Statisztikai Szemle

Közzététel: 2018. szeptember 28.

A tanulmány címe:

Többváltozós statisztikai R Open alkalmazások

Szerzők:

Hajdu Ottó, az MTA doktora, az Eötvös Loránd Tudományegyetem és a Neumann János Egyetem egyetemi tanára E-mail: hajdu@gti.elte.hu

DOI: https://doi.org/10.20311/stat2018.10.hu1021

Hajdu Ottó,
az MTA doktora, az Eötvös Loránd Tudományegyetem és a Neumann János Egyetem egyetemi tanára
E-mail: hajdu@gti.elte.hu

Többváltozós statisztikai R Open alkalmazások

DOI: 10.20311/stat2018.10.hu1021

Az R programnyelv és annak statisztikai funkciói közismertek, de rugalmassága, számítási lehetőségei olyannyira szerteágazók, hogy a legmegfelelőbb R package (csomag) kiválasztása nehéz. A *Statisztikai Szemle* 2016-ban megjelentetett egy részletes ismertetőt az R programról (*Daróczi* [2016]). Jelen tanulmány célja többváltozós statisztikai módszerekhez R kód szoftverfejlesztési R package felhasználási javaslatokat adni statisztikai módszerspecifikus alkalmazásokra, sajátjogú szoftvert fejlesztve a felhasználó számára.

Javasoljuk e cikket mindazoknak, akik járatosak statisztikai problémák megfogalmazásában, és azok módszertani megoldását saját jogú statisztikai szoftver keretében képzelik el.

1. Bevezető R instrukciók

A kutatási feladatnak leginkább megfelelő számítási rutin, az ún. R package kiválasztása mind tartalmilag, mind formailag meghatározza az R outputot. Ezek tárgyalása nem lehet teljes körű, mert az R package köre folyamatosan bővül.

Figyelmünk a többváltozós statisztikai módszerek felé irányul, az adataink keresztmetszetiek. Nevezetes módszereket kiemelve, esettanulmányokon keresztül tárgyaljuk az eljárás R alkalmazását, ahol az adott szoftver lefuttatása, az eredmények reprodukálása végett az adatok szerves részei a cikknek.

Technikailag az R-nyelv alkalmazásának alapja egy „konzol” mely sorról sorra várja az utasításokat. Soronként külön is lehet érvényesíteni az utasításokat [Enter],

de lehet kötegelten, a konzolba egyszerű szövegmásolás [Ctrl + C; Ctrl + V] alkalmazásával is. Előbbi esetben az eredmények is kötegelten, együtt láthatók.[1]

Az empirikus célú alkalmazásainkhoz értelemszerűen empirikus statisztikai megfigyeléseket használtunk, de *kalkulátor* szándékkal használva a *konzolt*, adatokat generálhatunk, tetszőleges matematikai, statisztikai vagy adatkezelési műveleteket hajthatunk végre.

A tanulmány felépítése a következő. Elsőként tárgyaljuk az alapvető szintaktikát az R konzol használatához, ezt követően a kiválasztott statisztikai modellek specifikus megvalósításait, melyek fő mozzanatai az adatbevitel, a megfelelő/szükséges számítási package-motorok kiválasztása, majd az eredményeket package-specifikusan jelenítjük meg.

A tárgyalt többváltozós statisztikai módszerek a következők: lineáris regresszió, általánosított lineáris regresszió, főkomponens-analízis, kanonikus korrelációszámítás, klaszteranalízis, korrespondenciaanalízis, klasszifikációs módszerek, döntési fa, logisztikus regressziószámítás, Bayes-klasszifikációanalízis és latens változókat tartalmazó SEM (structural equation modeling – strukturális egyenletek modellje).

A felsorolásból kiemeljük a kismintás következtetés „exact logistic" és a latens változós SEM-programjait, mivel ezek az algoritmusok standard statisztikai szoftverekben nehezen hozzáférhetők.

Fontos részlet a felhasználó számára, hogy az R konzol a # (hashmark jel) utáni karaktereket adott sorban nem utasításként, hanem kommentként értelmezi, majd az új sor [Enter] új funkciót nyit. Így a # mögött fogunk magyarázatokat fűzni az egyes programsorok tartalmához, hogy a konzolba másolásuk után a releváns parancssort végrehajtsa a program.

A cikk R programjainak alkalmazása, eredményeinek reprodukálása érdekében az Olvasónak a következő lépéseket kell követnie, miután az R program R projekt verzióját a *https://R-project.org* oldalról vagy a Microsoft R Open verziót a *https://mran.microsoft.com/download* oldalról letöltötte:

> *1*. Az R programokban a csatolt empirikus adatok korrekt, aktuális elérési útvonalának megadása. Az empirikus adatok csak a *Statisztikai Szemle* honlapján megjelenő Mellékletből tölthetők le csv kiterjesztésű fájlok formájában (http://www.ksh.hu/statszemle_archivum#year=2018/issue=10.). Meg kell adni a pontos elérési utat, cikkünkben ez az F meghajtó gyökérkönyvtára. A parancssoroknak ezt a részét át kell írni akkor, ha a fájlok máshova kerülnek.

[1] Az R szoftvercsomaggal kapcsolatban a *Statisztikai Szemlé*ben megjelent tanulmányok: *Daróczi–Tóth* [2013], *Daróczi* [2016]. A szoftvercsomag használatával kapcsolatos információk az interneten is könnyen elérhetők.

2. A parancssorként hivatkozott R kódban a „BeginCopy … EndCopy" bekezdésben foglalt utasítások R konzolba másolása.

3. Az alkalmazott R package programok pótlólagos installálása, ha a program letöltésekor nem történt meg alapértelmezetten.

A számítási eredményeket terjedelmi okból a cikk nem közli, elérésük érdekében az aktuális R parancsokat az R konzolba be kell bemásolni, melyeket ezután végrehajtja a program, és az eredmények megjelennek. A numerikus eredmények elemzése most nem célunk, a hangsúly egy statisztikai szoftverfejlesztési lehetőségen van.

Az R konzol számára készített kódokat a „Begin…End" szekcióval határolt utasítások alkotják, melyeket sorszámozott parancssorok segítségével hivatkozunk meg. Ezen kódok adják a tanulmány fő mondanivalóját, melyek mindegyike egy-egy példa adott statisztika-módszertani területen egy saját fejlesztésű, egyben saját jogú statisztikai szoftver kidolgozására.

Az R függvényekhez rövid magyarázó megjegyzést fűzünk az első megjelenéskor, de később is, ha indokolt, mindig a # komment jelet követően. Az esettanulmányok eredményeinek olvasása az R konzol használatát igényli, a mellékelt adatállományok elérését feltételezve. Természetesen az eredmények papíralapon is lementhetők, de ez terjedelmes oldalszámú, csaknem könyvméretű outputot ad. Fontos szintaktikai instrukció a cikk olvasásához, hogy a megfelelő

```
# BeginCopyTémacím
    R-parancsok (ahol az adatállomány elérési útvonal alapvető)
# EndCopyTémacím
```

blokkban szereplő R parancsokat kell bemásolni[2] a konzolba, ahol azok automatikusan lefutnak az adott adatokon, és az eredmények a képernyőn megjelennek. Fontos, hogy egy algoritmus eredménye lehet numerikus, de lehet egy ábra is. A „plot" ábrákat használva, az R konzol „menüsorában" adott egy „Windows c." legördítési lehetőség, itt lehet kapcsolni a konzol és valamely ábrák megtekintése között.

Az eredmények értelmezését és a többváltozós statisztikai módszerek elméleti hátterének alapvető ismeretét feltételezzük, de erre vonatkozóan útmutatást adunk tanulmányunk Irodalmában. Részletes magyar nyelvű módszertani ismeretet ad az egyes statisztikai témákhoz *Hajdu* [2003].

Jelen cikk számításai az MRO (Microsoft R Open 3.4.4) verzióval készültek. Az MRO konzolja lényegében nem különbözik a klasszikus R project konzoltól, de az MRO gyorsabb és nagyobb (terabájtos Big-data) adattestek elemzésére is képes.[3]

[2] A másolható parancssorokat megtalálják a tanulmány internetes Mellékletében, rtf kiterjesztéssel lásd http://www.ksh.hu/statszemle_archivum#year=2018/issue=10.

[3] Lásd még: https://docs.microsoft.com/en-us/machine-learning-server/what-is-machine-learning-server

2. A jelölésrendszer

Mindenekelőtt a tárgyalt statisztikai módszerek alkalmazásához elengedhetetlen az R szintaktika alapelemeit áttekinteni. E fejezet nem alkalmaz empirikus adatokat, de generál mesterségeseket, az LM(.) parancs futtatja a Lineáris Modellt, aminek szélső esete a lin.regresszió. A „collection" terminológia az R nyelvben a vektor kifejezése, alkalmazása: *c*(elem1, elem2, …). Az R nyelv objektumorientált környezet, az utasításokat (statisztikák, számítási eredmények, grafikák elérése érdekében) a megfelelő objektumokra érvényesítjük. Az objektum elnevezésében a „dot" karakter használata megengedett! Tekintsük az alapvető általános műveleti és statisztikai utasításokat. A „Begin…End" parancssorok nem alkalmaznak ismételten korábban már bevezetett utasításokat, objektumokat, annak érdekében, hogy a „kód" „önállóan" is futtatható legyen. Az R konzol az utasításokban, az objektum megnevezésekben szigorúan megkülönbözteti a kis- és nagybetű használatát (ezzel számos hibaüzenet elkerülhető). Mivel a többváltozós statisztika talán legalapvetőbb matematikai pillére a lineáris algebra, ezért a tárgyalást a mátrixműveletekkel kezdjük.

Az R kódjaink sorát az 1. parancssor „Begin…End" szekciója tartalmazza. A szekció által határolt programsorok bemásolandók az R konzolba [Ctrl + C; Ctrl + V], akár soronként külön-külön [Enter] módon érvényesítve, akár kötegelten egyben, az egész print-output hatást nyerve, „szoftverként".

