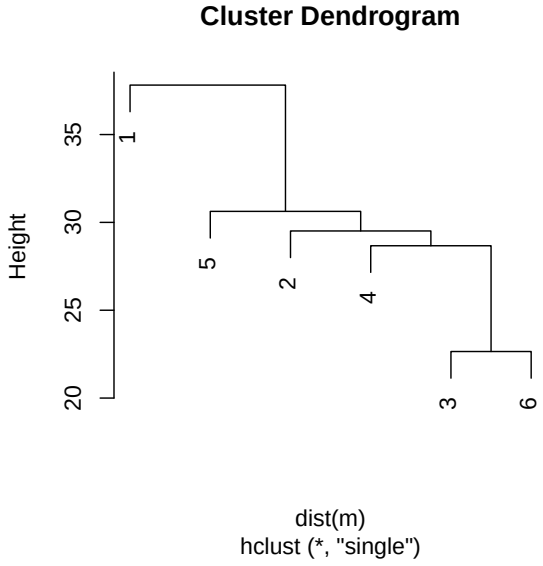
A hierarchikus klaszerezésnek is két fajtája van: az összevonó (agglomerative) és a felosztó (divisive).

Az összevonó módszer azon az elgondoláson alapul, hogy az első lépésben valamennyi klaszterezésre váró egyedet külön-külön egyszemélyes klaszterekben képzelünk el. Első lépésben tehát annyi klaszterünk van, ahány elemű a mintánk. A

második lépésben abból a két elemből, amely a legközelebb van egymáshoz, az eljárás közös klasztert készít. A harmadik lépésben két dolog történhet. Első esetben talál az eljárás egy olyan elemet, amely közel van a kételemű klaszterhez, és ekkor a kételemű klasztert háromeleműve bővíti. A második esetben talál két, egymáshoz közel eső elemet, és ebből egy új kételemű klasztert hoz létre. Az építkezés mindaddig folytatódik, míg valamennyi elemünk egyetlen klaszterben tömörül.

A felosztó módszer az összevonó módszerrel ellentétesen működik. Első lépében tehát egy klaszterünk van. A második lépésben két részre osztjuk a minden elemet tartalmazó klaszterünket. A harmadik lépésnél kiválasztjuk, hogy a két klaszter közül melyiket osszuk valamilyen módon két klaszterré. A felosztást addig folytatjuk, amíg végül minden klaszter egyetlen elemből áll. Az eljárás minden egyes lépésénél el kell döntenünk, hogy melyik klasztert akarjuk felosztani és hogyan.

A hierarchikus klaszterezést grafikusan is ábrázolható dendrogram vagy egymásba ágyazott halmazok segítségével. A dendrogram nem más, mint egy gráfelméleti fa, ahol a levelek kivételével minden pont (klaszter) a fában a gyerekeinek az egyesítése (szubklaszter), a fa gyökere pedig az összes pontot tartalmazó klaszter. Tehát a dendrogram segítségével jól prezentálható a csoport-alcsoport kapcsolat, és hogy milyen sorrendben vontuk össze illetve osztottuk fel a klasztereket. Ha elvágjuk a fát egy bizonyos magasságban, akkor azon az adott ponton megpróbálhatjuk értelmezni a klaszterezés eredményét. A 1.1 ábra példa egy dendrogramra, amely hat pont hierarchikus klaszterezését illusztrálja.



1.1. ábra. Hat pont hierarchikus klaszterezésének ábrázolása dendrogrammal

Az összevonó módszer algoritmusait ismertetem részletesen a továbbiakban [4] [12] könyvek alapján. A felosztó módszer algoritmusai ehhez hasonlóan dolgozhatóak ki.

1.1.1. Összevonó módszer

Az összevonó módszerek működése nagyon hasonló. A folyamat lépései az alábbiak szerint írható le általánosan.

1. A távolságmátrix kiszámolása, ha szükséges.
2. A két legközelebbi klaszter egyesítése.
3. A távolságmátrix újraszámolása.

A 2. és 3. lépést addig ismétli, amíg a végén már csak egyetlen, minden elemet tartalmazó klaszter marad.

Tár- és időigény

Vizsgáljuk meg az alap algoritmus tárigényét. Tegyük fel, hogy a távolságmátrix szimmetrikus, ekkor $\frac{1}{2}m^2$ tár szükséges a mátrix számításához, ahol m a megfigyelt elemek száma. A keletkező klaszterek nyomonkövetéséhez $m-1$ tár szükséges. Ezért az algoritmus tárolási bonyolultsága $O(m^2)$.

Most nézzük meg az algoritmus bonyolultságát a futásidő vonatkozásában. A távolságmátrix kiszámításának időigénye $O(m^2)$ az algoritmus 1. lépésében. A 2. és 3. lépés $m-1$-szer iterálja, hiszen kezdetben m klaszterünk van, és minden iteráció során két klasztert egyesítünk. Az $i-edik$ iteráció során a 2. lépésben $O((m-i+1)^2)$ idő szükséges a két legközelebbi klaszter megtalálásához. A 3. lépés $O(m-i+1)$ időt igényel a távolságmátrix módosításához. Így az algoritmus időigénye $O(m^3)$.

Az algoritmus idő- és tárigénye erősen korlátozza a klaszterezni kívánt adatok méretét. Tehát nagy adatbázison nem ajánlatos összevonó módszert alkalmazni.

Módszerek

A módszerek lényegében abban különböznek egymástól, hogy hogyan definiáljuk egy elem és egy klaszter, illetve két klaszter távolságát. Az alábbiakban példa segítségével illusztrálom az egyszerű és teljes láncmódszer működését, valamint röviden bemutatom a többi összevonó hierarchikus klaszterező eljárást.

Egyszerű láncmódszer

Egyszerű láncmódszer Nearest Neighbour vagy Single Linkage Method néven szerepel az angol nyelvű szakirodalomban. Ez az egyik legegyszerűbb eljárás. Egy elem és egy klaszter távolságán az adott elem és a hozzá legközelebb eső klaszterelem távolságát érti. Két klaszter távolsága pedig nem más, mint az egymáshoz legközelebb eső két, külön klaszterbe tartozó elem távolsága. Ezek közül a távolságok közül választja ki a minimálisat, és ennek megfelelően történik a soron következő összevonás. Tegyük fel, hogy öt egyedet szeretnénk osztályozni, jelölje D_1 az egyedek közötti távolságokat tartalmazó mátrixot.

$$D_1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 2 & 6 & 10 & 9 \\ 2 & 0 & 5 & 9 & 8 \\ 6 & 5 & 0 & 4 & 5 \\ 10 & 9 & 4 & 0 & 3 \\ 9 & 8 & 5 & 3 & 0 \end{bmatrix} \end{array}$$

A mátrix $i-edik$ sorában $j-edik$ oszlopában lévő elem adja meg az i és j egyed közötti d_{ij} távolságot.

Az első lépésben az 1 és 2 egyedek egysítésével készítünk új klasztert, hiszen a d_{12} a legkisebb eleme a D_1 mátrixnak. Jelölje a keletkezett klasztert (12). Az (12) klaszter és a 3, 4 és 5 egyed közötti távolságok a D_1 mátrix segítségével az alábbiak:

$$d_{(12)3} = \min\{d_{13}, d_{23}\} = d_{23} = 5$$

$$d_{(12)4} = \min\{d_{14}, d_{24}\} = d_{24} = 9$$

$$d_{(12)5} = \min\{d_{15}, d_{25}\} = d_{25} = 8$$

Most már felírható D_2 az új távolságmátrix.

$$D_2 = \begin{array}{c} \\ (12) \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{cccc} (12) & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 5 & 9 & 8 \\ 5 & 0 & 4 & 5 \\ 9 & 4 & 0 & 3 \\ 8 & 5 & 3 & 0 \end{bmatrix} \end{array}$$

A D_2 mártix legkisebb eleme a $d_{(45)}$, tehát a 4 és 5 egyedek egyesítésével készítünk új klasztert, amit (45) jelöl.

Ekkor a távolságok az alábbiak szerint számolhatóak:

$$d_{(12)3} = 5$$

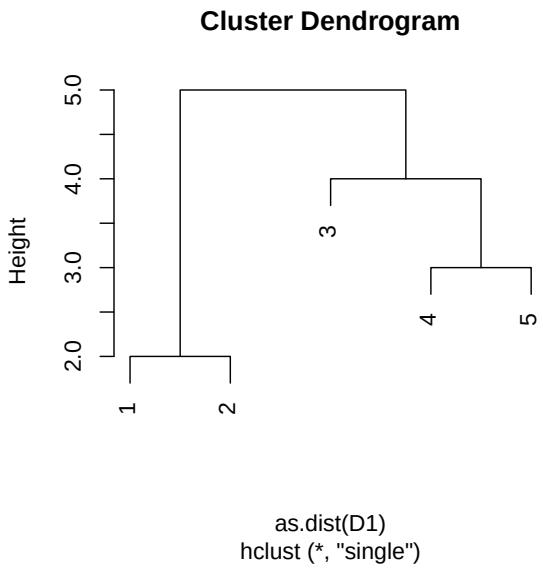
$$d_{(12)(45)} = \min\{d_{14}, d_{15}, d_{24}, d_{25}\} = d_{25} = 8$$

$$d_{(45)3} = \min\{d_{34}, d_{35}\} = d_{34} = 4$$

Ezeket a távolságokat mátrixba rendezve kapjuk a D_3 mátrixot.

$$D_3 = \begin{array}{c} \\ (12) \\ 3 \\ (45) \end{array} \begin{array}{c} \begin{array}{ccc} (12) & 3 & (45) \end{array} \\ \begin{bmatrix} 0 & 5 & 8 \\ 5 & 0 & 4 \\ 8 & 4 & 0 \end{bmatrix} \end{array}$$

A D_3 mátrix legkisebb eleme a $d_{(45)3}$, tehát a 3 egyedet a (45) klaszterrel egyesítem, amit (345) jelöl. Az utolsó lépésben egyesítem a (12) és a (345) klasztereket. Az eredményt a 1.2 ábrában lévő dendrogrammal reprezentáljuk.



1.2. ábra. Egyszerű láncmódszer

Teljes láncmódszer

A teljes láncmódszer Furthest Neighbour vagy Complete Linkage Method néven ismert az angol nyelvű szakirodalomban. Az egyszerű láncmódszertől abban különbözik, hogy egy elem és egy klaszter távolságán az adott elem és a tőle legtávolabb eső klaszterelem távolságát érti. Két klaszter távolsága pedig az egymástól legtávolabb

eső két, külön klaszterbe tartozó elem távolsága. Az összevonás ennél a módszernél úgy történik, hogy ezen legnagyobb távolságok közül választja ki a minimálisat.

