Felhőtlen statisztika a felhőben

Daróczi Gergely

PhD-hallgató, az Easystats Ltd. vezető fejlesztője

E-mail: daroczig@rapporter.net

Tóth Gergely

PhD-hallgató, az Easystats Magyarország Kft kutatásvezetője és az MTA-ELTE-Peripato kutatási segédmunkatársa

E-mail: gergely.toth@rapporter.net

A tanulmány áttekintést nyújt az adatelemzést segítő számítógépes eszközök modernkori történetéről, majd egy magyar fejlesztésű, felhőben futó, tehát online adatelemző és riportkészítő platformot mutat be az R statisztikai programcsomag és annak kiterjesztéseire építve. A program lehetőséget nyújt a hagyományos adatelemző eljárások alkalmazására és egyedi, szöveges riportok készítésére is, amelyet egy térbeli modelleken alapuló esettanulmányon keresztül ismertetnek a szerzők.

TÁRGYSZÓ:
Szoftver.
Adatelemzés.
Riport/jelentés.

Az általánosan alkalmazott statisztikai eljárások köre, illetve azok robusztussága, pontossága sokat változott, fejlődött a statisztika több száz éves története során. Természetesen ehhez hozzájárult a matematikai, valamint az egyéb elméleti kutatásokon kívül a számítások módjának átalakulása is. Míg korábban különböző segédtáblázatok és a logarléc, majd később a számológép bővítette a statisztikusok eszköztárát, napjainkban már többnyire számítógépek végzik – előre meghatározott algoritmusok szerint – a számításokat. Ez a változás azt is maga után vonta, hogy a felhasználó nem feltétlenül ismeri, sőt legtöbb esetben nem is kívánja maga teljességében megismerni az alkalmazott eljárások elméleti hátterét, illetve működési elvét, hanem elégséges számára a már implementált algoritmus megfontolt kiválasztása, majd az eredmények értelmezése. Az említett változásoknak megfelelően már igen korán, a számítógépek elterjedésével párhuzamosan megjelentek olyan statisztikai szoftvercsomagok, amelyek nagyban segítik a statisztikai döntések előkészítését. Sőt, egyes programok különböző „varázslókat" és programsegédeket is felajánlanak a felhasználóknak, hogy már az algoritmusok kiválasztásában is eligazítást nyújtsanak.

Mindehhez természetesen az egyébként egyre olcsóbbá váló hardverháttér és azon kívül a programok olykor igen busás áron megvásárolható licence is szükséges. Napjainkban azonban a mobil eszközök (okostelefonok, tabletek) elterjedésével és a felhasználók párhuzamos eszközhasználatával ezen programok működtetése több problémát is felvet:

– Hogyan futtatható statisztikai program az alacsony energiafogyasztásra optimalizált mobil készülékeken?

– Hogyan érhetők el a munkahelyi gépen készült elemzés részletei a mobil eszközök segítségével?

– Hogyan lehetséges kollaboratív munka egy olyan szoftverrel, amely egy gépen fut?

Tanulmányunkban egy tervezett, ezen kérdésekre válaszolni tudó, online (divatos szóval a közösségi hálózatokra is opcionálisan támaszkodó, ún. Web2-es) statisztikai programcsomagot mutatunk be a Rapportert.

1. A statisztikai elemző eszközök fejlődéstörténete

A statisztika több száz éves történetében jelentős mérföldkövet jelentett előbb a számológépek megjelenése, majd a számítástechnika hihetetlen mértékű, máig ren-

dületlenül tartó fejlődése. A továbbiakban először a hardver-, majd a szoftverfejlődés legjellemzőbb állomásait mutatjuk be.

1.1. A számítástechnikai hardver eszközök fejlődésének legfontosabb lépcsőfokai

A mechanikus számológépek és az első programozható eszközök megjelenését követően 1937-ban az Iowai Egyetemen kezdték el az első elektronikus számítógép, az Atanasoff–Berry Computer (ABC) fejlesztését. Sajnálatos, hogy a sokáig elhúzódó fejlesztés végül (1942) nem vezetett a várt eredményre, és az ABC végül nem volt képes a tervekben meghatározott funkciók teljes körét kielégíteni.

Nem sokkal később (1946) azonban a Pennsylvaniai Egyetemen megépült az első általános célú digitális számítógép, az ENIAC (electronic numerical integrator and computer – programozható elektronikus, digitális számítógép), az Egyesült Államok Szárazföldi Hadseregének (US Army) megrendelésére. Ez az első programozható digitális számítógép a maga 30 tonnájával valóban technikai csodának számított az 1940-es években: teljesítményben lekörözte az Egyesült Államok Haditengerészte (US Navy) által finanszírozott, Harvardon fejlesztett és a programok belső tárolására alkalmatlan Mark II-t is.