Figyelem! Az adott parancssor esetén csak a „Begin…End" közötti utasítások másolandók a konzolba.

2.1. Mátrixműveletek, modelljelölések, lineáris regresszió

A következőkben elengedhetetlen mátrixaritmetikai, statisztikai és R konzol szintaktikákra irányítjuk a figyelmet.

Az 1. parancssor nem használ empirikus adatokat, hanem a konzol kalkulátori alkalmazásához, megértéséhez leginkább szükséges szintaktikát vezeti be mesterséges adatok generálásával.

Az alapvető modelldefiníciós jelölésekre a lineáris modellt használjuk kiindulásként, de az itt bevezetett szintaktika más modellek esetében is alkalmazható. Alapvető, hogy a „~" jel bal oldalán szerepel a modell „*Y*" függő változója, jobb oldalán pedig a prediktor „*X*" független változók „+" jellel összekapcsolt köre. A lineáris regressziós példánk nem használ empirikus adatokat, de generál egy háromváltozós ($y \sim x + d$) mesterséges adatállományt, a lineáris lm(.) modellfüggvényt illusztrálandó. A program ábrázolja az „*x*-time-proxy" és a „*d*-strukturális törés" dummy változó létrehozásának a mikéntjét is. A számítási modell eredményeket a „lin.reg.h0" nevű objektumba foglaltuk, és kitérünk arra is, hogyan lehet egy objektum elemeit lekérdezni, kilistázni. Az output listázásának többféle lehetősége is rendelkezésre áll.

Visszatérve a mátrixalgebrához, a következőkben olyan mátrixműveletek R használatát tárgyaljuk, melyek az empirikus adatok esetében is alkalmazhatók. Definiálunk három változót, rendre [*y*, *x*, *d*], majd ezeket mátrixba foglaljuk, és oszlop/sor, majd sor/oszlop irányú műveleteket végzünk. A mátrixinvertálás és a sajátérték-, sajátvektor-utasítások is megjelennek. A mátrix egyedi elemére hivatkozás: [*i*, *j*], ahol *i* a sor, *j* pedig az oszlop indexe és [*i*, *j*] mátrixot (adatállományt) azonosít. Az [*i*,] argumentum sorra, a [,*j*] argumentum pedig oszlopra hivatkozik. A korábban már tárgyalt szintaktikán túlmenően a # komment megjegyzések adnak további eligazítást az alfejezet programjához.

Alapvetőként megjelenik az alfejezetben az SVD (singular value decomposition – szinguláris értékfelbontás) utasítása is, elsősorban a főkomponens-analízis és a korrespondenciaanalízis tárgyalása érdekében, magában foglalva a sajátérték, sajátvektor feladatot is, speciális esetként, valamint látható a mátrixinvertálás és mátrixszorzás speciális R szintaktikája is. Mindazonáltal előállítjuk az SSCP (sums of squares and cross products – négyzetösszegek és keresztszorzatok mátrixa) szóródási mátrixot is programozva.

A következő többváltozós alkalmazás a Wilks-lambda varianciahányados mérőszám példáján keresztül mutatja be adott mátrix determinánsának és sajátérték, sajátvektor számítási részleteit. Ennek érdekében előbb definiálunk egy külső, majd egy belső kovarianciamátrixot, melyek összege a totális kovarianciamátrixot adja. A Wilks-lambda a belső variancia százalékos arányát számszerűsíti a totális általános variancián belül. Ezért komplementere értelemszerűen az ún. külső „variance explained" mérték. Általánosságban a GV (generalized variance – többváltozós variancia) mértéke a vonatkozó adattér kovarianciamátrixának a determinánsa.

$$Wilks\text{-}lambda = GV_B / GV_T$$

Ez a példa telefonbeszélgetések ideje (perc) és költsége (forint) kétdimenziós pontfelhőjének a szóródását vizsgálja díjkörzetek csoportjainak a függvényében. (Lásd az 1. parancssort.)

1. parancssor

```
# BeginCopyLineárisAlgebra
    1:10 # Futóindex definiálása: 1-től 10-ig egyesével
    x <- 1:10 # Az x-nek_nevezett objektumba való értékadás operátora: <-
    x # Kiírja az x objektumot
    print(x) # Szintén kiírja az x objektumot
    show(x) # Szintén kiírja az x objektumot
    summary(x) # x összefoglaló jellemzőit mutatja
    sd(x) # x szórása, a standard „deviation"
```

```
mean(x) # x átlaga
mode(x) # x adattípusa (numerikus vagy karakter-e stb. ?)
z <- mode(x) # z adatmódjának definiálása
z # z módjának lekérdezése, kiíratása
z <- is.factor(x) # z kategória (faktor, kategória) kimenetű-e ?
z # Válasz a faktor kérdésre
z <- is.numeric(x) # z folytonos kimenetű
z # Print z
length(x) # x mintaméretének az ellenőrzése
attributes(x) # Null
NULL # Ez a válasz
rm(z) # Remove z
z # A kizárás hatására z már nem elérhető
z <- seq(1,100,by=10) # Új z tízesével léptetve, de nem haladva meg a 100-at
z # Print új z
z <- seq(1,100,length=10) # Az intervallum beosztás 100-ig megy, 10 szegmensre hasítva
z # Print z
y1 <- c(3,5,4,7) # Egy 4 elemű vektor (collection) létrehozása
y1 # Print y1
y2 <- c(6,8,10,8,10,13) # Egy másik, 6 elemű vektor (collection) létrehozása
y2 # Print y2
y <- c(y1,y2) # A két y1, y2 collection kombinálása egy közös vektorba
y # Print y
length(y) # Adatméret lekérdezése
d <- c(rep(1,3),rep(0,7)) # Elemek ismétlése dummy definiálása érdekében, melyben 3 db 1
   és 7 db 0 van
d # Print d
y[2] # y 2. eleme
y[2:5] # y 2., 3., 4., 5. elemei
y[ c(2,4,10) ] # y 2., 4., 10. elemei
y[ -c(2,4,10) ] # y elemei a 2., 4., 10. elemek nélkül
y[ (y<5) ] # A logikai kiértékelés leszűkíti a printelést az 5-nél kisebb y értékekre
# A lineáris OLS regressziós modell R-definiálása:
lin.reg <- lm(y~x+d)    # „y is regressed on x, d”, az „lm” lineáris modellben, a Const alapér-
   telmezett tartozék
   # Az eredményeket tartalmazó „lin.reg” objektum nevében megengedett a Dot: "." sze-
   parátor
   # A prediktor listában a magyarázó változók szeparáló (felsoroló) jele: +
lin.reg # Az objektum nevének beírása a konzolba és érvényesítése [Enter] megmutatja a
   regressziós eredményeket
lin.reg.h0 <- lm(y~-1+x+d) # Elhagyjuk a tengelymetszetet a „–1” prediktor szerepeltetésé-
   vel
lin.reg.h0 # Kiírja a lineáris regressziós h0-t
summary(lin.reg.h0) # Az „lm” modell összefoglaló eredményei
objects(lin.reg.h0) # Fontos: listázza az elérhető,lekérhető „lm” objektumokat
ls(lin.reg.h0) # Fontos: így is listázhatjuk a lekérhető „lm” objektumokat
```

```
# A parciális lineáris regressziós koefficiensek számítása az OLS-becslőfüggvény programja
    alapján
    Const <- c(rep(1,10)) # A Const tag 10 elemű összegző vektorának előállítása
    Const # Az összegző vektor kiíratása
    X <- matrix(c(Const,x,d), ncol=3) # Const, x, d háromoszlopos mátrixa oszlopfolytonosan, az
        „X” objektumba foglalva
    X # Az „X” objektum elemeinek a kiírása
    X[1,1] # Az „X” objektum [1,1] elemének kiírása, hivatkozása
    X[2,] # Az „X” objektum [2,] sorának kiírása, hivatkozása
    X[,2] # Az „X” objektum [,2] oszlopának kiírása, hivatkozása
    t(X) # Az „X” mátrix transzponálása
    SSCP <- t(X) %*% X # A mátrixszorzás most minden oszlopot minden oszloppal skalárisan
        szoroz
    SSCP <- crossprod(X,X) # Ekvivalens, tömörebb módon a keresztszorzat-négyzetösszeg
        (SSCP-) mátrix
    solve(SSCP)%*% t(X) %*% y # Az SSCP-mátrix invertálása és a lineáris regressziós koeffici-
        ensek definíció szerinti számítása

# A Wilks-lambda alkalmazás
    Belso <- matrix(c(3, 7.5, 7.5, 21), ncol=2)  # A belső Cov mátrix (kovarianciamátrix) létreho-
        zása, hipotetikus elemek
    Belso # A belső Cov mátrix kiíratása
    Kulso <- matrix(c(15, 6, 6, 4), ncol=2) # A külső Cov mátrix létrehozása, hipotetikus elemek
    Kulso # A külső Cov mátrix kíratása
    Total <- Belso + Kulso # A totális = Belső + Külső Cov mátrix
    Total # A totális Cov mátrix kiíratása
    GV.B <- det(Belso) # A belső általánosított variancia mint a belső kovarianciamátrix deter-
        minánsa
    GV.B # A determináns kiíratása
    GV.T <- det(Total) # A totális általánosított variancia
    GV.T # GV.T kiíratása
    Wilks <- det(Belso) / det(Total) # A Wilks-lambda definíció szerinti számítása
    Wilks # A Wilks-lambda értéke
    1-Wilks # A komplementer Wilks jelentése: „variance explained”

# A Wilks-lambda sajátérték-alapú számítása
    B.K <- solve(Belso) %*% Kulso # Belső kovarianciamátrix inverze szorozva a külső
        kovarianciamátrixszal
    eigen <- eigen(B.K) # A szorzatmátrix sajátértékszámítása
    eigen # A sajátértékek számítása
    eigen$value # A $ jel az eigen objektum $value információját nyitja meg
    eigen$vectors # Az eigen objektum $vektor (sajátvektor) információját mutatja
    eigen$value[1] # Az 1. sajátérték megtekintése
    eigen$value[2] # A 2. sajátérték megtekintése
    Wilks <- 1 / ( 1 + eigen$value[1] ) / ( 1 + eigen$value[2] )
    Wilks # Wilks =
# EndCopyLineárisAlgebra
```