Az egyszerű láncmódszernél látott példán keresztül nézzük meg a teljes láncmódszer működését. A D_1 mátrixból könnyen kiolvashatjuk, hogy az első lépésben az 1 és a 2 egyedeket vonjuk össze egy klaszterré. Ekkor az (12) klaszter és a 3, 4 és 5 egyedek közötti távolságok az alábbiak:

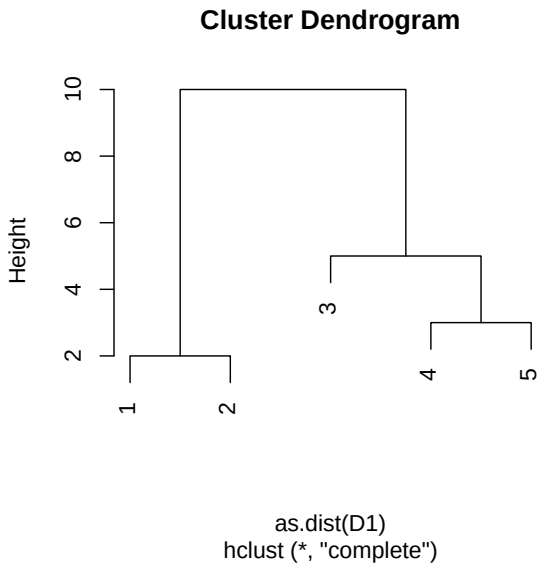
$$d_{(12)3} = \max\{d_{13}, d_{23}\} = d_{13} = 6$$

$$d_{(12)4} = \max\{d_{14}, d_{24}\} = d_{14} = 10$$

$$d_{(12)5} = \max\{d_{15}, d_{25}\} = d_{15} = 9$$

A végeredmény az 1.3 ábrában látható dendrogrammal illusztrálható, ami nagyon hasonló az egyszerű láncmódszernél kapott eredményhez.

Nem minden esetben hasonló a két különböző módszerrel kapott dendrogram.



1.3. ábra. Teljes láncmódszer

Mindkét módszernél a szélsőségek erősen befolyásolják a klaszterstruktúrát. Ilyen esetben szerencsésebb olyan módszereket választani, ahol a klaszterstruktúrát nem a véletlentől erősen befolyásolt két szélső elem távolságával, hanem a klaszter egészét megragadó mutató segítségével próbáljuk kialakítani.

Átlagos láncmódszer

Az angol nyelvű irodalomban Group Average Method néven ismert eljárás egy elem és egy klaszter távolságát az adott elem és az összes klaszterelem közötti távolságok átlagaként definiálja. Két klaszter távolsága nem más, mint a két klaszterből

az összes lehetséges módon kiválasztott elempárok távolságainak átlaga. A minimális távolságú klasztereket vonja össze minden egyes lépésben.

Ez az eljárás köztes megoldás az egyszerű és a teljes láncmódszer között.

Centroid-módszer

A centroid módszernél a klaszterek távolságát a klaszterközéppontok közötti távolság definiálja. A klaszterközéppont a klaszterbe tartozó összes elem súlypontja. Az összevonás kritériuma, hogy két klaszter középpontja közötti távolság minimális legyen az összes többi lehetséges klaszterösszevonáshoz képest.

Ez a módszer [12] szerint abban különbözik a többi összevonó hierarchikus eljárástól, hogy az összevont klaszterek távolsága nem feltétlenül monoton vagy szigorúan monton növő, ahogy az egyszemélyes klaszterektől eljutunk az összes elemet tartalmazó klaszterig. Előfordulhat, hogy az adott lépesben összevont klaszterek hasonlóbbak (kisebb köztük a távolság), mint két korábban egyesített klaszter távolsága.

Medián-módszer

Az centroid-módszer hátránya, hogy különböző méretű klaszterek egyesítésénél az új klaszterközéppont sokkal közelebb lesz a nagyobb klaszterhez, konkrétan abba a klaszterbe esik, ez a kisebb klaszter tulajdonképpeni eltűnéséhez vezethet.

A medián-módszernél úgy tekintünk az egyesíteni kívánt klaszterekre, mintha azonos méretűek lennének, így az összevonásuk során keletkezett új klaszter helyzete mindig a két egysített klaszter között lesz. Ráadásul, ha (i)-vel és (j)-vel jelöljük a két egyesített klaszter eredeti középpontját, valamint (h)-val egy harmadik klaszter középpontját, akkor az új klaszter távolsága a harmadik klasztertől az (i), (j) és (h) pontok által határolt háromszög súlypontja mentén fekszik. Innen származik az elnevezés, ugyanis a súlypont angolul median.

Ward-féle eljárás

A Ward-féle eljárás hasonlít az átlagos láncmódszerre, ha a távolságot két pont közötti távolság négyzetével definiáljuk.

A Ward-féle eljárás a klasztereken belüli távolságnégyzetek összegének minimalizálására törekszik. Azt a két klasztert egyesíti, amelyek a legkisebb négyzetes hibanövekedést okozzák.

Vegyük észre, hogy ez az eljárás a K-közép módszer hierarchikus megfelelője, ugyanazt a célfüggvényt használja.

1.2. Particionáló Klaszterezés

A particionáló algoritmusok alapgondolata, hogy a megfelelő klaszterezést a pillanatnyi eredményként kapott klaszterezés folyamatos pontosításával iterálva érjük el. Az algoritmus akkor ér véget, ha az iterációs lépés során nem (vagy csak alig) változik a partíció, illetve a klasztereket reprezentáló középpontjai.

1.2.1. K-közép módszer

A legnépszerűbb és legegyszerűbb klaszterező algoritmust, a K-közép módszert 1955-ben publikálták először. Ez az algoritmus jól alkalmazható nagy adathalmazokon.

A particionáló módszereknél ahelyett, hogy az adathalmaz elemszámával megegyező számú egyelemű klaszterek összeépítésével jutnánk el az általunk optimálisnak vélt klaszterstruktúrához, előzetes várakozásokra támaszkodva, vagy szerencsétlenebb esetben találomra kell eldöntenünk, hogy hány klaszterbe kívánjuk tömöríteni az elemeket.

Ha eldöntöttük, hogy hány klasztert kívánunk létrehozni, az eljárás minden klaszterhez egy-egy középpontot rendel. Megkeresi azokat az elemeket, amelyek az adott középpontokhoz a legközelebb vannak, és szükség esetén átsorolja őket a megfelelő klaszterbe. Ezeket az új csoportokat tekinti egy-egy klaszternek, és kiszámolja a klaszterközéppontokat. Ez a folyamat az iterálási folyamat. Addig iterálunk, amíg a klaszterközéppontok már nem változnak. A klaszterközéppont a klaszterbe tartozó elemek súlypontja. A célfüggvény a négyzetes hiba összege.

Előnyei:

- O(tkn) futási idő, ahol n a pontok száma, k az osztályok száma és t az iterációk száma.
- Egyszerű implementáció.

Hátrányai:

- Csak akkor alkalmazható, ha tudunk középpontokat számolni.
- Gyakran lokális optimumban áll meg az algoritmus.
- Az osztályok k számát előre meg kell adni.
- Nem kezeli jól a zajos adatokat és a magányos pontokat.

1.3. A klaszterek stabilitásának vizsgálata

Az ellenőrzés nagyon fontos a klaszteranalízisnél, ugyanis a klaszterező eljárások akkor is készítenek egy csoportosítást, ha meglehetősen homogén az adathalmaz. A legtöbb módszer feltételez egy bizonyos modellt vagy klaszterek egy prototípusát, és ez lehet, hogy érvényes az adatok bizonyos részére, de nem az egészére. A klaszterek létrejötte nem biztosíték a valóságos és értelmes csoportok megtalálására.

A stabilitás vizsgálata nagyon fontos a klaszterek érvényességének ellenőrzése szempontjából. A stabilitás azt jelenti, hogy az értelmes és érvényes klaszterek nem tűnnek el olyan könnyen, ha kissé változtatunk az adathalmazon. A stabilitás erősen függ az adathalmaztól, különösen attól, hogy mennyire különülnek el a különböző klaszterek egymástól, és mennyire homogének az egyes klaszterek.

1.3.1. Az adathalmaz torzításának lehetőségei

A klaszterek érzékenységének vizsgálatához változtatunk kissé az adathalmazon, majd megnézzük, hogy az így kapott új klaszter mennyire hasonlít az eredetihez. A továbbiakban ismertetek néhány adathalmazt torzító módszert.

Teljes adathalmaz helyett részhalmaz vétele

Az eljáráshoz válasszunk egy $m < n$ számot, ami jelölje a részhalmaz elemszámát. Ha m túl nagy, akkor az új adathalmaz nem lesz eléggé torz a stabilitásvizsgálathoz. Ha m túl kicsi, akkor a klaszterezés eredménye sokkal rosszabb lesz, mint amit az eredeti adathalmazon kaptunk. Hennig [5]-ben $m = [n/2]$-vel dolgozik, ahol $'[x]'$ az x szám egészrészet jelöli.

Elemek zajjal való helyettesítése

Statisztikai módszerek instabilitásának megmutatásánál gyakran használt módszer az elemek zajjal való helyettesítése. A torzítás során kiválasztunk m elemet az adathalmazból. A kiválasztott elemeket olyan elemekkel helyettesítünk, amelyek egy adott eloszlású zajból származnak. Tehát szükségünk van egy $m < n$ számra, amely a zajjal helyettesíteni kívánt elemek számát jelöli, és meg kell adnunk a zaj eloszlását. A zaj eloszlását nehéz megválasztani.

Jittering

A "jittering" azt jelenti, hogy az adathalmaz minden egyes pontjához hozzáadunk egy kicsi zajt. Ezzel azt próbáljuk reprezentáljuk, hogy minden adathoz tartozik

valamilyen mérési hiba. Szükség van a mérési hiba eloszlására, ami általában normál eloszlású.

1.3.2. Partíciók összehasonlítása

Ahhoz, hogy megállapíthassuk az eredeti klaszter stabilitását az új klaszter vonatkozásában, szükségünk van valamilyen hasonlóságmértékre. A mérték tartalmazáson alapuljon, hogy alkalmazható legyen különböző klaszterező eljárásokra. A partíciók összehasonlítása egy elég tág témakör, külön irodalommal [8] rendelkezik. Nagyon sok mértek létezik, Denoeud, Garreta és Guénoche tanulmányukban [3] bemutatta és összehasonlította a legfontosabbak működését. A klaszteranalízisben a Hennig cikkeiben [5] [6] is alkalmazott Jaccard index és a Rand index használata a legelterjedtebb.

$\mathbf{X}_n$: Egy n elemű $\mathbf{X}_n = (x_1, x_2, ..., x_n)$ halmaz.

C : $C = (C_1, C_2, ..., C_s)$ az $\mathbf{X}_n$ egy partíciója, ahol $C_i \cap C_j = \emptyset$, ha $i \neq j \leq s$, továbbá $\cup_{j=1}^{s} C_j = \mathbf{X}_n$.

D : $D = (D_1, D_2, ..., D_k)$ az $\mathbf{X}_n$ egy másik partíciója, ahol $D_i \cap D_j = \emptyset$, ha $i \neq j \leq k$, továbbá $\cup_{j=1}^{k} D_j = \mathbf{X}_n$.

a : Azoknak az elempároknak a száma $\mathbf{X}_n$-ben, amelyek azonos halmazba tartoznak a C partíció szerint, és azonos halmazba tartoznak a D partíció szerint is.

b : Azoknak az elempároknak a száma $\mathbf{X}_n$-ben, amelyek azonos halmazba tartoznak a C partíció szerint, és különböző halmazba tartoznak a D partíció szerint.

c : Azoknak az elempároknak a száma $\mathbf{X}_n$-ben, amelyek különböző halmazba tartoznak a C partíció szerint, és azonos halmazba tartoznak a D partíció szerint.

d : Azoknak az elempároknak a száma $\mathbf{X}_n$-ben, amelyek különböző halmazba tartoznak a C partíció szerint, és különböző halmazba tartoznak a D partíció szerint is.

1.3.1. Definíció (Jaccard index).

$$J(C, D) = \frac{a}{a + b + c}$$

1.3.2. Definíció (Rand index).

$$R(C, D) = \frac{a + d}{a + b + c + d} = \frac{a + d}{\binom{n}{2}}$$

Az indexek értéke 0 és 1 közé esik. Ha az index értéke nagy, akkor nagy mértékben hasonlóak a partíciók, ha kicsi akkor kevésbé.