Az ENIAC utódja *Neumann János* vezetésével készült. Az immáron központi vezérlőegységgel is rendelkező EDVAC (electronic discrete variable automatic computer – elektronikus diszkrét változós automata számítógép) (1949) saját memóriával ellátva újabb mérföldkövet jelentett a számítógépek történetében. Ez az állítás különösen igaz abból a szempontból, hogy Neumann publikálta kutatási eredményeit (*Neumann* [1993]), így a készülék átadásakor a világon már több hasonló elvű gép is működött (többek között a Cambridge-i Egyetemen épített EDSAC (electronic delay storage automatic calculator – első tárolt programú számítógép)).

Ekkor már a világ számos egyetemén folytak hasonló fejlesztések, sőt megjelentek az első kereskedelmi forgalomba szánt eszközök is. Az első UNIVAC (universal automatic calculator – első kereskedelmi fogalomban kapható univerzális számítógép) gépet éppen az Egyesült Államok statisztikai hivatalában vették használatba 1951-ben (*Stern* [1981]), ahol már korábban is használtak elektronikus segédeszközöket.

Ilyen történelmi jelentőségű segédeszköz volt a lyukkártya-feldolgozógép, amelyre először az 1890-as népszámlálás tabulálása során támaszkodtak. A *Herman Hollerith* nevével fémjelzett szerkezet, a későbbi IBM (International Business Machines) előfutáraként számon tartott Tabulating Machine Company nevéhez köthető (*Truesdell* [1965]). A lyukkártya-feldolgozógép hatalmas siker volt mind az Egyesült Államokban, mind Európában, azonban a bérleti díjak erős emelkedésének

hatására alternatív megoldást keresett az US Census Bureau (Egyesül Államok Népszámlálási Hivatala), előbb *Simon North*-tal sikertelenül együttműködve, majd *James Powers* és *John Mauchly* megkeresésekor az UNIVAC támogatásába kezdtek.

Az 1951-ben piacra került gép sikere sokak előtt ismert: a statisztikai hivatal mellett az amerikai hadsereg számos intézménye, illetve több piaci szereplő is használatba vehette a számítógépeket az 1950-es évek elején, többek között például az ACNielsen (piackutató cég) is szerepelt a szerződő felek között.

E mellett párhuzamosan futott az IBM ún. „mainframe" termékcsaládja is, amely már az 1950-es évektől elérhető volt, de szélesebb körű elterjedése csak az 1960-as évek második generációs termékeihez kötődik (*Renfro* [2004]). Ezen gépek (IBM 7090/7094) – amelyeket többek között a NASA (National Aeronautics and Space Administration – Nemzeti Repülési és Űrhajózási Hivatal) használt az Apollo-programokban – már a korábbi olajhűtésűekkel ellentétben levegővel hűtve meglehetősen stabil működést tettek lehetővé.

A számítógépcsalád 360-as típusát az IBM 1964-ben jelentette be. A legerősebb típus már több tízezer utasítást volt képes végrehajtani másodpercenként, illetve relatíve nagy, akár 8 MB (megabájt) memóriájával méltó módon jelentett hatalmas sikert tervezőinek. Széles körű elterjedését segítette a korábbi kompatibilitási problémák kiküszöbölése, így a programok hordozhatóvá váltak – sőt, az akkori programok akár ma is működnek bármelyik IBM zSeries termékcsaládba tartozó szerveren.

A nyugati számítógépekkel párhuzamosan hazánkban a Műszaki Egyetemen és az MTA Kibernetikai Kutatócsoportjában is folytak hasonló kutatások. A *Kozma László* építette MESZ-1 (jelfogós programvezérelt) számítógépet követően készült M3 (első magyar elektronikus számítógép) valóságos csoda volt: költsége az említett UNIVAC gép töredékét tette ki, azonban a nyolcas számrendszerben „gondolkodó" gép programozása meglehetősen nagy nehézséget jelentett.

A korábbiakban leírt első elektroncsöves és a második generációs, tranzisztoros gépekhez képest alapvető újdonságot jelentett a harmadik generációs számítógépek megjelenéséhez elengedhetetlenül szükséges integrált áramkör (integrated circuit – IC) 1958-as feltalálása, majd az 1960-as évek elején annak tömeggyártása és elterjedése.

A számítási kapacitás a korábbi sokszorosára nőtt, a negyedik generációs gépek (amelyek valójában a harmadik generáció még „integráltabb", nagyobb sűrűségű, tökéletesített változatait jelentették) az 1960-as évek IBM gépeinek teljesítményét akár százszorosan is felülmúlták.