2.2. Empirikus adatbeolvasás, regressziószámítás

Ebben az alfejezetben a parciális PLS-t (partial least squares – legkisebb négyzetek módszere) alkalmazzuk. Az empirikus illusztratív adatállomány autómárkák gyári jellemzőit tartalmazza. A változók rendre: *hengerűrtartalom* (cm^3), *lóerő* (LE), *végsebesség* (km/h), *gyorsulás* 100 km/h (mp), *fogyasztás 90, 130 km/h sebesség mellett és városban* (liter), majd *a súlytömeg* (t). A „text" szövegformátumú adatállomány neve csv kiterjesztéssel: „Autok.csv". A „.csv" text adatformátum alkalmazása preferált! Az adatokat a „read.table" utasítás olvassa be, a szeparálásukra választott karakter most éppen a semicolon (pontosvessző). Az „NA" szimbólum jelentése: not available, vagyis „missing value", míg a dec="." szintaktika a tizedesjel megválasztását teszi lehetővé. A „header=TRUE/FALSE" parancs a változónevek megadását értelmezi az első adatrekordban, ha adottak. Az eredmények a következő programmal érhetők el.

2. parancssor

```
# BeginCopyPLS
   # Input adatok megadása
      Autok <- read.table("F:/Autok.csv", header=TRUE, sep=";", na.strings="NA", dec=".",
         strip.white=TRUE)
      summary(Autok) # Az adatok összefoglaló statisztikai leírása
      fix(Autok) # Az adatállomány editálását teszi lehetővé, ha szükséges
      cor.Autok <- cor(Autok[,c("f90","f130","vf","gy100","henger","loero","tomeg","vegseb")],
         use="complete") # A változók korrelációs mátrixának számítása
      cor.Autok # A korrelációs mátrix kiíratása
      cov.Autok <- cov(Autok[,c("f90","f130","gy100","henger","loero","tomeg","vegseb","vf")],
         use="complete") # A változók kovarianciamátrixának számítása
      cov.Autok # A kovarianciamátrix kiíratása
   # A PLS-regresszió végrehajtása
      library(pls)  # Betölti a szükséges PLS packege-t!
      Autok.PLS.Comp <- plsr(f90 ~ f130+vf+gy100+henger+loero+vegseb+tomeg, ncomp=3,
         data=Autok, scale=TRUE, validation="CV")
      Autok.CV <- MSEP(Autok.PLS.Comp, estimate=c("train", "CV"))
      plot(Autok.CV, col=1, type="l", legendpos="topright", main="")
      Component <- which.min(Autok.CV$val["CV",,])-1
      Component
      Auto.PLS.reg <- plsr(f90 ~ f130+vf+gy100+henger+loero+vegseb+tomeg,
         ncomp=Component, data=Autok, scale=TRUE)
      Auto.PLS.res <- residuals(Auto.PLS.reg)
      plot(Auto.PLS.res[,,Component],pch=15,cex=0.5, ylab="Residuals", main="" )
      abline(h=c(-2, 0, 2), lty=c(2,1,2))
      summary(Auto.PLS.reg )
#EndCopyPLS
```

2.3. Főkomponens-analízis

A principal components (főkomponensek) a statisztikai modellezés alapvető pillérét képezik, módszertanuk közismert. A PCA (principal componenet analysis – főkomponens-analízis) alkalmazására példaként egy kereskedelmi bank ügyfélkörének pénzügyi minősítését használjuk mérleg- és eredménymutatóik alapján. A vizsgált ügyfélkört szám szerint 24 vállalkozás alkotja, az indikátorok definíciói pedig rendre a következők:

1. gyors likviditási ráta = (forgóeszköz – készlet) / rövid lejáratú kötelezettség
2. likviditási ráta = forgóeszköz / rövid lejáratú kötelezettség
3. eladósodottság = 100 * hosszú lejáratú kötelezettség / (hosszú lejáratú kötelezettség + saját tőke)
4. bonitás = 100 * hosszú lejáratú kötelezettség / saját tőke
5. eszközarányos jövedelmezőség = cash-flow / eszközök
6. árbevétel-arányos jövedelmezőség = cash-flow / nettó árbevétel.

Az adatállomány neve: „Ugyfel.csv”. A PCA lényegileg – információtömörítés végett – az indikátorokat sűríti főkomponensekbe, melyek száma nem haladhatja meg az indikátorok számát. Az első komponens hordozza az adattestben levő szóródási információ maximált hányadát, majd a következő komponensek magyarázó ereje, varianciája „lecseng”. Példánkban – szóródási értelemben – a fontossági sorrend rendre: 2,606, 1,881, 1,222, 0,166, 0,085, 0,041, melyek összege 6, és a kumulált százalékos megoszlásuk rendre: 43,441, 74,788, 95,148, 97,914, 99,325, 100. Láthatóan az első három főkomponens nyújtja a 6 indikátor 6 egységnyi információjának döntő, 95,148 százalékát. A 6 darab variancia módszertanilag a (6 × 6) rendű korrelációs mátrix 6 darab sajátértéke. A főkomponensek ügyfélsorosan egy-egy 24 elemű vektort alkotnak, és közgazdasági értelmüket azon pénzügyi indikátorok adják, melyekkel a legszorosabban korrelálnak.

Speciálisan két megoldást tárgyalunk: a „FactoMineR” package PCA függvényét és az SVD lineáris algebrai módszer svd(.) függvényének az alkalmazását használjuk. A FactoMineR package installálandó![4] A FactoMineR megközelítés praktikusan, az SVD R-megoldás pedig elméletileg fontos. Míg a FactoMineR szoftveralkalmazás figyelemfelhívó, addig az SVD program az elméleti háttérre helyezi a hangsúlyt. Az SVD ügyfélkör-alkalmazás R programja jelen cikk szerzőjének hozzájárulása.

[4] A FactoMineR package installálása: az > R prompt alatt a konzolban érvényesíteni kell a source („http://factominer.free.fr/install-facto.r”) utasítást.

3. parancssor

```
# BeginCopyFőkomponens
  # A FactoMineR megoldás
    Ugyfel <- read.table("F:/Ugyfel.csv", header=TRUE, sep=";", na.strings="NA", dec=".",
      strip.white=TRUE, row.names=1) # Adatbeolvasás szövegfájlból
    show(Ugyfel) # Lássuk az alapadatokat
    summary(Ugyfel) # Az adatállomány leíró statisztikái
    library(FactoMineR) # A FactoMineR packaget tölti be a memóriába, melynek PCA-
      függvényét használjuk a következőkben
    Ugyfel.pca <- PCA(Ugyfel, ncp=3) # A PCA-függvénnyel hozza létre az Ugyfel.pca objektu-
      mot úgy, hogy csak az első legnagyobb sajátértékű főkomponenst tartsa meg
    summary(Ugyfel.pca) # Kiírja a korrelációs mátrix sajátértékeit és azok kumulált megoszlá-
      sát, továbbá a faktorsúlyokat, a négyzetes faktorsúlyokat és a négyzetes faktorsúlyok re-
      latív hozzájárulását saját főkomponensük sajátértékéhez.
  # Az SVD-megoldás
    # Az SVD-dekompozíció egy mátrixalgebrai módszer, aminek alkalmazása mátrix típusú ob-
      jektumok használatát igényli.
    # Ezért az eredeti text_fájl adatállományt mátrix típusúvá kell konvertálni.
    # Az SVD-dekompozíció formálisan: X = U * D * V_transzponált , ahol X tetszőleges (n, p)
      rendű „real-value” matrix:
    # U oszlopai a bal oldali, V oszlopai pedig a jobb oldali szinguláris vektorokat tartalmazzák,
      míg D egy (p, p) rendű diagonális mátrix,
    # melynek főátlóján az ún. szinguláris értékek szerepelnek.
    # V oszlopait az X adatmátrix R korrelációs mátrixának a sajátvektorai adják, míg a D szingu-
      láris értékek az R korrelációs mátrix sajátértékeinek a négyzetgyökei.
    # Lévén a sajátérték jelentése itt a főkomponens varianciája, ezért a szinguláris érték jelen-
      tése: szórás.
    X <- as.matrix(Ugyfel) # Az X objektum már mátrixtípusban tartalmazza az alapadatokat
    X <- scale(X) # Az X mátrix oszlopainak a standardizálása
    X <- X / sqrt(23) # Az X * X keresztszorzatmátrix normálása a korrigált mintamérettel
    X # A standardizált X mátrix ellenőrzése
    F <- svd(X) # Az SVD-függvény hajtja végre a számításokat, F tartalmazza az eredményeket
    F # Az eredmények kiíratása
    # F$u # Csak F főkomponenseit (bal oldali szinguláris vektorai) íratja ki
    # F$d # Csak F diagonális, szinguláris értékeit íratja ki: sqrt(sajátérték)
    # F$v # Csak a sajátvektorokat (jobb oldali szinguláris vektorokat) íratja ki
    St.fokomponens <- scale(F$u) # A standardizált főkomponensek létrehozása
    St.fokomponens # A standardizált főkomponensek kiíratása
# EndCopyFőkomponens
```

2.4. Kanonikus korrelációszámítás

A kanonikus korrelációszámítás két változókör között méri a korreláció szorosságát. Példánkban a befektetési portfóliók alkotják a köröket, rendre: részvények, valuták és állampapírok. Az adatállomány a „PortfolioCancorr.csv” fájlban érhető el.