1.4. Alkalmazási Területek

Biológia

A biológiában a rendszertan az élőlények csoportosításával, kategorizálásával foglalkozó tudományág. Minden élőlény rendszerezhető ország, törzs, osztály, rend, család, nemzetség (növényeknél és gombáknál) vagy nem (állatoknál), faj kategóriák szerint. Ez egy hierarhikus osztályozó rendszer, így nem is olyan meglepő, hogy ezen a területen alkalmazták a klaszteranalízist a kezdetekben.

Manapság a biológusok az óriási mennyiségű genetikai információ elemzésénél használják ezt a módszert. Például gének olyan csoportjának meghatározására, amelyeknek hasonló szerepük van.

Társadalomtudomány

Gyakran alkalmazott módszer társadalmi struktúra modellek megalkotásánál, amikor emberek egy nagyobb csoportját kisebb közösségekbe sorolják.

A klaszterelemzés kiválóan használható a piackutatásban. A jelenlegi és a potenciális vásárlókról rendelkezésre álló információk alapján szegmentálható a piac, ami hasznos lehet további elemzések, termékfejlesztés, tesztpiacok kiválasztása és marketing tevékenység szempontjából.

A [11] konkrét társadalomtudományi példákat elemez SPSS segítségével.

Informatika

A World Wide Web több milliárd weboldalt tartalmaz, így oldalak ezreit kaphatjuk egy keresés eredményeként. A klaszterezés segítségével kisebb csoportokra oszthatjuk a keresendő szóra kapott találatokat. Például [12] szerint a "film" szó keresésénél a weboldalak filmajánlók, mozik, filmsztárok, kritikák kategóriába sorolhatóak. Minden kategória alkategóriákra osztható, ami egy hierarchikus rendszert eredményez, ezzel megkönnyítve a felhasználó számára a keresés szűkítését.

Orvostudomány

Egy betegségnek vagy egy kórnak számos változata lehet, és a klaszteranalízis segítségével felismerhetjük ezeket a típusokat. Például, ezzel a módszerrel határozták meg a depresszió különböző fajtáit. Valamint klaszterezést alkalmaznak a járványok, betegségek térbeli és időbeli terjedésének kiderítésére.

Meteorológia

A sok paramétertől függő meteorológiai mezők és folyamatok tanulmányozásának egyik lehetséges módja a klaszterezés. Magyarországon korábban a szélmezők összehasonlítása és osztályozása pauszpapírra rajzolt szélmezők egymásratételével, vizuális úton történt. Ez igen munkaigényes és objektivitásban is megkérdőjelezhető eljárás volt. 1981-ben Tutsek Endre [13] egy - elsősorban környezetvédelmi alkalmazásokra készült - gyorsabb, objektívebb és sokkal kisebb munkaigényű szélmező klaszterező eljárás dolgozott ki.

Közgazdaságtan

N. De Costa, J. Cunha és S. De Silva tanulmányukban [2] észak- és dél-amerikai értékpapírokat csoportosított hierachikus klaszteterező eljárással. Megmutatták, hogy azok a befektetők, akik klaszteranalízis segítségével keletkező részvénycsoportokat figyelembe véve választottak részvényeket portfóliójukba, profitáltak vagy kevesebbet veszítettek.

Dolgozatommal én is egy közgazdaságtani alkalmazást mutatok be. A magyar részvényeket klaszterezem egyéni viselkedésük és egymáshoz való viszonyuk alapján. A részvények ilyen módon történő elemzése és csoportosítása befektetőknek nyújt segítséget portfólió készítésnél.

2. fejezet

Részvényelemzés

Ahhoz, hogy elemezni tudjam a részvényeket, készítettem egy adatbázist. Majd egyéni viselkedésük és együttes mozgásuk szerint vizsgáltam a részényeket.

2.1. Adatbázis készítés

Az adatbázis elkészítését az adatok gyűjtésével kezdtem, majd ezeket az adatokat javítottam, formáltam és a megfelelő módon rendszereztem.

2.1.1. Adatok gyűjtése

Az adatbázishoz a Magyarországon kereskedett részvények napi záróárait gyűjtöttem össze. Az adatokat a Budapesti Értéktőzsde (BÉT) honlapjáról (www.bet.hu) töltöttem le.

Az 1997. május 12-étől kezdődő és 2010. április 07-éig tartó időszakot vettem figyelembe, ami 3225 kereskedési napot tartalmazott, ugyanis szombaton, vasárnap és ünnepnapokon nincs kereskedés a tőzsdén. Ebben az időintervallumban összesen 87 részvénnyel kereskedtek. Hozzávettem még az adatbázishoz a BUX részvényindex napi záróárait is. Így az adatbázis (88x3225)-ös méretű, amiben vannak hiányzó adatok is. A BUX részvényindex mellett még 16 részvényhez tartozik teljes idősor, ilyen a DANUBIUS, EGIS, FOTEX, MOL, OTP, TVK. A többi részvényhez tartozó idősor hiányos, ugyanis vannak olyan részvények, amelyekkel vagy csak 1997. május 12-e után kezdtek el kereskedni, vagy már 2010. április 07-e előtt kivonták a forgalomból vagy mindkettő igaz az adott részvényre. 1997 után kezdtek kereskedni például az AAA, BOOK, EXTERNET, FHB, TVNETWORK részvényekkel. A BCHEM, GLOBUS, IBUSZ, PICK részvényeket pedig már kivonták a forgalomból.

Az összegyűjtött adatokhoz tartozó idősorokat zX-szel jelöltem, melyeket egy EXCEL táblázatba rendeztem, majd előkészítettem őket az elemzéshez.

2.1.2. Adatok előkészítése a felhasználáshoz

Tehát rendelkezésre álltak 88 részvénynek több mint tizenkét éves időszakra vonatkozó napi záróárai.

Ahhoz, hogy a részvényekhez tartozó idősorokat statisztikai módszerekkel vizsgálhassuk, stacionárius idősorokra van szükség.

2.1.1. Definíció. *Egy $X(t)$ $t \in \mathrm{R}$ idősor erősen stacionárius, ha $(X_{t_1}, X_{t_2}, ..., X_{t_n}) \sim (X_{t_1+k}, X_{t_2+k}, ..., X_{t_n+k})$ $\forall$ $(t_1, t_2, ..., t_n)$-re és k-ra.*

2.1.2. Definíció. *Egy $X(t)$ $t \in \mathrm{R}$ idősor gyengén stacionárius, ha várható érkéke eltolás invariáns, azaz $E(X(t)) = E(X(t+k))$, és autokovariancia függvénye is eltolás invariáns, azaz $cov(X(t), X(s)) = cov(X(t+k), X(s+k))$ $\forall$ t-re, s-re és k-ra.*

A legtöbb esetben az eredeti idősorokon megfigyelhető valamilyen trend vagy szezonalitás, így nem stacionáriusak. Ezért kiszámoltam a napi hozamokat, majd vettem a hozamok logaritmusát, és ezek lett az új idősorok.

Tehát ha az eredeti idősor $zX(t)$, akkor az új idősor $nX(t) = \ln \frac{zX(t+1)}{zX(t)}$. Az új idősorok már függetlenek az időtől.

A továbbiakban a napi záróárak helyett a napi logaritmikus hozamokkal dolgoztam. Az új időssorokat nX-szel jelöltem, amelyet egy EXCEL táblázatba raktam, és ez lett az adatbázisom.

2.2. Részvények egyéni vizsgálata

A továbbiakban azt vizsgálom, hogyan viselkednek az adott részvények. Leíró statisztikákkal jellemzem az idősorokat, valamint modelleket illesztek az adatokra. Ezeket a statisztikai és pénzügyi mutatókat tekintem a változóknak, ami szerint klaszterezem a részvényeket.

Ahhoz, hogy ezeket a mutatókat kiszámoljam, az R nevű szoftvert használom, ami egy olyan programozási nyelv és környezet, amely különösen alkalmas statisztikai számítások és grafikai megjelenítési feladatok megvalósítására.

2.2.1. Leíró statisztikák

Ebben a részben bemutatom azokat a statisztikákat, amelyeket az idősorok leírására használok. A statisztikákat az R-ben történő programozásnál használt nevük szerint jelölöm:

mean : Számtani vagy aritmetikai középértéken n elemű minta átlagát, azaz az elemek összegének n-ed részét értjük.

$$A(X_1, X_2, ..., X_n) = \frac{X_1 + X_2 + ... + X_n}{n}$$

var : Az X valószínűségi változó szórásnégyzetét (variancia) a

$$D^2(X) = E(X - E(X))^2$$

képlet adja meg, ha $\exists E(X)$ az X várhatóértéke.

Q50 : A medián,

- ha páratlan elemszámú a minta, akkor a rendezett minta középső eleme.
- ha páros elemszámú a minta, akkor a rendezett minta két középső elemének számtani közepe.

absmean : Az elemek abszolútértékének számtani közepe.

skewness : Az $E(X)$ várható értékű X valószínűségi változó ferdesége azt mutatja meg, hogy mennyire nem szimmetrikus a valószínűségi változó eloszlása.

$$\frac{E[(X - E(X))^3]}{(E[(X - E(X)^2)])^{\frac{3}{2}}}$$

kurtosis : Az $E(X)$ várható értékű X valószínűségi változó lapultsága (csúcsossága) azt mutatja meg, hogy a valószínűségi változó sűrűségfüggvényének "csúcsossága" vagy "laposssága" hogyan viszonyul a normális eloszláséhoz.

$$\frac{E[(X - E(X))^4]}{(E[(X - E(X))^2])^2} - 3$$

mm.dist : A medián és a számtani közép távolságának a szóráshoz való viszonya.

$$\frac{Q50 - mean}{\sqrt{var}}$$

Q01 : 1. Percentilis az a kavntilis, amelyre 0.01 a valószínűsége annak, hogy a valószínűségi változó értéke nála kisebb.

Q99 : 99. Percentilis az a kvantilis, amelyre 0.99 a valószínűsége annak, hogy a valószínűségi változó értéke nála kisebb.

Q10 : 1. Decilis az a kvantilis, amelyre 0.1 a valószínűsége annak, hogy a valószínűségi változó értéke nála kisebb.

Q90 : 9. Decilis az a kvantilis, amelyre 0.9 a valószínűsége annak, hogy a valószínűségi változó értéke nála kisebb.

Q25 : Az alsó kvartilis a legkisebb és a medián között középen elhelyezkedő adat számértéke a rendezett mintában.

Q75 : A felső kvartilis a legkisebb és a medián között középen elhelyezkedő adat számértéke a rendezett mintában.

rrange : Terjedelem szóráshoz való viszonya.

$$\frac{max - min}{\sqrt{var}}$$

r.half : Interkvartilis terjedelem szóráshoz való viszonya.

$$\frac{Q75 - Q25}{\sqrt{var}}$$

iv.hossz : A lokális szélsőértékek közötti ívhosszak számtani közepe.

iv2hossz : A lokális szélsőértékek közötti ívhosszak négyzetének számtani közepe.

2.2.2. Modellillesztések

Ebben a részben bemutatom azokat a módszereket, amelyekkel valamilyen modellt illesztettem az idősorokra.

Lokális polinomiális regresszió

A LOWESS módszer célja, hogy lokálisan illesszen rövid egyenes szakaszokat az nX-ben lévő idősorokra a súlyozott legkisebb négyzetek módszerével. Az ilyen jellegű elemzés célja, hogy nagy számú változó összefüggéséről sok adat esetén megbízható, robusztus eredményt kapjunk.

Exponenciális simítás

Az exponenciális simítás lényege abban van, hogy egy adott időponthoz tartozó értéket úgy definiálunk, hogy abban benne foglaltatnak a múltbeli értékek is az időben visszafelé haladva egyre kisebb súllyal. A súly értéke 0 és 1 között lehet. Amennyiben 1-hez közeli súlyt választunk, akkor kis mértékben fogjuk kisimítani az idősorunkat, azaz nagy súlyt kap az aktuális érték, és kis súlyt kapnak a múltbeli értékek. Nulla közeli súly választása esetén pedig erős simítást hajtunk végre az idősoron, az ingadozásokat szinte teljesen kiszűrjük, és egy hullámzó görbét fogunk kapni. Ebben az esetben kis súlyt fog kapni az aktuális érték, és nagy súlyt fognak kapni a múltbeli értékek.

Kálmán szűrés

A Kalman szűrési módszer [9] dinamikus rendszerek állapotának becslésére alkalmas. A rendszert leíró paraméterek becsült értéke egyrészt az adott időpontban végzett mérés, másrészt a korábbi mérések alapján végzett előrejelzés együttes figyelembe vételével határozható meg.

ARMA

Adott egy $X(t)$ idősor, akkor az idősorra illesztett ARMA modell segítségével megérthejük, és talán előre is jelezhetjük az idősor jövőbeli értékeit.

Az ARMA(p, q) modell két részből áll, egy p-edrendű autoregresszív AR(p) modellből, és egy q-adrendű mozgó átlag MA(q) modellből.

Az AR(p) modell a következő módon írható le:

$$X(t) = c + \sum_{i=1}^{p} \varphi_i X(t-i) + \varepsilon(t),$$

ahol $\varphi_1, \varphi_2, ..., \varphi_p$ valós együtthatók, amelyek a modell paraméterei, c konstans és $\varepsilon(t)$ fehér zaj. Az egyszerűség kedvéért a c elhagyható.

A MA(q) modell a következő módon írható le:

$$X(t) = \mu + \varepsilon(t) + \sum_{i=1}^{q} \theta_i \varepsilon(t-i),$$

ahol $\theta_1, \theta_2, ..., \theta_q$ valós együtthatók, amelyek a modell paraméterei, μ az $X(t)$ várhatóértéke és $\varepsilon(t)$ fehér zaj. Gyakran felteszik, hogy $\mu = 0$.

Tehát az ARMA(p, q)-val jelölt p-edrendű és q-adrendű autoregresszív mozgóátlag folyamat az alábbiak szerint adható meg:

$$X(t) = \varepsilon(t) + \sum_{i=1}^{p} \varphi_i X(t-i) + \sum_{i=1}^{q} \theta_i \varepsilon(t-i),$$

ahol φ_i-k és θ_j-k valós együtthatók, a modell paraméterei és $\varepsilon(t)$ fehér zaj.

Az $\varepsilon(t)$ fehér zajról feltehetjük, hogy független, azonos eloszlású valószínűségi változó, amely 0 és σ^2 paraméterű normális eloszlású.

A Függelékben megtekinthető, hogy melyik ARMA modelleket használtam.

GARCH

A GARCH(p, q) modellt a következő képlet írja le:

$$X(t) = \sigma(t)\varepsilon(t),$$

ahol $\varepsilon(t)$ fehér zaj és $\sigma(t)$ a következő módon definiálható:

$$\sigma^2(t) = \alpha_0 + \sum_{i=1}^{q} \alpha_i X^2(t-i) + \sum_{i=1}^{p} \beta_i \sigma^2(t-i)$$

A GARCH(1, 1) modellt használtam.

Az idősorokat leíró statisztikákat és modellillesztésekkel kapott mutatókat egy S mátrixba rendeztem. Az S mátrix (39x88)-as méretű, ugyanis mind a 88 részvényhez 39 féle mutató tartozik.

2.3. Részvények együttes mozgása

Fontos, hogy egy portfólióban legyenek olyan részvények, amelyek ellentétesen, egymástól függetlenül vagy egymáshoz képest kevésbé összefüggően mozognak. Ez azért jó egy befektetőnek, mert ha az egyik részvényének az ára elkezd zuhanni, akkor a másik részvény kompenzálja valamelyest a veszteség mértékét. Mivel a magyar részvényekre ugyanúgy hatnak a gazdasági változások, nagyon nehéz olyan részvénypárt találni, amelyek hosszútávon ellentétesen mozognak.

A részvények együttes mozgásának leírásához a részvények idősorainak keresztkorrélációit használtam. Azért, hogy a késleltetett hatásokat is felfedezhessem, kiszámoltam azokat a keresztkorrelációkat is, amelyeknél egyik idősort 1, 2, 3, 4 illetve 5 nappal eltoltam.

2.3.1. Definíció. *Az X és Y valószínűséi változók korrelációja*

$$R(X,Y) = corr(X,Y) = \frac{cov(X,Y)}{D(X)D(Y)} = \frac{E[(X-E(X))(Y-E(Y))]}{D(X)D(Y)}$$

2.3.2. Definíció. *Keresztkorreláció két különböző idősor adatai közötti korreláció számítás.*

A korreláció jelzi két tetszőleges érték közötti lineáris kapcsolat nagyságát és irányát (avagy ezek egymáshoz való viszonyát). A korreláció -1 és +1 közé esik, és egyenlőség akkor és csak akkor áll fenn, ha a két változó lineáris kapcsolatban áll egymással.

Az nX-ben lévő részvények idősoraihoz tartozó keresztkorrelációkat a Függelékben látható módon számoltam ki. A 0, 1, 2, 3, 4 illetve 5 nappal eltolt keresztkorrelációkat rendre az $R0$, $R1$, $R2$, $R3$, $R4$, $R5$ adathalmaz tartalmazza. Ezeket az adathalmazokat összeadva kaptam meg az R adathalmazt, amivel végülis jellemeztem a részvények egymáshoz viszonyított mozgását.

3. fejezet

Klaszterelemzés

Bemutatom dendrogramokkal és táblázatokkal illusztrálva a részvények egyéni viselkedése és együttes mozgása szerinti klaszterstruktúrákat.

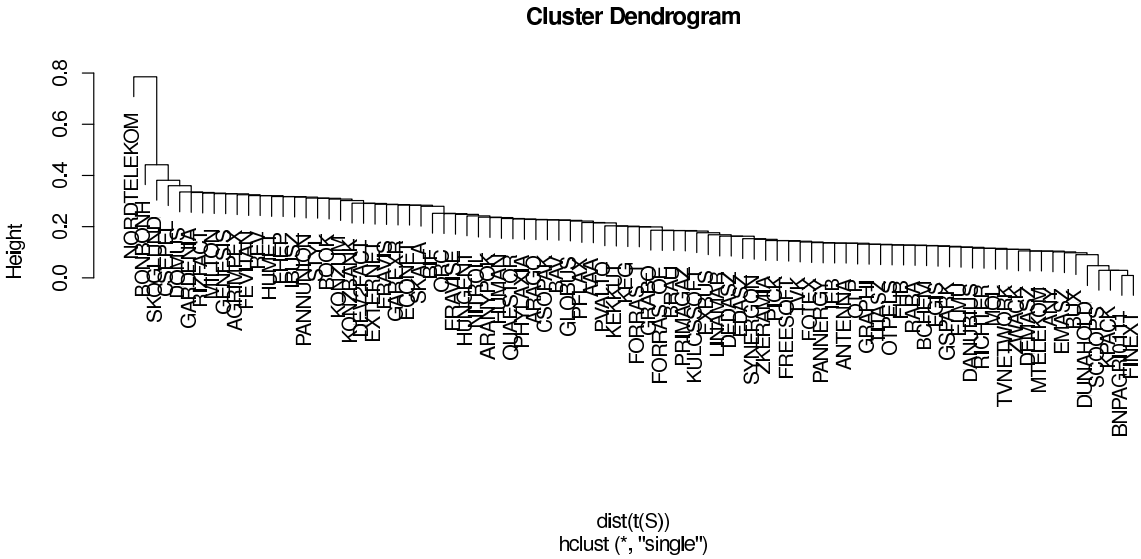
3.1. Egyéni viselkedés szerinti klaszterezés

Az alábbiakban a részvények egyéni viselkedését leíró S adathalmazon történő klaszterezés eredményeit illusztrálom. A k-közép módszerrel kapott partíciókat a 3.1 táblázat mutatja be. A 3.1 ábra az egyszerű láncmódszer, a 3.2 ábra a teljes láncmódszer, a 3.3 ábra az átlagos láncmódszer, a 3.4 ábra a centroid-módszer, a 3.5 ábra a medián-módszer, míg a 3.6 ábra a Ward-féle eljárás dendrogramját mutatja.

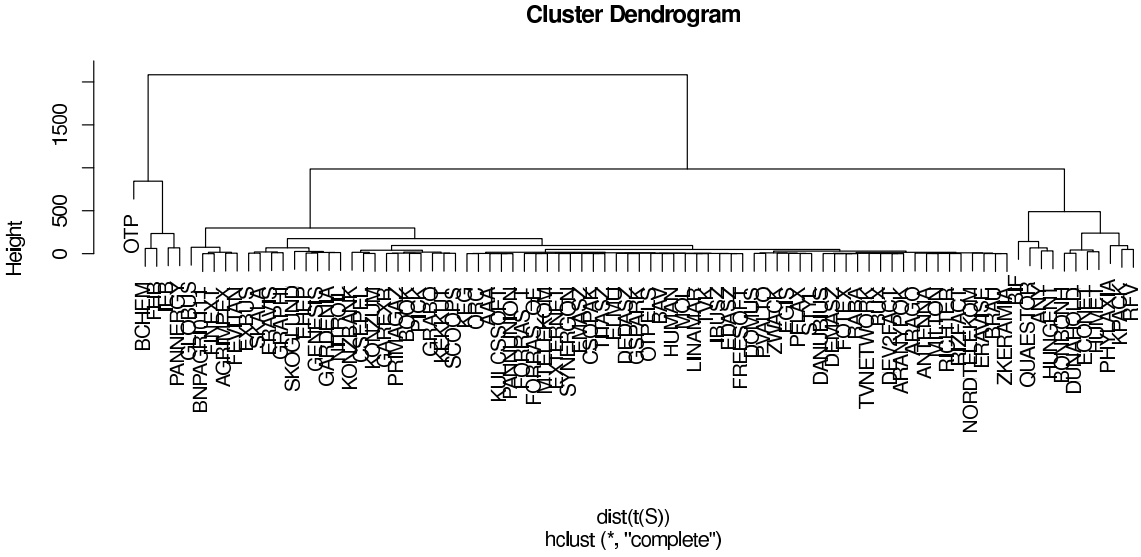
Remekül kiolvasható a 3.1 táblázatból, hogy az azonos iparághoz tartozó részvények azonos klaszterbe kerültek. Például a 7 klaszterbe került a DEDASZ, DEMASZ, EDASZ, ELMU, EMASZ, TITASZ, azaz az áramszolgáltatók. Az EGIS és a RICHTER a 4 klaszterbe tartozik, mindkettő gyógyszergyár.