Az általunk vizsgált párosításban a „Set1” csoport változói képviselik a *részvény-portfólió árfolyamait*, a „Set2” csoport pedig a *valuták árfolyamait*. A két kör változóinak számossága általában különbözhet egymástól, de esetünkben $p = 5$ részvény és $q = 5$ valuta kapcsolatának szembeállítását hajtjuk végre, nevezetesen:

X = [OTP, Richter, Telekom, MOL, TVK], Y = [EUR, USD, CHF, GBP, JPY] keresztárfolyamok a forinttal szemben.

Módszertanilag az X = Set1 és az Y = Set2 változóit külön-külön sűrítjük egy-egy mesterséges kanonikus CV_x és CV_y változóba, és mérjük közöttük a klasszikus lineáris korrelációs együtthatót úgy, hogy e korreláció értéke maximalizált legyen. A CV_x és CV_y változók kanonikus CV változópárt alkotnak. Ilyen kanonikus változópárból legfeljebb $\min(p, q)$ darab állítható elő, példánkban 5. A maximalizált kanonikus korrelációk jelen esetben rendre: 0,9674451, 0,7450019, 0,5251022, 0,4565509, 0,2439582.

Az X, Y változók között lehet irányított $X \rightarrow Y$ oksági kapcsolat, de lehet az oksági irány kölcsönös $X \leftrightarrow Y$ is. Felhívjuk a figyelmet a „yacca” R-package alkalmazására kanonikus korrelációszámítás esetén.

4. parancssor

```
# BeginCopyKanonikusKorr
   Portfolio <- read.table("F:/PortfolioCancorr.csv", header=TRUE, sep=";", na.strings="NA",
      dec=".", strip.white=TRUE, row.names=1)
   summary(Portfolio)
   Reszveny <- Portfolio[,2:6 ] # Az adatállomány 5:7 oszlopai alkotják a Reszveny set elemeit
   Valuta <- Portfolio[,11:15 ] # Az adatállomány (11:15) oszlopai alkotják a Valuták set elemeit
   library(yacca) # A javasolt package betöltése
   cca.fit <- cca(Reszveny, Valuta) # cca.fit őrzi meg az eredményeket
   summary(cca.fit) # Egy részletes eredménylista
   plot(cca.fit)
# EndCopyKanonikusKorr
```

2.5. Klaszteranalízis

A klaszteranalízis olyan adatsűrítési módszer, mely a megfigyelési egységek számát hivatott csökkenteni. Visszatérünk a banki ügyfélkör „Ugyfel.csv” adatállomány elemzéséhez. Két alapvető eljárást, a hierarchikus agglomeratív klaszterfa- (dendrogram-) módszert, majd az ún. K-közép iteratív R technikáját tárgyaljuk.

A hierarchikus, agglomeratív algoritmus minden lépésben két klasztert von össze közös klaszterbe, éppen az egymáshoz legközelebbi, leghasonlóbbat. Az eljárás indulásaként minden ügyfelet önálló klaszterként tekintve, azok fokozatos összevonásával jutunk el az ügyfélkör egészéhez, a teljes sokasághoz. Hierarchikus pedig abban

az értelemben, hogy azok az ügyfelek, akik már közös klaszterbe kerültek korábban, a következő lépések összevonásai során már együtt is maradnak. Értelemszerűen a 24 ügyfelet 23 lépésben vonjuk össze egyetlen teljes ügyfélkörré.

Ezzel szemben, a K-közép algoritmus igényli a klaszterek K-számának rögzítését, még az eljárás elején. Például $K = 5$. Ez függ a mintaérettől. Ezután meg kell adni az egyes klaszterek induló tagságait, melyek a klaszterek induló centroidjait is adják. E centroidok körül értelmezzük az ügyfelek klaszteren belüli szóródását. Ezután minden *ügyfél* klaszter centroidjától vett távolságát mérjük, és az *ügyfelet* ahhoz a klaszterhez rendeljük, melynek a centroidjához a legközelebb áll. Ha történt átsorolás, megváltoznak a centroidok, és a vizsgálat újra indul. Az eljárás akkor terminál, mikor mindenki (minden *ügyfél*) a saját centroidjához áll a legközelebb. Az induló klasztertagságok megadására többféle módszer is rendelkezésre áll. A legegyszerűbb, hogy véletlenszerűen kiválasztunk K-számú random *ügyfelet* mint centroidot, illetve egy hierarchikus klaszter centroidjai adják az induló középpontokat, vagyis az induló klasztertagságokat. A középpontok nem föltétlenül számtani átlagok, lehetnek például robusztus mediánok sorozatai is!

Az alapvető cél a szeparáció és a kohézió együttes érvényesülése, ahol szándékunk egymástól minél távolabbi klaszterek, de klaszteren belül átlagosan minél homogénebb „szomszédok” feltárása, ha van ilyen trend, struktúra egyáltalán az adatok között. A klaszteranalízis egy exploratív célú statisztikai technika, melynek során távolságot kell mérni, és az agglomerációban már meglevő klasztereket kell összevonni. Definiálni kell tehát opcionálisan egy távolságmetrikát és egy klaszteregyesítési szabályt.

5. parancssor

```
# BeginCopyKlaszter
    Ugyfel <- read.table("F:/Ugyfel.csv", header=TRUE, sep=";", na.strings="NA", dec=".",
      strip.white=TRUE, row.names=1)
    Ugyfel # Az adatok megjelenítése
    summary(Ugyfel) # Az adatok leíró statisztikái
    d = dist(model.matrix(~-1+adossag+bonitas+eszkjov+gylikvr+likvr+narbjov, Ugyfel)) # A
      „dissimilarity” távolságmátrix létrehozása
    d # A „d” mátrix kiíratása
    library(cluster)
    Ugyfel.clust <- agnes(scale(Ugyfel[,1:6]),metric="euclidean",method="average") # Az agnes
      függvény alkalmazása
    plot(Ugyfel.clust$height, type="h", main="Dendrogram", xlab="Individual", ylab="Height")
    plot(Ugyfel.clust,main="Dendrogram", xlab="Individual", ylab="Height")
    Ugyfel.5cl <- cutree(Ugyfel.clust, k=5) # Ez egy új, klasztertagság-változó
    Ugyfel.5cl # Az 5 klasztertagság kiíratása
    Ugyfel.5cl <- as.factor(Ugyfel.5cl)
```

```
    Ugyfel.comp <- cbind.data.frame(Ugyfel,Ugyfel.5cl)
    colnames(Ugyfel.comp) [7] <- "5Klaszter" # A 7. oszlop fejrovata
    Ugyfel.comp # Az Ügyféladat-állomány és az 5 klasztertagságok együttes kiíratása

# Egy alternatív, FactoMineR-package, hierarchikus klaszterfüggvény Ward-módszerrel, a fő-
    komponenseken alkalmazva
    library(FactoMineR)
    Ugyfel.pca <- PCA(Ugyfel,ncp=3,graph=F)
    Ugyfel.hcpc <- HCPC(Ugyfel.pca, consol=FALSE)
    Ugyfel.hcpc # Az Ugyfel.hcpc objektum eredményeinek lekérdezése külön-külön
    # Egy alaternatív klaszter package függvény a főkomponenseken hajtva végre a klaszteree-
      zést
    # Ugyfel.hcl <- hclust(d , method = "ward.D2", members=NULL) # A hclust függvény alkal-
      mazásánál a Ward-módszer neve: „ward.D2)”
    # plot(Ugyfel.hcl,main="Ügyfélklaszterek dendrogramja", xlab="Ügyfelek", sub="Ward
      módszer; Távolság: euklideszi")

# K-közép klaszter
    Ugyfel.k.means <- kmeans(scale(Ugyfel[,1:6]),centers=5)
    # A K-középpontú klaszterezés számítása és numerikus eredményei, az 1–6 adatoszlopok,
      mint klaszterező változók alapján 5 klaszterre bontva
    Ugyfel.k.means # A K-közép fő eredményei
    Ugyfel.k.means$cluster # Csak a K-közép klasztertagságai
    Ugyfel.k.means$centers # Csak a K-közép klasztercentroidjai
    Ugyfel.k.means$size  # Csak a klasztermeretek
# EndCopyKlaszter
```

2.6. Egyszerű és többszörös korrespondenciaanalízis

A korrespondenciaanalízis egy vizuális többváltozós adatredukciós exploratív, asszociációs módszer, melynek adatállományát kategóriakimenetű változók alkotják, és alapvető célja, hogy az együtt gyakran előforduló kategóriákat a síkon ábrázolva a lehető legközelebb húzza egymáshoz, vagyis asszociálja. A módszer kategóriakimenetű adatok főkomponens-analízisének is tekinthető, mellyel a kategóriák közötti asszociációkat mutatjuk ki. A módszer két változata: a CA (correspondence analysis – egyszerű korrespondenciaanalízis) és az MCA (multiple correspondence analysis – többszörös korrespondenciaanalízis).

A két módszert az alkalmazott adatállomány sajátosságai és konklúziói különböztetik meg, a következők szerint.

Az egyszerű korrespondenciaanalízis egy statisztikai kontingenciatábla gyakoriságait vetíti vizuálisan a síkra, mint egy térképre úgy, hogy a sorkategóriák – mint pontok az oszloptérben – a lehető legközelebb helyezkedjenek el azon oszlopkategóriákhoz, melyekkel szorosan asszociálnak, ha ilyen sor-, oszlopkategória-

asszociációk egyáltalán léteznek az adatállományban. Például a bekapcsolt biztonsági öv, sérülés esetén vonzza a könnyű sérülést, míg a be nem kapcsolt öv többnyire súlyos sérüléssel jár. A CA-analízis egy gyakorisági tábla vizuális értelmezése abban az értelemben, hogy mely sorkategóriák mely oszlopkategóriákat vonzzák, és melyeket taszítják. A változók számát redukáljuk minél kevesebb dimenzióra, és törekszünk egy- vagy kétdimenzióban ábrázolni. Ha a megfelelően alacsony adatredukciós veszteség kettőnél több dimenziót igényel, akkor a síkbeli ábrázolás dimenziópárosításokkal történik.