Bux	2devfact	Aaa	Agrimpex	Antenna	Any
7	7	7	3	4	2
Arago	Aranypok	Bav	Bchem	BIF	Bnpagrio
4	7	7	2	2	3
Bonbonh	Book	Brau	Csepel	Csopak	Danubius
1	8	7	8	7	7
Dedasz	Demasz	Domus	Dunahold	Econet	Edasz
7	7	4	1	1	7
Egis	Ehep	Elmu	Emasz	Eravis	Eravise
4	3	7	7	8	7
Exbus	Externet	Fevitan	Fhb	Finext	Forrasoe
8	7	3	2	6	7
Forrast	Fotex	Freesoft	Gardenia	Garexr	Genesis
7	7	7	3	8	3
Globus	Grabo	Graphi	Gspark	Human	Humet
3	4	8	7	7	1
Hungent	Ibusz	Ieb	Keg	Kekkut	Konzbank
2	7	5	7	4	8
Konzum	Kpack	Kulcssoft	Linamar	Milton	Mol
8	1	7	7	4	7
Mtelekom	Nitroil	Nordtelekom	Orc	Otp	Otpels
7	3	4	7	5	7
Pannergy	Pannunion	Pflax	Phylaxia	Pick	Primagaz
5	7	4	1	8	8
Pvalto	Quaestor	Raba	Rfv	Richter	Rizfact
4	2	7	1	4	4
Scoops	Skala	Skoglund	Styl	Synergon	Titasz
4	8	3	4	7	7
Tvk	Tvnetwork	Zkeramia	Zwack		
7	7	7	4		

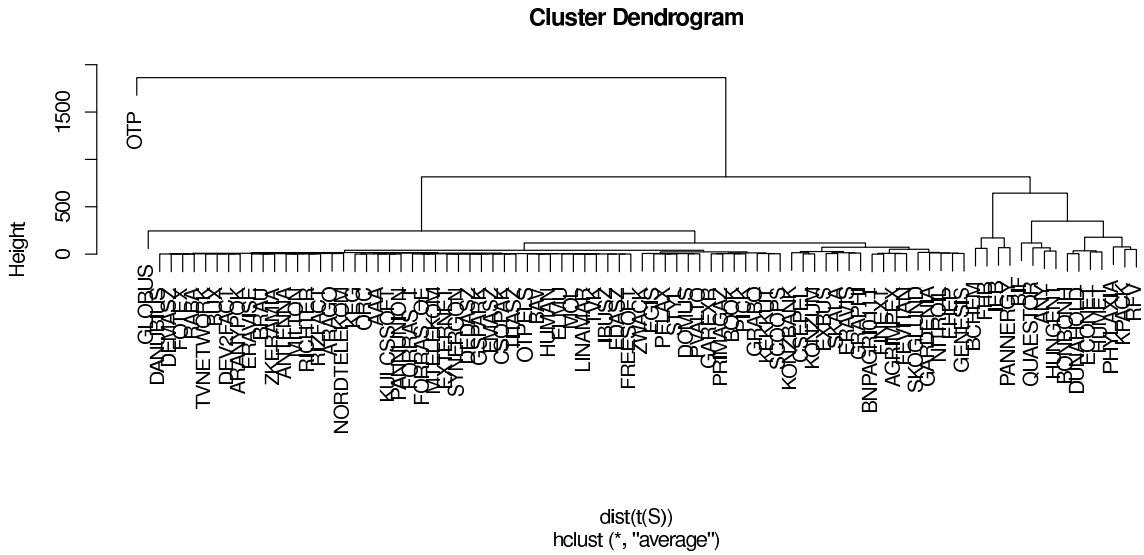
3.1. táblázat. K-közép módszer S-re 8 klaszterrel



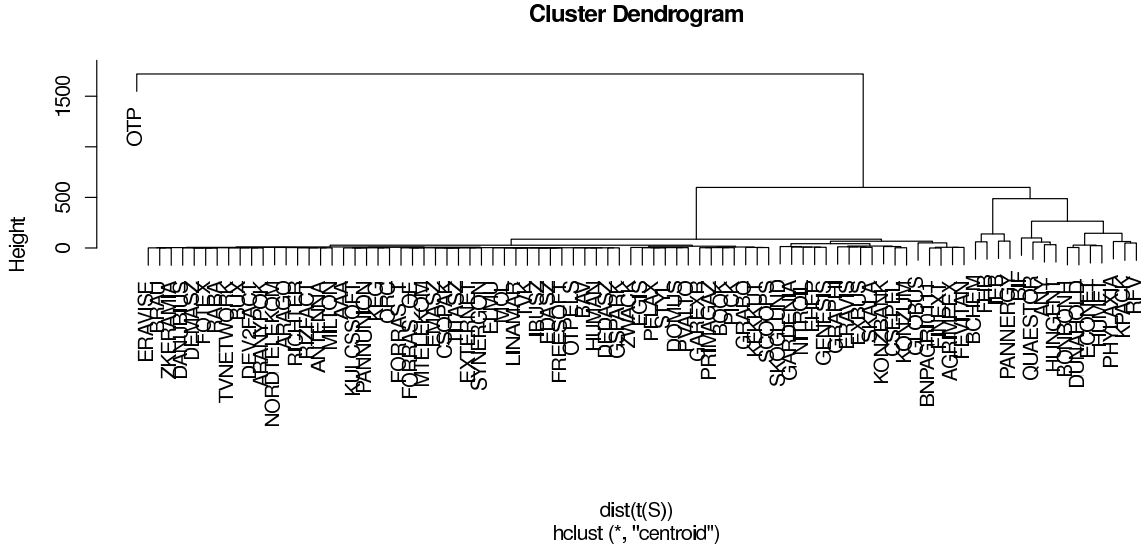
3.1. ábra. S klaszterezésének dendrogramja egyszerű láncmódszerrel



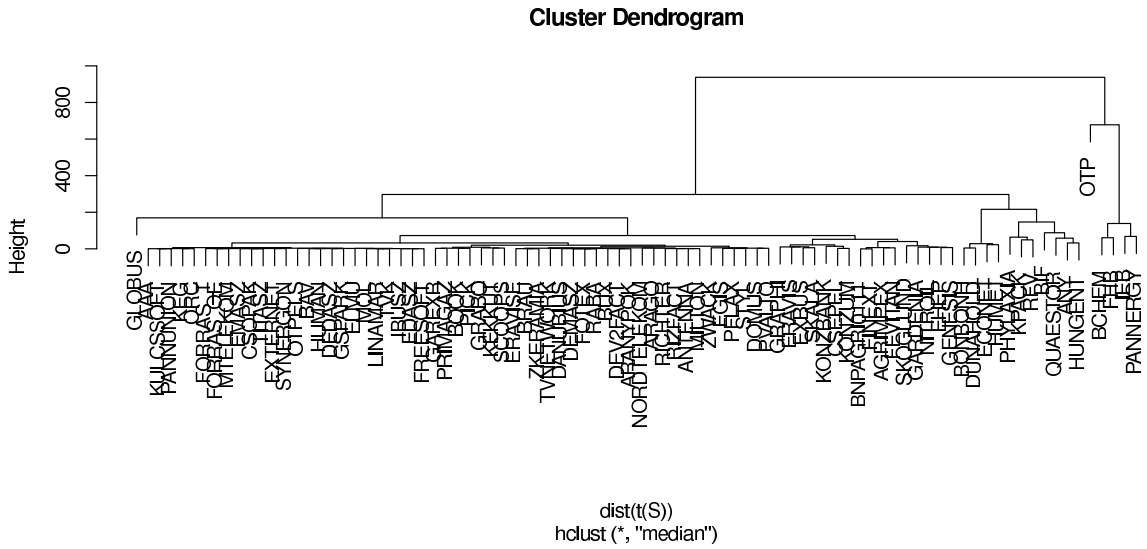
3.2. ábra. S klaszterezésének dendrogramja teljes láncmódszerrel



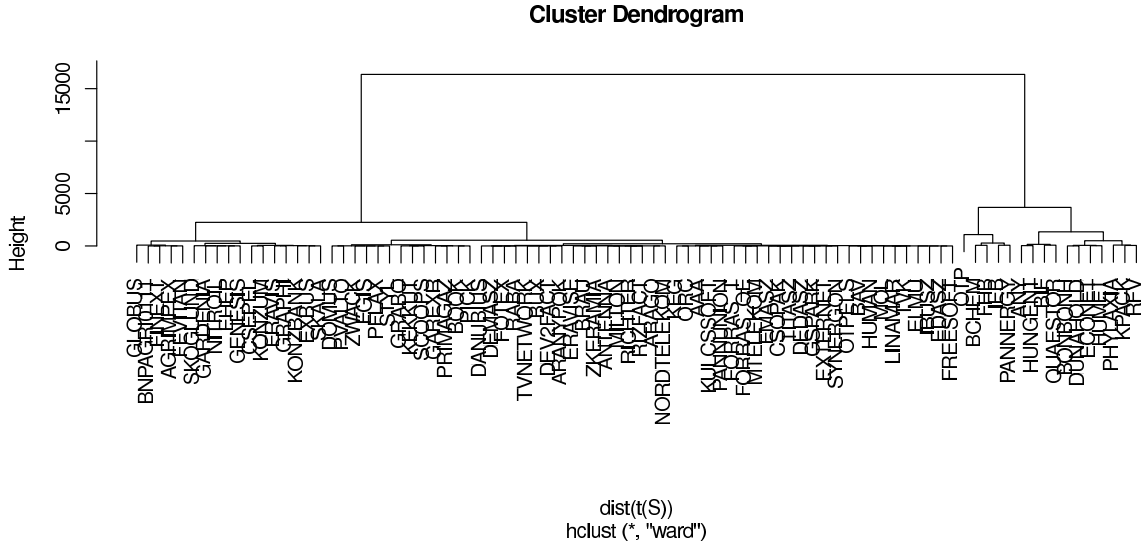
3.3. ábra. S klaszterezésének dendrogramja átlagos láncmódszerrel



3.4. ábra. S klaszterezésének dendrogramja centroid-módszerrel



3.5. ábra. S klaszterezésének dendrogramja medián-módszerrel

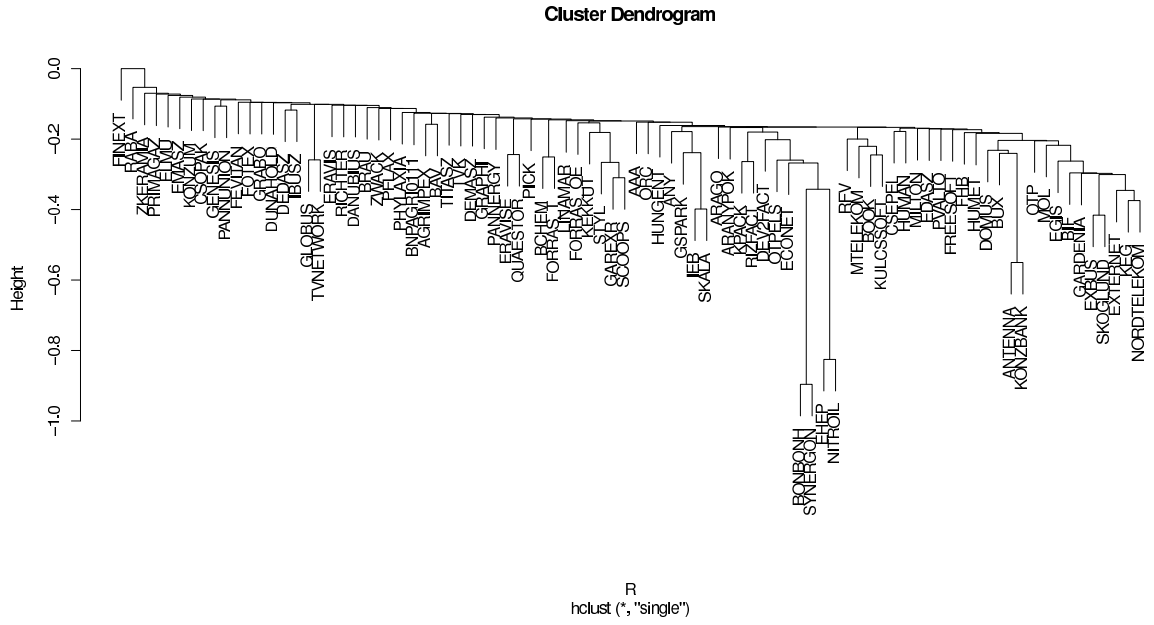


3.6. ábra. S klaszterezésének dendrogramja Ward-féle eljárással

3.2. Együttes mozgás szerinti klaszterezés

Az alábbiakban a részvények együttes mozgását leíró R adathalmazon történő klaszterezés eredményeit illusztrálom. A k-közép módszerrel kapott partíciókat a 3.2 táblázat mutatja be. A 3.7 ábra az egyszerű láncmódszer, a 3.8 ábra a teljes láncmódszer, a 3.9 ábra az átlagos láncmódszer, a 3.10 ábra a centroid-módszer, a 3.11 ábra a medián-módszer, míg a 3.12 ábra a Ward-féle eljárás dendrogramját mutatja.