A CA-példánkban a tárgyalt kategóriák épületek tervtípusa és jellege szerint a következők:

1. Az építési terv skálája: [*típus*, *ajánlott*, *egyedi*, *ismételt terv*].

2. Az épület jellegének skálája: [telepi_többszintes, egyedi_többszintes, csoportos, családi_ház, egyéb].

A feltárt asszociációk ismeretében ún. kiegészítő (supplementary) sorszerkezetek előrejelezhetők az oszlopok terében és megfordítva. Példánkban kiegészítő sorok a *panel*- vagy a *téglafalszerkezet* és kiegészítő oszlopok a *hitel*- vagy *magánfinanszírozás*.

A CA-output tartalmazza az egyes dimenziók ún. inerciáinak kumulált százalékos megoszlásait, a sorok, az oszlopok és a cellák totális χ^2-inerciához való hozzájárulásait, a kontingenciatábla gyakoriságainak százalékos struktúráját, valamint az egyes pontok (sorok, oszlopok) síkbeli „Cos2” reprezentáltságát. A síkbeli ábrázolás végett megjelennek a megfigyelt és kiegészítő sorok-oszlopok korrespondenciakoordinátái is.

Ezzel szemben a többszörös korrespondenciaanalízis (MCA-) adatállománya klaszszikus többváltozós (n, p) rendű adatmátrix, mely i soraiban (rekordjaiban) a megfigyeléseket (például közúti balesetek), j oszlopaiban pedig a balesetek kategóriakimeneteit (például könnyű, súlyos, halálos sérülés), vagyis a változókat tartalmazza. Itt adott változó (a sérülés kimenete) egyes kategóriáihoz való tartozást [0, 1] kimenetű ún. indikátor-/dummy változókkal definiáljuk. Ha az X sérülés változó kimeneteinek a száma például 3, akkor az alkalmazandó indikátorváltozók száma is 3.

Az MCA-módszer a megfigyelés/változó adatállományt úgy kezeli, mint egy egyszerű kontingenciatáblát, ahol a sorokat az individuális megfigyelések (például közúti balesetek) képezik, oszlopait pedig az indikátorváltozók alkotják. Az indikátorváltozókat az MCA szolgáltatja, annak ismeretében, hogy a kategóriakimeneteknek mennyi a száma adott változó esetén. Mivel az MCA a síkbeli ábrázolás érdekében az (i, j) kategóriákhoz koordinátákat rendel, amelyek az eredetileg kategóriakimenetű változókat folytonos skálára transzformálják át. Természetesen az eredetileg folytonos skálájú változó is diszkretizálható, és így alkalmazható rá az MCA-módszer. Az MCA-módszer bemutatását szolgáló adatállomány személyi sérüléssel járó $n = 50$

közúti baleseteken alapul, ahol a változók és kimeneteik rendre: *kár* (*kis_1, közepes_2*, *nagy_3*), *sérülés* (*könnyű_1, súlyos_2, halálos_3*), *sebesség* (*lassú_1, közepes_2*, *száguldó_3*), *légzsák* (nemnyit_0, nyit_1), *biztonsági öv* (*kikapcsolt_0*, *bekapcsolt_1*).

6. parancssor

```
# BeginCopyCA
   # Épületek tervtípusai, falszerkezetük, CA-előrejelzés: kiegészítő sorok és oszlopok elhelyezése a síkban
      library(FactoMineR)   # Betölti a szükséges FactoMineR programcsomagot
      Terv <- read.table("F:/Tervek.csv", header=TRUE, sep=";", na.strings="NA", dec=".",
         strip.white=TRUE, row.names=1)
      Terv # Az adatok megtekintése
      Terv.CA <- CA(Terv, col.sup=5:6, row.sup=6:7) # Az 5–6. oszlopok és a 6–7. sorok kiegészítő jellegűek
      summary(Terv.CA)
      Terv.CA # A rendelkezésre álló eredményobjekumok listázása
      Terv.CA$row$inertia # Az inerciák megjelenítése (megegyezik az oszlopokéval!)
      Terv.CA$call$marge.col # Az oszlopok marginális megoszlásai, súlyai
      Terv.CA$call$marge.row # A sorok marginális megoszlásai, súlyai
      Asszociacio.Terv <- chisq.test(Terv[,1:4]) # A χ² -teszteredmények definiálása
      Asszociacio.Terv$stat # Hozzáférés a χ² -statisztika értékéhez
      round(Asszociacio.Terv$residuals,2) # A függetlenség esetétől való relatív reziduális gyakoriságok: negatív és pozitív előjelű asszociációk
      round(100*Asszociacio.Terv$residuals^2/Asszociacio.Terv$stat,2) # Az egyes cellák százalékos hozzájárulásai a teljes χ² -inerciához
      plot(Terv.CA, axes=1:2)
      plot(Terv.CA, axes=c(1,3)) # Az 1. és 3. dimenziókat ábrázoljuk
      plot(Terv.CA ,invisible="col") # Az oszlopkategóriák nem láthatók az ábrán
      plot(Terv.CA ,invisible="row") # A sorkategóriák nem láthatók az ábrán
# EndCopyCA
# Közúti balesetek többszörös korrespondenciaanalízise:
```

7. parancssor

```
# BeginCopyMCA
   # Adatbeolvasás: az adatok numerikusak
      Baleset <- read.table("F:/BalesetMCA.csv", header=TRUE, sep=";", na.strings="NA",
         dec=".", strip.white=TRUE) # Az MCA kategóriakimenetű skálát igényel, ami a következő „faktorkonvertálással” érhető el, ciklusba szervezve:
      for (i in 1 : ncol(Baleset)) { Baleset[,i]<-factor(Baleset[,i]) }
      Baleset # Az adatállomány megjelenítése
      summary(Baleset) # Az adatállomány leírása: a skála már nominális, a lehetséges kimenetek a gyakoriságaikkal
```

```
    library(FactoMineR) # A szükséges csomagok betöltése
    BalesetMCA<-MCA(Baleset, quali.sup=1:2, level.ventil=0) # Az 1 a „Kár” változó
      „supplementary”, asszociálandó, nem alakítja az MCA-térképet, # Csak ábrázolásra kerül
    barplot(BalesetMCA$eig[,2],names=paste("Dim",1:nrow(BalesetMCA$eig))) # A sajátérté-
      kek hisztogramja
    summary(BalesetMCA) # Az MCA-adatállományon végzett CA szokásos eredményei
    BalesetMCA$ind$coord # Az alkalmazott kategóriaskálák folytonos koordinátákká transz-
      formálása, például főkomponens-analízis input adataként
# EndCopyMCA
```

2.7. Klasszifikációs feladatok

Kategóriakimenetű célváltozó előre definiált kategóriái közül a legvalószínűbb előrejelzése – prediktorváltozók értékeinek az ismeretében – klasszifikálási feladatot jelent. A kategóriák száma lehet kettőnél több, és a prediktorváltozók száma is lehet egynél több.

A feladat a kategóriák a priori, szubjektív valószínűségeit átvezetni – a prediktorváltozók értékei alapján – a posteriori feltételes valószínűségi struktúrába, ahol a prediktor *X* értékek jelentik a feltételt, és a maximális posterior valószínűség jelzi a keresett, legvalószínűbb kategóriát.

A priori modell – a prediktor *X* változók értékeinek az ismerete nélküli konstans előrejelzésű null modell, mely tipikusan a célváltozó kategóriáinak mintán belüli relatív gyakoriságait alkalmazza konstans klasszifikációként, de lehet más, szakmailag megalapozott szakértői becslés is. A prediktorváltozók körének definiálása előbb egy induló szakmai *X* változólistát, majd a listán belüli statisztikai, algoritmikus szelekciós technikát igényel.

A priori valószínűségekből a posteriori valószínűségekbe átmenet alapvető módszertani eszközei – több más között – egyfelől a bayesi elv alkalmazása, másfelől a logisztikus regresszió módszere, majd a döntési fák és a neurális hálózatok.

Bayes-klasszifikáció

A Bayes-módszer adatállománya példánkban a Budapesti Értéktőzsde 76 tőzsdetag brókercége egy adott időpontra vonatkozóan. Az adatok a BET.csv fájlban foglaltak, ahol a változók közgazdasági tartalma rendre: probléma (*Problem*), O.K., gyanús (*Gyanus*), csőd (*Csod*), jövedelmezőség (*Jovedelmezoseg*), forgóeszközarány (*Forgoeszkozarany*), sajáttőkearány (*Sajattokearany*), adósságszint (*Adossagszint*), eszközök forgási sebessége (*Eszkozok forgasi sebessege*), likviditás (*Likviditas*).

A diszkrét célváltozó a *Problem*, skálája *Problem* = [0, 1, 2], amely kimenetek azonosítására rendre az *O.K.*, *Gyanus*, *Csod* indikátor [0, 1] dummy változók „1”

értékei utalnak. *O.K.* = 1: nincs fizetési probléma a céggel, *Gyanus* = 1: a „homályos zónába" tartozik, míg *Csod* = 1: a brókercég csődbe ment. Sorsuk szerinti gyakorisági megoszlásaik: 57, 11 és 8 darab. A többi változók prediktorok a *Problem* célváltozó szintjének az előrejelzéséhez. A Bayes-klasszifikáció feltételezi a prediktorok normalitását, és alapesetben kvadratikus jellegű a modell. Ha a kategóriák szóródási (kovariancia-) mátrixai egyezőségének a hipotézise elfogadható, akkor – numerikus egyszerűsítési okból – a lineáris modell is alkalmazható, egyébként nem. A Bayes-klasszifikációt tehát meg kell, hogy előzze egy normalitási és egy homogenitási teszt.

A következő R kód tartalmazza mind a kvadratikus, mind a lineáris klasszifikációt, a MASS-package „qda" és „lda" függvényét alkalmazva.

8. parancssor

```
# BeginCopyBayes
    BET <- read.table("F:/BET.csv", header=TRUE, sep=";", na.strings="NA", dec=".",
      strip.white=TRUE, row.names=1) # Az adatállomány betöltése
    show(BET) # Az adatállomány printelése monitorra
  # A Problem, Csod, Gyanus és O.K. változók kategóriakimenetűvé (faktortípusra) transzformálása, ciklusban:
    BET <- within(BET, {
      Problem <- as.factor(Problem)
      Csod <- as.factor(Csod)
      OK <- as.factor(OK)})
    summary(BET) # A transzformált adatállomány felülírta az eredetit és leíró statisztikái
    library(MASS) # Betölti a szükséges MASS-package programcsomagot
  # Kvadratikus klasszifikáció, az eredményobjektum elnevezése: BET.result:
    BET.result <- qda(Problem ~ jovedelem+feszkar+stokear+adossag+eszkfseb+likvid,
      data=BET, prior = c(1,1,1)/3,CV = TRUE )
    # A prior valószínűségek egyenlők, és egybeesnek a relatív gyakoriságokkal, egyaránt 1/3
    # A $ jel a BET.result objektum listaelemeit publikálja a $ jel jobb oldalán szerepelve:
    Klasszifikacio <- BET.result$class
    matrix(Klasszifikacio) # Oszlopformában írja ki a sorvektort
    Posterior <- BET.result$post # A Bayes-posterior valószínűségek számítása és objektumba foglalása
    round(Posterior , 3 ) # 3-tizedesre kerekíti a posteriorokat
    Klassz.matrix <- xtabs(~BET$Problem + Klasszifikacio, data=BET) # A klasszifikációs mátrix számítása, tabulálása
    Klassz.matrix <- matrix(Klassz.matrix, 3) # A klasszifikációs mátrix (3, 3) rendű mátrix formába konvertálva
    Klassz.matrix # A klasszifikációs mátrix gyakoriságainak és relatív gyakoriságainak megjelenítése
      round(100*Klassz.matrix / sum(Klassz.matrix),2)
      round(100*prop.table(Klassz.matrix,margin=1),2)
      round(100*prop.table(Klassz.matrix,margin=2),2)
  # Lineáris klasszifikáció
```

```
    BET.result <- lda(Problem ~ jovedelem+feszkar+stokear+adossag+eszkfseb+likvid,
      data=BET,prior = c(1,1,1)/3,CV = TRUE )
    BET.result
    Klasszifikacio <- BET.result$class
    matrix(Klasszifikacio) # Oszlopformában írja ki a klasszifikációkat
    Posterior <- BET.result$post
    round(Posterior , 3 )
    Klassz.matrix <- xtabs(~BET$Problem + Klasszifikacio, data=BET)
    Klassz.matrix
# EndCopyBayes
```

A logisztikus regresszió klasszifikációs alkalmazása

A klasszifikáció során a magyarázó változók szintjeinek ismert *X* kombinációja – ún. kovariánsa – mellett kalkuláljuk az *Y* célváltozó kategóriáinak a feltételes valószínűségeit, és az *i* megfigyelési egységet a legvalószínűbb kategóriához rendeljük hozzá.

Csődkockázati szempontból gazdasági vállalkozásokat minősítünk mérlegük és eredményük alapján, a döntés pénzügyi hatásaira is tekintettel. E feladat egyféle megközelítése a Bayes-elv említett alkalmazása, de egy másik alapvető módszer a logisztikus regresszió.

Ha az eredményjellegű *Y* (dependent, response) változó bináris, vagyis két lehetséges kimenete „*Y* = 1” és „*Y* = 0”, „Igen/Nem”, akkor bináris vagy dichotom logisztikus regresszióról beszélünk. A függő változó eloszlásának az ismeretében (független, véletlen mintavétel esetén) *Y* Bernoulli-folyamatot követ, így a logisztikus regresszió paramétereinek becslésére az ML-módszer (maximum likelihood – legnagyobb valószínűség) kézenfekvően kínálkozik. Ha a függő változónak kettőnél több kimenete van, akkor multinominális regresszióról beszélünk.

Az adatállomány változói pénzügyi arányszámok, közgazdasági tartalmuk rendre: *csőd*, *eszközök forgási sebessége*, *sajáttőke arány*, *bonitás*, *jövedelmezőség*, *forgóeszközarány*, *likviditás*. Minden *X* prediktorváltozó 100-zal szorzott formában szerepel az adatállományban, a százalékpontos *X* változásra megfelelő pillanatnyi *Y* növekedési ütem százalékos értelmezése érdekében. A célváltozó a *csőd*: igen/nem esemény előrejelzése. Az adatállomány neve „Csod.csv”. Az R függvények alapvetően a „glm” package rutint alkalmazzák.

9. parancssor

```
# BeginCopyBinárisLogisztikus
    Csod <- read.table("F:/Csod.csv", header=TRUE, sep=";", na.strings="NA", dec=".",
      strip.white=TRUE)
    summary(Csod) # Leíró adatok megtekintése
  # A logisztikus regresszió modellje, binary case, a glm függvény alkalmazásával: H1 modell vs.
    H0 modell becslése
```

```
    Csod.glm.binomial.H1 <-
      glm(Csod~jovedelm100+feszkar100+stokear100+bonitas100+eszkfseb100+likvid100,
      data=Csod, family=binomial) # A H1 modelleredmények tárolása
    summary(Csod.glm.binomial.H1) # A H1 modelleredmények összefoglalása
    beta <- coef(Csod.glm.binomial.H1) # A H1 koefficiensek tárolása
    exp(beta) # A csőd odds-ratio számítása
    Csod.glm.binomial.H0 <- glm(Csod ~ jovedelm100+feszkar100, data=Csod,
      family=binomial) # A H0 modelleredmények
    summary(Csod.glm.binomial.H0) # A H0 eredmények összefoglalása
    beta <- coef(Csod.glm.binomial.H0) # A H0 koefficiensek tárolása
    exp(beta) # H0 szerinti odds-ratio ráták
  # Nested-modellek összevetése χ^2 -teszt alapján
    anova(Csod.glm.binomial.H0,Csod.glm.binomial.H1) # Varianciaanalízis: null-modell v.s. je-
      len modell
    # Cut-value szelektálás: klasszifikációs mátrixok generálása
    Posterior <- predict(Csod.glm.binomial.H1, newdata=Csod, type="response")
    Posterior.cut.0.01 <- as.numeric(Posterior>0.01) # Cut-érték = 0,01
    table(Csod$Csod, Posterior.cut.0.01 )
    Posterior.cut.0.03 <- as.numeric(Posterior>0.03) # Cut-érték = 0,03
    table(Csod$Csod, Posterior.cut.0.03 )
    Posterior.cut.0.05 <- as.numeric(Posterior>0.05) # Cut-érték = 0,05
    table(Csod$Csod, Posterior.cut.0.05 )
    Posterior.cut.0.1 <- as.numeric(Posterior>0.1) # Cut-érték = 0,10
    table(Csod$Csod, Posterior.cut.0.1)
  # Interakció(k) alkalmazása
    Csod.glm.binomial.H0.interakcio <- glm(Csod ~ jovedelm100 +feszkar100 +
      jovedelm100:feszkar100, data=Csod, family=binomial) # ":" az interakció képzése, ope-
      rátora
    summary(Csod.glm.binomial.H0.interakcio)
    beta <- coef(Csod.glm.binomial.H0.interakcio)
    exp(beta)
    anova(Csod.glm.binomial.H0,Csod.glm.binomial.H0.interakcio)
# EndCopyBinárisLogisztikus
  # A multinomiális, több kategóriakimenetű logisztikus regresszió alkalmazása
```

10. parancssor

```
# BeginCopyMultinomiálisLogisztikus: A multinomiális logisztikus regresszió alkalmazása
    BET <- read.table("F:/aData/Cross/BET.csv", header=TRUE, sep=";", na.strings="NA",
      dec=".", strip.white=TRUE, row.names=1)
    library(nnet) # A szükséges package beolvasása
    library(MASS) # A szükséges package beolvasása
    BET.mnom.H1 <- multinom(Problem ~
      jovedelem+feszkar+stokear+adossag+eszkfseb+likvid, data=BET)
    summary(BET.mnom.H1)
    BetaH1<-coef(BET.mnom.H1)
    OddsRatiosH1<-exp(BetaH1)
```

```
    t(OddsRatiosH1)
    BET.mnom.H0 <- multinom(Problem ~ jovedelem+feszkar, data=BET)
    summary(BET.mnom.H0)
    BetaH0<-coef(BET.mnom.H0)
    OddsRatiosH0<-exp(BetaH0)
    t(OddsRatiosH0)
# EndCopyMultinomiálisLogisztikus
```

Egzakt logisztikus regresszió

Ha az eredményjellegű (dependent, response) változó bináris, vagyis két lehetséges kimenete „1” és „0”, „igen/nem”, a függő változó eloszlásának az ismeretében a logisztikus regresszió paramétereinek becslésére az ML-módszer alkalmas, amely eljárás viszont kedvező mintavételi következtetési tulajdonságai (minimum variancia, konzisztencia) aszimptotikusan, nagymintás esetben érvényesülnek. Azonban például a vállalkozások csődhelyzet-klasszifikálása a kismintás következtetés tipikus esete lehet, hiszen a csődesemény bizonyos tevékenységi körökben, iparágakban relatíve ritka jelenség. Tehát egy szakágazati szintű „csődmodell” kismintás becslése kényszerű adottság. Jelen szakasz alapvető célja, hogy a csődkockázat mérése kapcsán a logisztikus regresszió ML becslési problémáira felhívja a figyelmet és kezelésükre megfelelő R-kódot javasoljon.