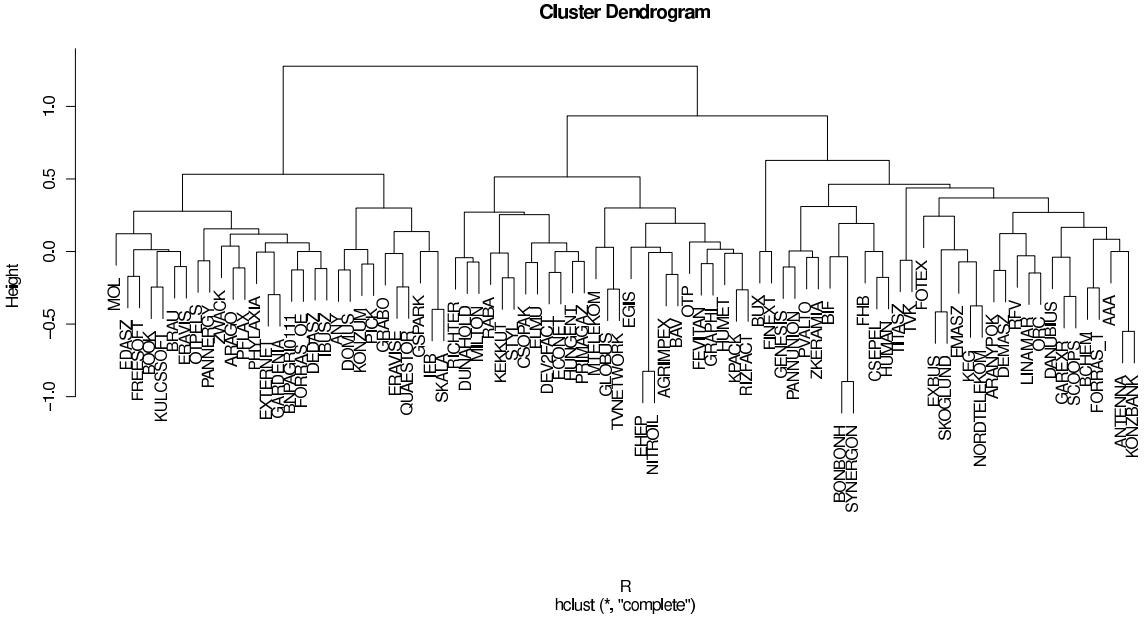
A 3.2 táblázat alapján itt is elmondható, hogy az EGIS és a RICHTER gyógyszergyárak egy klaszterbe kerültek, mégpedig az 1-be. A TVK és a MOL vegyipari vállalatok részvényei is az 1 klaszterben találhatóak.



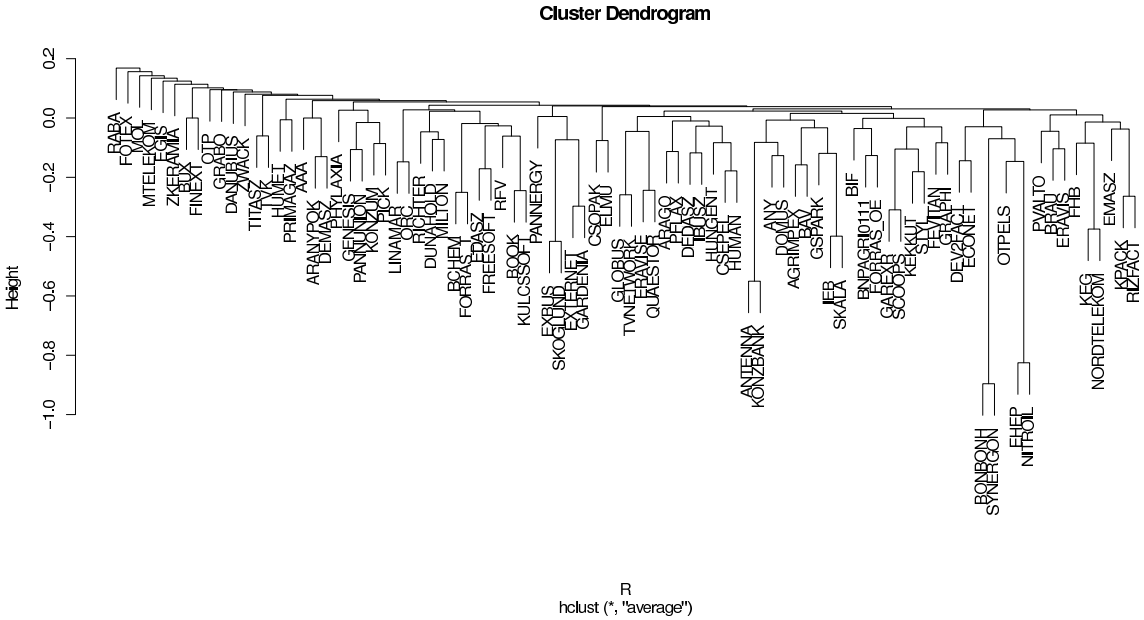
3.7. ábra. R klaszterezésének dendrogramja egyszerű láncmódszerrel

Bux	2devfact	Aaa	Agrimpex	Antenna	Any
1	7	3	7	6	7
Arago	Aranypok	Bav	Bchem	BIF	Bnpagrio
7	7	7	5	7	7
Bonbonh	Book	Brau	Csepel	Csopak	Danubius
7	3	6	7	6	1
Dedasz	Demasz	Domus	Dunahold	Econet	Edasz
7	1	7	7	3	4
Egis	Ehep	Elmu	Emasz	Eravis	Eravise
1	7	3	3	7	4
Exbus	Externet	Fevitan	Fhb	Finext	Forrasoe
5	7	7	3	7	3
Forrast	Fotex	Freesoft	Gardenia	Garexr	Genesis
7	1	3	7	7	3
Globus	Grabo	Graphi	Gspark	Human	Humet
6	5	6	8	5	7
Hungent	Ibusz	Ieb	Keg	Kekkut	Konzbank
7	7	6	3	6	7
Konzum	Kpack	Kulcssoft	Linamar	Milton	Mol
3	7	3	6	7	1
Mtelekom	Nitroil	Nordtelekom	Orc	Otp	Otpels
1	2	7	3	1	6
Pannergy	Pannunion	Pflax	Phylaxia	Pick	Primagaz
5	3	3	6	5	5
Pvalto	Quaestor	Raba	Rfv	Richter	Rizfact
7	7	1	7	1	6
Scoops	Skala	Skoglund	Styl	Synergon	Titasz
4	7	7	7	1	7
Tvk	Tvnetwork	Zkeramia	Zwack		
1	7	5	6		

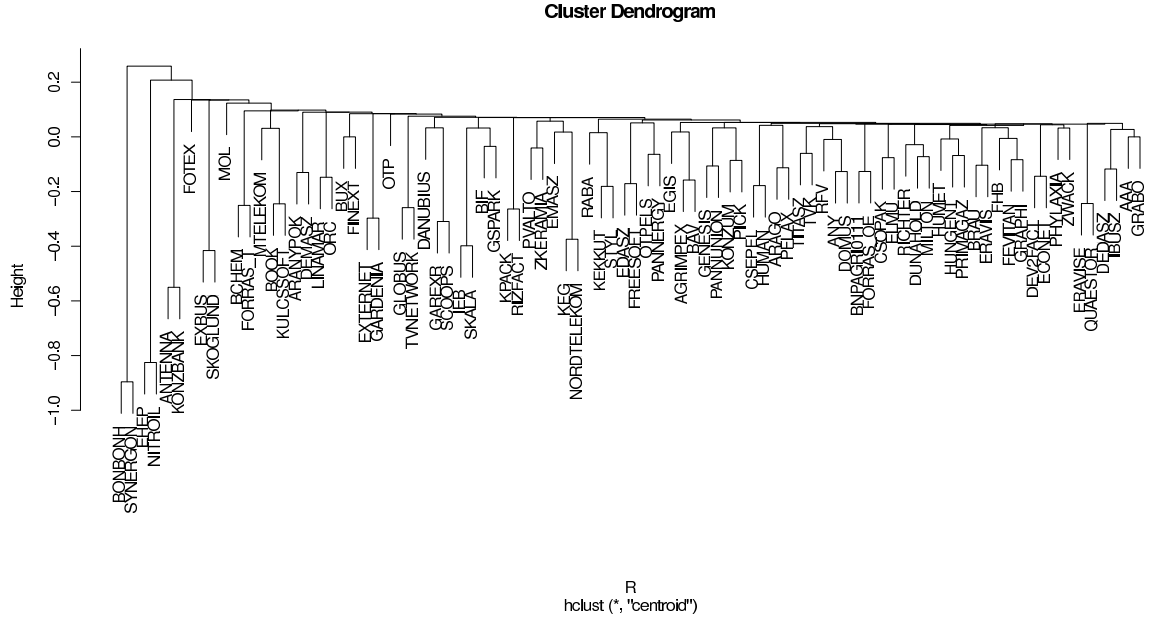
3.2. táblázat. K-közép módszer R-re 8 klaszterrel



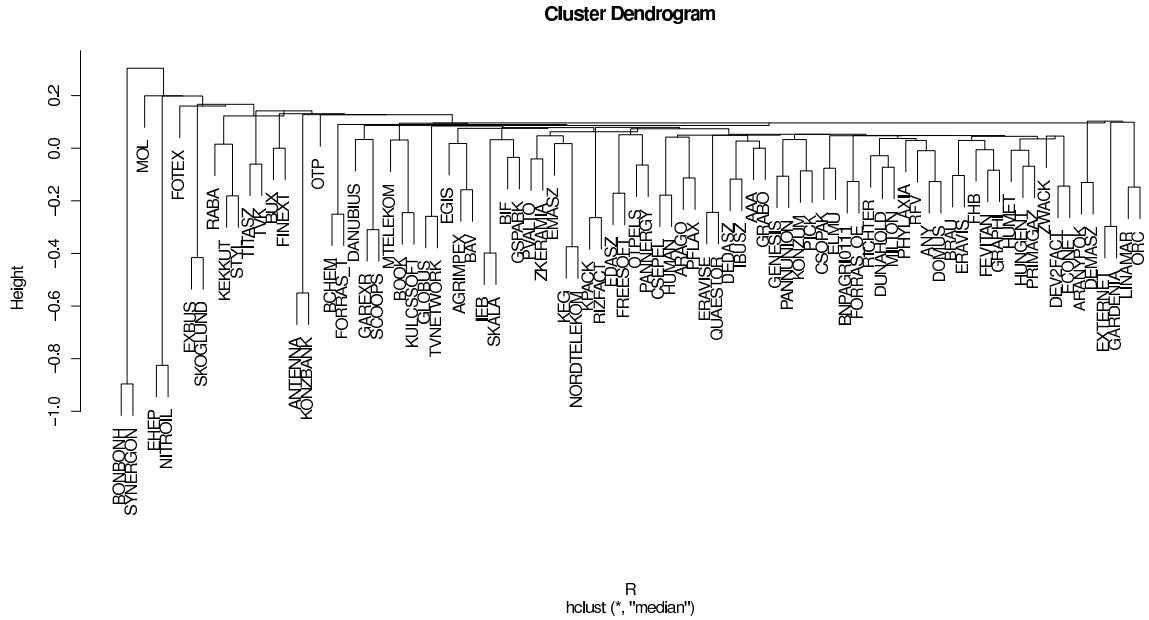
3.8. ábra. R klaszterezésének dendrogramja teljes láncmódszerrel