A feltétel nélküli ML-eljárás alkalmazása szempontjából alapvető probléma a kiegyensúlyozatlan minta esete, melyben (tekintet nélkül a mintanagyságra) relatíve nagyon alacsony (akár 5 százalék alatti) a csődesemények aránya, másfelől a szeparált minta esete, melyben a csődesemény egyértelműen a magyarázó változó egy adott szegmenséhez, a komplementer „működő” események pedig egy egyértelműen elhatárolt másik szegmenséhez tartoznak. Míg az előbbi esetben van egyedi ML-megoldás, de az torzított és magas mintavételi varianciájú, addig az utóbbiban nem is létezik ML-megoldás. A harmadik lényeges problémát az okozza, mikor a priori információnk van a csődesemények arányáról a sokaságban (ez az információ például a nemzetgazdaságban rendelkezésre áll), és ez az arány jelentősen eltér a megfelelő mintabeli aránytól, további torzítást okozva a paraméterek becslésében.

A ritka „1” esemény kezelését a klasszikus logisztikus regresszió aszimptotikus eredményeinek megfelelő korrekcióval történő alkalmazása, vagy a csőd-/működési események egzakt permutációin alapuló ún. ELR (egzakt logisztikus regresszió) (nem aszimptotikus) egyaránt szolgálja. Az ELR-eljárás a regressziós paraméterek elégséges statisztikáinak az egzakt, feltételes, permutációs eloszlásán alapuló módszertana. Mikor az aszimptotikus ML-becslés nem létezik, az ELR-módszer használatával akkor is következtetni tudunk a regressziós paraméterekre.

A következő példa csődbe ment és működő gazdasági vállalkozások klasszifikálását illusztrálja.

11. parancssor

```
# BeginCopyExactLogistic
    Csod <- read.table(text = "
    jovedelem adossag csod gyakorisag
    0 1 1  2
    0 1 1  4
    0 0 1  2
    0 1 0 10
    1 1 0 40
    1 1 1  6
    1 1 0 20
    1 0 0 12
    1 1 0  2
    1 0 0  2 ", header = TRUE)
    Csod
    library(elrm)
    Csod <- Csod[rep(1:nrow(Csod), Csod$gyakorisag), -4]
    x <- xtabs(~csod + interaction(jovedelem, adossag), data = Csod)
    x
    cdat <- cdat <- data.frame(jovedelem = rep(0:1, 2), adossag = rep(0:1, each = 2),csod =
      x[2,], ntrials = colSums(x))
    cdat
    m.Csod <- elrm(formula = csod/ntrials ~ jovedelem+adossag, interest =
      ~jovedelem+adossag, iter = 22000, dataset = cdat, burnIn = 2000)
    summary(m.Csod)
  # Egy alternatív struktúrájú, kontingenciatábla input adatállomány alkalmazása
    Csod <- read.table(text = "
    jovedelem adossag csod gyakorisag
    0 0 2  2
    1 0 0 14
    0 1 6 16
    1 1 6 68 ", header = TRUE)
    Csod
  #Főbb exact logistic regressziós modellek (package) betöltések:
    #library(LogisticDx)
    #library(logistf)
    #library(clogitL1)
    #library(clogitboost)
    #library(elrm)
# EndCopyExactLogitstic
```

2.8. Döntési fák

A döntési fa (CART [classification and regression trees – osztályozási és regreszsziós fák]) egy fastruktúra-alapú klasszifikációs modell. Az eljárás mind exploratív, mind konfirmatív céllal alkalmazható. A döntési fa célja a mintát X prediktorváltozók alapján diszjunkt alcsoportokra bontani úgy, hogy adott X csoport az előrejelzendő Y tekintetében minél homogénebb legyen. A létrejött alcsoport megnevezése *node* (csomópont), definíciója pedig az X_1, X_2,…, X_p tengelyeken kialakult osztályok sorozata. E szegmensek vagy kettévágások egymást követő sorozatával vagy kettőnél több osztópont egyidejű elhelyezésével alakulnak ki X mentén. E hasítás (split) a skála bontását jelenti annak mintabeli x kimeneteinél: ordinális esetben minden osztópont alsó és felső szegmensre, nominális skála esetén pedig az összes lehetséges kimenetkombináció közül valamelyikre bont. A végső hasítás nem csak vágások sorozataként, hanem egy kezdeti (sűrű) felosztás fokozatos összevonásaként is megvalósítható. Az előrejelzés az ún. „terminált (leafe) node" alapján történik, melyet már nem hasítunk tovább. Minden terminált node nyújt egy önálló, nodespecifikus előrejelzési szabályt a node X definíciója ismeretében. Kategóriakimenetű Y esetén a leggyakoribb Y kimenet, folytonos Y esetén pedig a node részátlaga az előrejelzés. Egy előrejelzés azon múlik tehát, hogy az X megfigyelést lecsorgatva a fán, az melyik terminált node eleme lesz. Ha a fa növekedése csak kettévágásokon alapszik, akkor a fa binárisan növekvő típusú. A végső cél tehát minél tisztább node elérése kategóriakimenetű Y tekintetében, illetve minél homogénebb node elérése folytonos Y tekintetében, minél kevesebb node alkalmazásával. Egy node annál tisztább, minél egyenlőtlenebb kimeneteiben, és annál zavarosabb, minél inkább keverednek benne Y kimenetei.

A hasítás megvalósítása az X prediktor mérési skáláján annak ordinális vagy nominális jellege által meghatározott. Bináris módon vágva: *a*) ordinális skálán ha X kimenetei: {20, 30, 44, 50, 66, 72}, akkor a vágások alsó szegmensei az osztályközepek, rendre: ≤ 25, ≤ 37, ≤ 47, ≤ 53, ≤ 69; *b*) ordinális skálán ha X kimenetei {*könnyű, súlyos, halálos*} akkor a vágások alsó_felső szegmensei rendre: könnyű_nem könnyű, nem halálos_halálos; *c*) Nominális skálán, ha X kimenetei {*falu*, *város*, *Budapest*} akkor minden vágás: falu_nem falu, város_nem város, Budapest_nem Budapest.

Bármilyen jellegű a döntési fa, általános érvényű mozzanatai a következők. Ha a mintaméret megengedi, a megfigyeléseinket, eseteinket egy ún. learning (tanuló) vagy training set részre és egy ún. teszt validation set részre bontjuk. A fát (a döntési szabályt) a tanulómintán építjük fel, és az előrejelzését a tesztmintán ellenőrizzük. A „gyökér" root-node mindig a csoportosítandó, de még nem csoportosított minta. A növekedési folyamatot hasítási kritérium kormányozza. A fa túlnövekedését ésszerű leállási feltételek betartásával felügyeljük. A fa pruningolása (nyesése) az irreleváns

node-ok elhagyásával egyszerűsíti a döntési szabály alkalmazását előrejelzéskor. A hiányzó értékek (missing value) megfelelő kezelése lehetővé teszi, hogy releváns prediktorváltozó ne maradjon ki.

A következő esettanulmány 1669 vállalkozás csődbement/nem kimenetelét vizsgálja pénzügyi indikátorok alapján. Az adatállomány a már korábban megismert Csod.csv adatok, ahol az indikátorok rendre: csőd, eszközök forgási sebessége, sajáttőkearány, bonitás, jövedelmezőség, forgóeszközarány, likviditás. Minden X prediktor 100-zal szorzott formában szerepel az adatállományban. A célváltozó a csődesemény (igen/nem) előrejelzése.

12. parancssor

```
# BeginCopyCRTcsőd
    library(rpart) # library(party) alternatív csomag betöltése
    Csod <- read.table("F:/Csod.csv", header=TRUE, sep=";", na.strings="NA", dec=".",
      strip.white=TRUE)
    Csod[,1] <- factor(Csod[,1])
    summary(Csod)
    Csod.tree.model <- rpart(
      Csod~eszkfseb100+stokear100+bonitas100+jovedelm100+feszkar100+likvid100,
      data=Csod, maxcompete=5)
    Csod.tree.model
    summary(Csod.tree.model)
    Csod.pred<-predict(Csod.tree.model,newdata=Csod)
    summary(Csod.pred)
# EndCopyCRTcsőd
```

2.9. Latens változók analízise: SEM

A SEM egymással korreláló változók kovarianciáit egy hipotetikus, latens változók kapcsolatait definiáló többegyenletes modell paramétereiből vezeti le. Előbb becsüljük a paramétereket a mintabeli manifeszt kovarianciák alapján, majd ezen empirikus kovarianciákat a hipotetikus modellből levezetett megfelelőikkel szembesítjük. Az empíria és a hipotézis közötti távolság alapján megítéljük a modell releváns vagy irreleváns voltát, majd a releváns modell illeszkedését a mintához, illetve két jól illeszkedő modell közül a paramétertakarékosabbat preferáljuk. A hipotetikus modellt strukturális egyenletek rendszere fogalmazza meg. Itt mind az endogén, mind az exogén változók lehetnek közvetlenül megfigyelhető manifeszt indikátorok és latens jellegű, közvetlenül nem megfigyelhető, de mérhető latens faktorok is.

A modell $\boldsymbol{\theta}$ strukturális paramétereinek a becsléséhez annyi (nemlineáris) egyenlet áll a rendelkezésünkre, ahány (nem duplikát) kovarianciát (a varianciákat is bele-

értve) modellezünk a p számú manifeszt indikátor egymásközti, (p, p) rendű $\boldsymbol{\Sigma}$ elméleti kovarianciamátrixában: $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}(\boldsymbol{\theta})$, ahol a $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ függvény a paraméterek és a $\boldsymbol{\Sigma}$ kovarianciamátrix elemeinek a kapcsolatát reprezentálja.

A feladat a paraméterek becslése a mintabeli **S** kovarianciamátrix alapján, majd a becsült $\hat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}(\hat{\boldsymbol{\theta}})$ kovarianciamátrix illeszkedésének a jellemzése.

A kovariancia-struktúraanalízis többek között olyan témaköröket fed le, mint a konfirmatív faktoranalízis, „path" analízis, a látens változós strukturális egyenletek becslése. Legegyszerűbb esetként a lineáris regressziós modellt is magában foglalja.