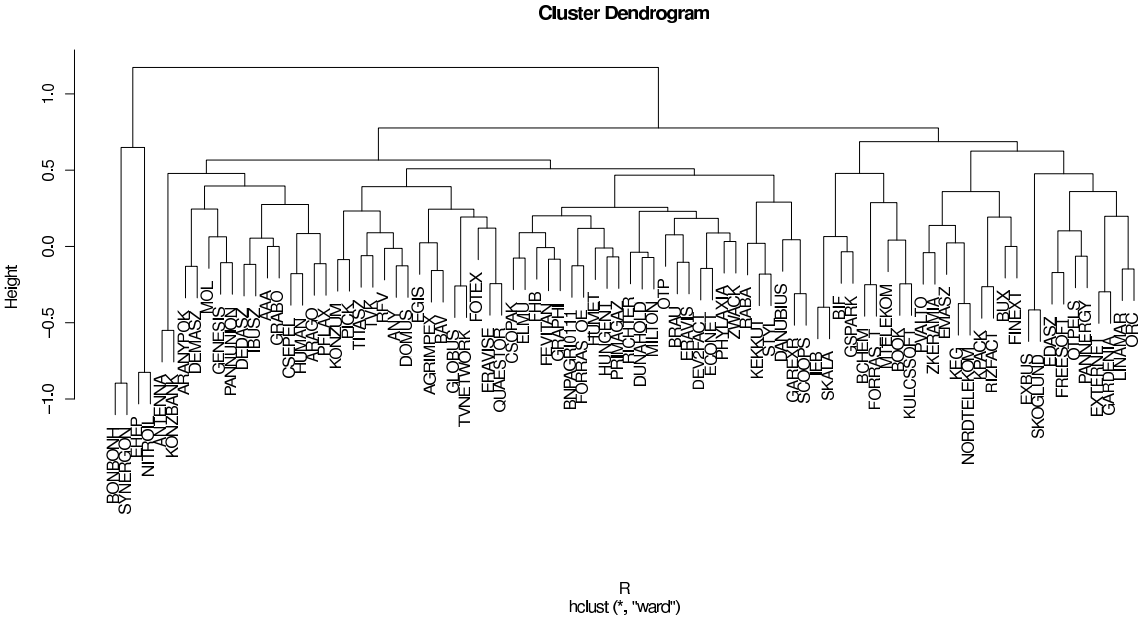
3.9. ábra. R klaszterezésének dendrogramja átlagos láncmódszerrel



3.10. ábra. R klaszterezésének dendrogramja centroid-módszerrel



3.11. ábra. R klaszterezésének dendrogramja medián-módszerrel



3.12. ábra. R klaszterezésének dendrogramja Ward-féle eljárással

4. fejezet

Stabilitásvizsgálat

Az alábbiakban megvizsgálom, hogy mind a részvények egyéni viselkedése szerinti klaszterezések, mind az együttes mozgásuk szerinti klaszterezések mennyire lettek stabilak.

A vizsgálathoz szükségem volt az eredeti S és R adathalmaz mellett torzított adathalmazokra is. Az eredeti adathalmazt úgy változtattam meg, hogy a teljes nX idősor helyett először az idősor első felével, azaz az (1,1612) időintervallummal, majd az idősor második felével, azaz az (1613, 3225) időintervallummal számoltam ki a statiztikai és pénzügyi mutatókat. Az így kapott új adathalmazokat Se, Re, illetve Sm, Rm jelöli, melyeket az R programmal számoltam ki (lásd Függelék).

Az S, Se és Sm, valamint az R, Re és Rm adathalmazokat egyszerű láncmódszerrel, teljes láncmódszerrel, átlagos láncmódszerrel, centroid módszerrel, medián módszerrel, Ward-féle eljárással és k-közép módszerrel klaszteretem. A klaszterek számát 6-nak választottam.

A klaszterezések során keletkezett partíciók összehasonlításához a Rand indexet használtam, amit az R e1071 classAgreement() programjával számoltam ki. A részvények egyéni viselkedése szerinti klaszterezések Rand index értékeit a 4.1 táblázat tartalmazza, az együttes mozgásuk szerinti klaszterezések Rand index értékeit a 4.2 táblázat foglalja össze.

	Egyszerű	Teljes	Átlagos	Centroid	Medián	Ward	K-közép
(S, Se)	0.87304	0.75339	0.82236	0.80642	0.82445	0.63166	0.69540
(S, Sm)	0.87304	0.59587	0.79127	0.81138	0.79127	0.65020	0.68051
(Se, Sm)	0.83176	0.45950	0.62983	0.62983	0.62983	0.43965	0.54414

4.1. táblázat. A Rand index értékei 6 klaszter esetén S-re

	Egyszerű	Teljes	Átlagos	Centroid	Medián	Ward	K-közép
(R, Re)	0.79101	0.69592	0.87304	0.86128	0.80224	0.51253	0.81687
(R, Rm)	0.83176	0.66248	0.91483	0.78239	0.83620	0.56661	0.80120
(Re, Rm)	0.79101	0.67032	0.83176	0.69017	0.68495	0.49555	0.78108

4.2. táblázat. A Rand index értékei 6 klaszter esetén R-re

Amint a táblázatokból kiolvasható az egyszerű és az átlagos láncmódszerrel keletkezett klaszterek a legstabilabbak, míg a legrosszabb Rand indexet eredményező módszernek a Ward-féle eljárás és a teljes lándcmódszer bizonyult.

Megnéztem mi történik akkor, ha a klaszterek számát 6-ról 8-ra változtatom. Az így keletkezett partíciók stabilitását leíró Rand indexeket az 4.3 táblázatban foglaltam össze.

	Egyszerű	Teljes	Átlagos	Centroid	Medián	Ward	K-közép
(S, Se)	0.79623	0.68808	0.71473	0.69696	0.71473	0.69252	0.75444
(S, Sm)	0.83594	0.59299	0.79519	0.81269	0.79493	0.64315	0.71812
(Se, Sm)	0.79623	0.48432	0.55747	0.55773	0.55773	0.49555	0.71290

4.3. táblázat. A Rand index értékei 8 klaszter esetén

Ebben az esetben is azt tapasztaltam, hogy az egyszerű és az átlagos láncmódszerrel kapjuk a legjobb Rand indexeket, míg a Ward-féle eljárás és a teljes láncmódszer eredményezi a kevésbé stabil klasztereket.

Tehát az egyszerű és az átlagos láncmódszer eredményezi a legstabilabb klasztereket a klaszterező eljárások vonatkozása szerinti stabilitásvizsgálatnál.

Érdemes lenne tovább vizsgálni, hogy a klaszterek számának megválasztása, vajon befolyásolja-e a klaszterek stabilitását. Az előbbi két példából úgy tűnik, hogy a hierarchikus módszereknél a klaszterszám növekedése a stabilitás csökkenését eredményezte, viszont a k-közép módszernél a klaszterszám növelésével nőt a stabilitás is. Ebből meg nem vonhatunk le következtetéseket, de lehetőséget ad a stabilitás klaszterszám vonatkozásában való vizsgálatára.

5. fejezet

Összefoglalás

A szakdolgozat célja a magyar részvények elemzése klaszteranalízis segítségével. Egy ilyen elemzés elkészítése befektetőknek lehet fontos.

A részvényeket kétféle módon vizsgáltam, egyéni viselkedésük és egymáshoz való viszonyuk szerint. Az idősorok egyéni viselkedésnek leírására 39 mutatót használtam. A részvények együttes mozgását a 0, 1, 2, 3, 4, 5 nappal eltolt kereszkorrelációk összegével jellemeztem.

Az így kapott adathalmazokon 7 különböző klaszterező eljárást hajotottam végre. Az eredmények azt tükrözték, hogy az azonos gazdasági szektorhoz, területhez tartozó részvények azonos klaszterbe kerültek.

Minden egyes eljárás stabilitását megvizsgáltam. A Rand index szerint mind a 7 módszer stabil partíciókat eredményezett. A stabilitás szempontjából az egyszerű és átlagos láncmódszer bizonyult a legjobbnak. A legrosszabbnak a teljes láncmódszer és a Ward-féle eljárás tekinthető.

További stabilitásvizsgálati lehetőség az osztályszámok szerinti elemzés. Megfigyeltem, hogy a klaszterszámok növelése a hierarchikus módszereknél a stabilitás csökkenését eredményezte, viszont a k-közép módszernél a stabilitás növekedését. Ezen az úton lehetne még tovább vizsgálni a témát.

Irodalomjegyzék

[1] H. H. Bock, *Origins and extensions of the k-means algorithm in cluster analysis*, Electronic Journal for History of Probability and Statistics Vol 4, n°2, December 2008.

[2] N. De Costa, J. Cunha, S. De Silva, *Stock Selection Based on Cluster Analysis*, Economics Bulletin, AccessEcon, vol. 7(3), pages 1-9., 2005.

[3] L. Denoeud, H. Garreta, A. Guénoche, *Comparison of distance indices between partition*

[4] B. Everitt, *Cluster analysis*, Heinemann Educational Books Ltd, 1980.

[5] C. Hennig, *Cluster-wise assessment of cluster stability*, Elsevier, 2006.

[6] C. Hennig, *Dissolution point and isolation robustness: Ropustness criteria for general cluster analysis methods*, Elsevier, 2007.

[7] A. K. Jain, *Data Clustering: 50 Years Beyond K-Means*, Pattern Recognition Letters, 2009.

[8] L. Lovász, M. D. Plummer, *Matching Theory*, Budapest, Akadémiai Kiadó 1986.

[9] R. E. Kalman, *A New Approach to Linear Filtering and Prediction Problems*, Research Institute for Advanced Study, Baltimore

[10] N. Solymosi, *Bevezetés az R-nyelv és környezet használatába*, Solymosi Norbert, 2005.

[11] M. Székelyi, I. Barna, *Túlélőkészlet az SPSS-hez*, Typotex, 2008.

[12] P. N. Tan, M. Steinbach, V. Kumar, *Introdution to Data Mining, Chapter 8., Cluster Analysis: Basic Concepts and Algorithms*, Addison-Wesley, 2005.

[13] E. Tutsek, *Szélmezőtípusok előállítása clusteranalízissel*, Központi Meteorológiai Intézet, 1981.

Függelék

```
# A vizsgalt statisztikak
# -------------------------

s<-mean(x); names(s)<-"mean"
s<-c(s,var(x)); names(s)[length(s)]<-"var"
s<-c(s,quantile(x,p=.50)); names(s)[length(s)]<-"Q50"
s<-c(s,mean(abs(x))); names(s)[length(s)]<-"absmean"
s<-c(s,skewness(abs(x))); names(s)[length(s)]<-"skewness"    # e1071
s<-c(s,kurtosis(abs(x))); names(s)[length(s)]<-"kurtosis"    # e1071
s<-c(s,(s[3]-s[1])/sqrt(s[2])); names(s)[length(s)]<-"mm.dist"
s<-c(s,quantile(x,p=.01)); names(s)[length(s)]<-"Q01"
s<-c(s,quantile(x,p=.99)); names(s)[length(s)]<-"Q99"
s<-c(s,quantile(x,p=.10)); names(s)[length(s)]<-"Q10"
s<-c(s,quantile(x,p=.90)); names(s)[length(s)]<-"Q90"
s<-c(s,quantile(x,p=.25)); names(s)[length(s)]<-"Q25"
s<-c(s,quantile(x,p=.75)); names(s)[length(s)]<-"Q75"
s<-c(s,(max(x)-min(x))/sqrt(s[2])); names(s)[length(s)]<-"rrange"
s<-c(s,(quantile(x,p=.75)-quantile(x,p=.25))/sqrt(s[2]); names(s)[length(s)]<-"r.half"
s<-c(s,mean(abs(diff(x))));names(s)[length(s)]<-"iv.hossz"
s<-c(s,mean(diff(x)^2));names(s)[length(s)]<-"iv2hossz"
s<-c(s,tryCatch(var(x-lowess(x)$y),error=function(e)0,finally=NULL));
 names(s)[length(s)]<-"lowess"    # lok.polinomialis.reg
s<-c(s,tryCatch(ets(x)$sigma2,error=function(e)0,finally=NULL));
 names(s)[length(s)]<-"ets"       # exp.smoothing # forecast
s<-c(s,tryCatch(StructTS(x,type="l")$coef[2],error=function(e)0,f=NULL));
 names(s)[length(s)]<-"kalman"    # Kalman mfi egyenlet hiba-szigma
s<-c(s,sd(volatility(tryCatch(garchFit(~garch(1,1),x,trace=FALSE),error=function(e)0,f=NULL),type="h")));
 names(s)[length(s)]<-"garch"     # Garch(1,1) volatility H
s<-c(s,tryCatch(arima(x,ord=c(1,0,0))$sigma2,error=function(e)0,f=NULL));
 names(s)[length(s)]<-"arma01"    # ARMA
s<-c(s,tryCatch(arima(x,ord=c(0,0,1))$sigma2,error=function(e)0,f=NULL)); names(s)[length(s)]<-"arma10"
s<-c(s,tryCatch(arima(x,ord=c(1,0,1))$sigma2,error=function(e)0,f=NULL)); names(s)[length(s)]<-"arma11"
s<-c(s,tryCatch(arima(x,ord=c(2,0,1))$sigma2,error=function(e)0,f=NULL)); names(s)[length(s)]<-"arma21"
s<-c(s,tryCatch(arima(x,ord=c(1,0,2))$sigma2,error=function(e)0,f=NULL)); names(s)[length(s)]<-"arma12"
s<-c(s,tryCatch(arima(x,ord=c(2,0,2))$sigma2,error=function(e)0,f=NULL)); names(s)[length(s)]<-"arma22"
s<-c(s,tryCatch(arima(x,ord=c(3,0,2))$sigma2,error=function(e)0,f=NULL)); names(s)[length(s)]<-"arma32"
s<-c(s,tryCatch(arima(x,ord=c(2,0,3))$sigma2,error=function(e)0,f=NULL)); names(s)[length(s)]<-"arma23"
s<-c(s,tryCatch(arima(x,ord=c(3,0,3))$sigma2,error=function(e)0,f=NULL)); names(s)[length(s)]<-"arma33"
s<-c(s,tryCatch(arima(x,ord=c(4,0,3))$sigma2,error=function(e)0,f=NULL)); names(s)[length(s)]<-"arma43"
s<-c(s,tryCatch(arima(x,ord=c(3,0,4))$sigma2,error=function(e)0,f=NULL)); names(s)[length(s)]<-"arma34"
s<-c(s,tryCatch(arima(x,ord=c(4,0,4))$sigma2,error=function(e)0,f=NULL)); names(s)[length(s)]<-"arma44"
s<-c(s,tryCatch(arima(x,ord=c(5,0,4))$sigma2,error=function(e)0,f=NULL)); names(s)[length(s)]<-"arma54"
s<-c(s,tryCatch(arima(x,ord=c(4,0,5))$sigma2,error=function(e)0,f=NULL)); names(s)[length(s)]<-"arma45"
s<-c(s,tryCatch(arima(x,ord=c(5,0,5))$sigma2,error=function(e)0,f=NULL)); names(s)[length(s)]<-"arma55"
s<-c(s,tryCatch(arima(x,ord=c(6,0,5))$sigma2,error=function(e)0,f=NULL)); names(s)[length(s)]<-"arma65"
```

```
s<-c(s,tryCatch(arima(x,ord=c(5,0,6))$sigma2,error=function(e)0,f=NULL)); names(s)[length(s)]<-"arma56"
s<-c(s,tryCatch(arima(x,ord=c(6,0,6))$sigma2,error=function(e)0,f=NULL)); names(s)[length(s)]<-"arma66"

# KotLeir.rov
# Kotveny leirasok kigyujtese
# Kotveny.Radat -> Leiras.Rdata
# ------------------------------


load("Kotveny.Rdata");nX<-nX[,2:dim(nX)[2]] # elhagyjuk a datum oszlopot
require(e1071)
require(forecast)#+tseries,quadprog,zoo#
require(fGarch)  #+timeDate,timeSeries,fBasics,Mass#

# ---------------------------------------- TELJES
load("Kotveny.Rdata");nX<-nX[,2:dim(nX)[2]]
k<-1
x<-nX[[k]];x<-x[!is.na(x)]
source("KotStat.rov") # s-be teszi az x statisztikait
S<-matrix(s,ncol=1);
colnames(S)<-colnames(nX)[k]
rownames(S)<-names(s)

for (k in 2:dim(nX)[2])
 {x<-nX[,k];x<-x[!is.na(x)]
  source("KotStat.rov") # s-be teszi az x statisztikait
  S<-cbind(S,s);colnames(S)[dim(S)[2]]<-colnames(nX)[k]
  save(S,file="Leiras.Rdata")
 }

# ---------------------------------------- ELSO
load("Kotveny.Rdata");nX<-nX[,2:dim(nX)[2]]
nX<-nX[1:1612,]

k<-1
x<-nX[[k]];x<-x[!is.na(x)]
source("KotStat.rov") # s-be teszi az x statisztikait
Se<-matrix(s,ncol=1);
colnames(Se)<-colnames(nX)[k]
rownames(Se)<-names(s)

for (k in 2:dim(nX)[2])
 {x<-nX[,k];x<-x[!is.na(x)]
  source("KotStat.rov") # s-be teszi az x statisztikait
  Se<-cbind(Se,s);colnames(Se)[dim(Se)[2]]<-colnames(nX)[k]
  save(Se,file="LeirasE.Rdata")
 }
# ---------------------------------------- MASODIK
load("Kotveny.Rdata");nX<-nX[,2:dim(nX)[2]]
nX<-nX[1613:3225,]

k<-1
x<-nX[[k]];x<-x[!is.na(x)]
source("KotStat.rov") # s-be teszi az x statisztikait
Sm<-matrix(s,ncol=1);
colnames(Sm)<-colnames(nX)[k]
rownames(Sm)<-names(s)
```

```
for (k in 2:dim(nX)[2])
 {x<-nX[,k];x<-x[!is.na(x)]
  source("KotStat.rov") # s-be teszi az x statisztikait
  Sm<-cbind(Sm,s);colnames(Sm)[dim(Sm)[2]]<-colnames(nX)[k]
  save(Sm,file="LeirasM.Rdata")
 }
# ----------------------------------------- VEGE


# KotKorr.rov
# Kotveny korrelaciok kigyujtese
# Kotveny.Radat -> Korrel.Rdata

# ------------------------------------------------ TELJES
load("Kotveny.Rdata")
nX<-nX[,2:dim(nX)[2]]# elhagyjuk a datum oszlopot

n<-dim(nX)[1];m<-dim(nX)[2]
R0<-cor(nX,nX,use='pair')                                                    ;R0[is.na(R0)]<-0
R1<-cor(nX[-1               ,],nX[-n                              ,],use='pair');R1[is.na(R1)]<-0
R2<-cor(nX[c(-1,-2)         ,],nX[c(-(n-1),-n)                    ,],use='pair');R2[is.na(R2)]<-0
R3<-cor(nX[c(-1,-2,-3)      ,],nX[c(-(n-2),-(n-1),-n)             ,],use='pair');R3[is.na(R3)]<-0
R4<-cor(nX[c(-1,-2,-3,-4)   ,],nX[c(-(n-3),-(n-2),-(n-1),-n)      ,],use='pair');R4[is.na(R4)]<-0
R5<-cor(nX[c(-1,-2,-3,-4,-5),],nX[c(-(n-4),-(n-3),-(n-2),-(n-1),-n),],use='pair');R5[is.na(R5)]<-0
R<-as.dist(R0+R1+R2+R3+R4+R5)

# ------------------------------------------------ ELSO
load("Kotveny.Rdata")
nX<-nX[,2:dim(nX)[2]]# elhagyjuk a datum oszlopot
nX<-nX[1:1612,]

n<-dim(nX)[1];m<-dim(nX)[2]
R0<-cor(nX,nX,use='pair')                                                    ;R0[is.na(R0)]<-0
R1<-cor(nX[-1               ,],nX[-n                              ,],use='pair');R1[is.na(R1)]<-0
R2<-cor(nX[c(-1,-2)         ,],nX[c(-(n-1),-n)                    ,],use='pair');R2[is.na(R2)]<-0
R3<-cor(nX[c(-1,-2,-3)      ,],nX[c(-(n-2),-(n-1),-n)             ,],use='pair');R3[is.na(R3)]<-0
R4<-cor(nX[c(-1,-2,-3,-4)   ,],nX[c(-(n-3),-(n-2),-(n-1),-n)      ,],use='pair');R4[is.na(R4)]<-0
R5<-cor(nX[c(-1,-2,-3,-4,-5),],nX[c(-(n-4),-(n-3),-(n-2),-(n-1),-n),],use='pair');R5[is.na(R5)]<-0
Re<-as.dist(R0+R1+R2+R3+R4+R5)

# ------------------------------------------------ MASODIK
load("Kotveny.Rdata")
nX<-nX[,2:dim(nX)[2]]# elhagyjuk a datum oszlopot
nX<-nX[1613:3225,]
n<-dim(nX)[1];m<-dim(nX)[2]
R0<-cor(nX,nX,use='pair')                                                    ;R0[is.na(R0)]<-0
R1<-cor(nX[-1               ,],nX[-n                              ,],use='pair');R1[is.na(R1)]<-0
R2<-cor(nX[c(-1,-2)         ,],nX[c(-(n-1),-n)                    ,],use='pair');R2[is.na(R2)]<-0
R3<-cor(nX[c(-1,-2,-3)      ,],nX[c(-(n-2),-(n-1),-n)             ,],use='pair');R3[is.na(R3)]<-0
R4<-cor(nX[c(-1,-2,-3,-4)   ,],nX[c(-(n-3),-(n-2),-(n-1),-n)      ,],use='pair');R4[is.na(R4)]<-0
R5<-cor(nX[c(-1,-2,-3,-4,-5),],nX[c(-(n-4),-(n-3),-(n-2),-(n-1),-n),],use='pair');R5[is.na(R5)]<-0
Rm<-as.dist(R0+R1+R2+R3+R4+R5)


save(R,Re,Rm,file='Korrel.Rdata')
```