A latens változós modell definiálása két alapvető lépésre bontható:

Elsőként definiáljuk a latens, közvetlenül nem megfigyelhető változókat azzal, hogy mely közvetlenül megfigyelhető manifeszt változók hatására mozdulnak. Ez képezi a modell mérési blokkját. A latens változók birtokában megfogalmazzuk a hipotetikus regressziós-korrelációs kapcsolatrendszert a latens változók között. Végül becsüljük a koefficienseket, és teszteljük a paraméterek szignifikanciáját.

Az esettanulmány személyautó márkák gyári jellemzői közötti ok-okozati kapcsolatrendszerét vizsgálja, rendre: *hengerűrtartalom* (cm^3), *lóerő* (LE), *végsebesség* (km/óra), *gyorsulás 100 km/óra sebességre* (mp), *fogyasztás 90 km/óra átlagsebesség mellett* (l/100 km), *fogyasztás 130 km/h átlagsebesség mellett* (l/100 km), *fogyasztás a városban* (l/100 km). A változók azonosítója az R programban ennek megfelelő.

A latens változókat tartalmazó SEM-modellek elemzésére más R package is rendelkezésre áll, de jelen cikk a „lavaan" package használatát javasolja. Tekintettel a módszer komplexitására, a „lavaan package" lavaan(.) függvényének az argumentumát részletesen megadjuk, amiben az opciókat a felhasználó kapcsolhatja.

13. parancssor

```
# BeginCopyAutókSEM
    Autok <- read.table("F:/Autok.csv", header=TRUE, sep=";", na.strings="NA", dec=".",
      strip.white=TRUE)
    Autok
    summary(Autok)
    library(lavaan)
    autok.modell <- '
    Fogyasztas =~ f90 + f130 + vf
    Teljesitmeny =~ loero + gy100 + vegseb
    gy100 ~~ vegseb
    f90 ~~ f130
    Fogyasztas ~ Teljesitmeny + henger + tomeg
    Teljesitmeny ~ Fogyasztas '
    fit <- lavaan( model = autok.modell, data = Autok, model.type = "sem", meanstructure =
      "default", int.ov.free = FALSE, int.lv.free = FALSE, fixed.x = "default", orthogonal = FALSE,
```

```
        std.lv = FALSE, auto.fix.first = TRUE, auto.fix.single = FALSE, auto.var = TRUE,
        auto.cov.lv.x = TRUE, auto.cov.y = FALSE, auto.th = FALSE, auto.delta = FALSE, std.ov =
        FALSE, missing = "default", ordered = NULL, sample.cov = NULL, sample.cov.rescale =
        "default", sample.mean = NULL, sample.nobs = NULL, ridge = 1e-05, group = NULL,
        group.label = NULL, group.equal = "", group.partial = "", cluster = NULL, constraints = "",
        estimator = "default", likelihood = "default", information = "default", se = "default", test
        = "default", bootstrap = 1000L, mimic = "default", representation = "default", do.fit =
        TRUE, control = list(), WLS.V = NULL, NACOV = NULL, zero.add = "default",
        zero.keep.margins = "default",start = "default",slotOptions = NULL, slotParTable = NULL,
        slotSampleStats = NULL, slotData = NULL, slotModel = NULL, verbose = FALSE, warn =
        TRUE, debug = FALSE )
    summary(fit, standardized=TRUE, fit.measures=TRUE, rsquare=TRUE)
  # További fontosabb eredmények különálló elérései, lehívásai:
    parTable(fit)
    vcov(fit)
    predict(fit)
    logLik(fit)
    update(fit)
    inspect(fit)
    parameterEstimates(fit, ci=TRUE, standardized=TRUE)
    Estimates <- parameterEstimates(fit, ci=TRUE, standardized=TRUE)
    subset(Estimates, op == "=~")
    MI <- modificationIndices(fit)
    subset(MI, mi > 0)
# EndCopyAutókSEM
```

3. Összefoglalás

Jelen tanulmány egy olyan statisztikai modellalkalmazási útmutató, mely az R project program használatára épül. Az R nyelv segítségével saját fejlesztésű és szabad rendelkezésű (open source) statisztikai szoftver hozható létre mint „termék”. A cikk alapvető, elengedhetetlen R szintaktikai ismeretekkel indul, majd empirikus adatok (esettanulmányok) során tárgyalja adott módszerek működésének R módját. A módszereket a többváltozós statisztika témakörből választottuk, keresztmetszeti adatokra alkalmazva, ezek rendre: az általánosított lineáris modell, a főkomponens-analízis, a kanonikus korrelációszámítás, a klaszteranalízis, a korrespondenciaanalízis, a döntési fa, a logisztikus regresszió, a lineáris és kvadratikus klasszifikáció, a diszkriminanciaanalízis és a latens változókat is tartalmazó SEM-modell.

A cikk alfejezeteinek bevezetői ismertető, értelmezési, útmutató jellegűek, a lényegi mondanivaló a sorszámozott parancssorokban van elhelyezve, ezek mindegyike egy-egy R kód, melyekhez a vonatkozó statisztikai adatokat a tanulmány internetes

Mellékletében csatoltuk, mert elérésük nélkül (esetleg helytelen útvonal megadás miatt) adott programrész értelemszerűen nem fut le. Továbbá a Mellékletből lehet a Begin…End parancssorok R kódjait másolni. Az esettanulmányok eredményeinek a megtekintése igényli az R konzol párhuzamos használatát a tanulmány olvasása közben!

Irodalom

AGRESTI, A. [2007]: *An Introduction to Categorical Data Analysis*. John Wiley & Sons. New York. http://dx.doi.org/10.1002/0470114754

BILDER, C. R. – LOUGHIN, T. M. [2014]: *Analysis of Categorical Data with R*. Text in Statistical Science. Chapman & Hall/CRC. Boca Raton.

BOLLEN, K. A. [1989]: *Structural Equations with Latent Variables*. John Wiley & Sons. New York. http://dx.doi.org/10.1002/9781118619179

BRYAN, F. J. – MANLY, J. A. – NAVARRO A. [2016]: *Multivariate Statistical Methods: A Primer*. Fourth Edition. Chapman & Hall/CRC. London.

CHAMBERS, J. M. [2016]: *Extending R*. The R Series. Chapman & Hall/CRC. London.

CORNILLON, P.-A. – GUYADER, A. – HUSSON, F. – JEGOU, N. – JOSSE, J. – KLOAREG, M. – MATZNER-LOBER, E. – ROUVIER, L. [2012]: *R for Statistics*. Chapman & Hall/CRC. London.

CRAWLEY, M. J. [2002]: *Statistical Computing. An Inrtroduction to Data Analysis Using S-plus.* John Wiley & Sons. New York.

DARÓCZI G. – TÓTH G. [2013]: Felhőtlen statisztika a felhőben. *Statisztikai Szemle*. 91. évf. 11. sz. 1118–1142. old.

DARÓCZI G. [2016]: Alkalmazott statisztika? R! *Statisztikai Szemle*. 94. évf. 11–12. sz. 1106–1122. old. https://doi.org/10.20311/stat2016.11-12.hu1108

DUNN, G. – EVERITT, B. S. – PICKLES, A. [1993]: *Modelling Covariances and Latent Variables Using EQS.* Chapman & Hall/CRC. London.

FOSTER, I. – GHANI, R. – JARMIN, R. S. – KREUTER, F. – LANE, J. [2016]: *Big Data and Social Science: A Practical Guide to Methods and Tools*. Statistics in the Social and Behavioral Sciences. Chapman & Hall/CRC. London.

FOX, J. [2016]: *Using the R Commander: A Point-and-Click Interface for R*. The R Series. Chapman & Hall/CRC. London.

HAJDU, O. [2002]: Category selection and classification based on correspondence coordinates. *Hungarian Statistical Review*. Vol. 80. Special Number 7. pp. 103–126.

HAJDU O. [2003]: *Többváltozós statisztikai számítások*. Statisztikai módszerek a társadalmi és gazdasági elemzésekben. Központi Statisztikai Hivatal. Budapest.

HAJDU, O. [2004]: Multitrait-multimethod models for profitability indicators. *Periodica Politechnika Social and Management Sciences*. Vol. 12. No. 2. pp. 211–222.

HAJDU, O. [2006]: Exact inference on poverty predictors based on logistic regression approach, *Hungarian Statistical Review*. Vol. 84. Special Number 13. pp. 134–147.

HAJDU, O. [2009]: Poverty, deprivation, exclusion: a structural equations modelling approach. *Hungarian Statistical Review*. Vol. 87. Special Numer 13. pp. 90–102.

HORTON, N. J. – KLEINMAN, K. [2015]: *Using R and RStudio for Data Management, Statistical Analysis, and Graphics.* Second Edition. Chapman & Hall. London.

HUNYADI L. [2001]: *Statisztikai következtetéselmélet közgazdászoknak.* Statisztikai módszerek a társadalmi és gazdasági elemzésekben. Központi Statisztikai Hivatal. Budapest.

HUSSON, F. – LE, S. – PAGÈS, J. [2017]: *Exploratory Multivariate Analysis by Example Using R*. Second Edition. Computer Science & Data Analysis. Chapman & Hall/CRC. London.

KLEINBAUM, D. G. – KLEIN, M. [2002]: *Logistic Regression. A Self-learning Text*. Springer. Berlin.

KONISHI, S. [2014]: *Introduction to Multivariate Analysis: Linear and Nonlinear Modeling*. Chapman & Hall/CRC. London.

RIZZO, M. L. [2007]: *Statistical Computing with R*. The R Series. Chapman & Hall/CRC. London.

ROSSEEL, Y. [2012]: lavaan: an R package for structural equation modeling. *Journal of Statistical Software*. Vol. 48. Issue 2. pp. 1–36. https://doi.org/10.18637/jss.v048.i02

UNWIN, A. [2015]: *Graphical Data Analysis with R*. The R Series. Chapman & Hall/CRC. London.

WICKHAM, H. [2014]: *Advanced R.* Chapman & Hall/CRC. London. http://dx.doi.org/10.1201/b17487