A mikroprocesszor megjelenése, a számítógépek közötti adatátvitel lehetősége és az egyre csökkenő hardverárak egyenes utat jelentettek az ún. személyi számítógépek megjelenéséhez, amely termékcsaláddal az IBM az 1980-as évek elején debütált, és számos más vállalat követte sikerét (Xerox, Hewlett Packard, Apple, Commodore).

A számítógépek hardvertörténete ettől kezdve lehet mindenki számára ismerős, hiszen mindennapi életünk részévé váltak, és az egyre apróbb gépek lassan a leghétköznapibb cselekedetünk végrehajtása során is szerepet kapnak – míg a statisztikai számítások területén egyértelmű a kizárólagos szerepük.

Mára a mindennapi munka eszközei a hordozható számítógépek (laptop, notebook), az „okos" mobiltelefonok, a PDA-k (personal digital assistant – digitális személyi asszisztens), sőt újabban a tablet PC-k is, amelyek a legtöbb esetben közvetlen kapcsolattal rendelkeznek a világhálóval. Ezek az új fejlesztésű eszközök általában igen korlátozott erőforrásokkal rendelkeznek a mobilitás elősegítése érdekében. E probléma megoldásának eredménye, hogy a számításigényes műveletek elvégzését és a nagy mennyiségű adatok tárolását áthárítják a kiszolgáló (szerver) számítógépekre.

Így születtek meg mára az ún. „cloud" technológiára alapozott, vagy egyszerűen csak online szolgáltatások. Ezek lényege, hogy a primer adatok és az algoritmusok a kiszolgáló (védett) szerveren találhatók, a kapcsolódó kliensek pedig azokon különböző kéréseket tudnak lefuttatni.

Ennek a módszernek lehet egy példája a Központi Statisztikai Hivatal (KSH) „Tájékoztatási adatbázisa", amelyben a látogatók a hivatal tulajdonában levő és maradó adatbázisokból tudnak anonimizált adatsorokat megtekinteni, letölteni – kizárólag a szerver számítási kapacitására hagyatkozva, és nem terhelve a felhasználói, azaz a kliensgép kapacitásait.

1.2. A számítástechnikai szoftver eszközök fejlődésének legfontosabb lépcsőfokai

A demográfia, és azzal párhuzamosan a statisztika több száz éves történetében nem kizárólag a számítási kapacitás egyre növekvő rendelkezésre állása miatt jelentős a számítástechnika fejlődése. A valószínűség-számítás kezdetei vagy a legkisebb négyzetek módszerének kidolgozása óta nemcsak robusztusabb, illetve többváltozós módszereket dolgoztak ki, hanem mára az analitikus kifejtés és levezetés helyett – természetesen ugyancsak a rendelkezésre álló olcsó számítási kapacitásra támaszkodva – általában a célszerűbb szimulációs módszereket alkalmazzák.

A teljesebb megértés érdekében röviden áttekintjük a statisztikai szoftverek fejlődését is.

Az első ökonometriai programok igen korán megjelentek (*Renfro* [2004]). Az EDSAC (Cambridge) már 1953-ban is futtatott ilyen szoftvert, de az 1950-es évek végén inkább csak az egyszerű alapműveletek elvégzésére használták ezeket, általános elterjedésük későbbre tehető.

Az ebben az időben készített programok speciális feladatra íródtak, amelyek között az átjárás sokszor lehetetlen volt. Egy sarkított példán keresztül bemutatva: egy

gépi kódban íródott ANOVA-eljárás teljesen más bemeneti adatstruktúrát követelhetett, mint egy ugyanazon gépre, azonos nyelven írott program, amely kereszttáblákat készített.

A statisztikai programcsomagok első generációjának megjelenéséig az 1960-as évek közepéig kellett várni, amelyek a korábbi problémákat áthidalva valóban komplex megoldást jelentettek, hiszen egységesen meghatározott bemenő adatokra a statisztikai módszerek széles skáláját tudták alkalmazni (*Leeuw* [2011]).

A BMD, majd a BMDP (BioMeDical Package) több mint 30 éves pályafutása 1965-ben az UCLA orvostudományi részlegén kezdődött. Az alapvetően egészségügyi számításokra felkészített statisztikai program kezdetben szabadon elérhető volt, később licenc-díjas termékké vált – egészen az 1996-os, SPPS Inc. általi felvásárlásáig, azóta a program nem áll aktív fejlesztés alatt.

A társadalomtudományi körökben is jól ismert SPSS (statistical package for the social sciences – társadalomtudományi statisztikai programcsomag) szoftvertermék 1968-ban jelent meg a Chicago-i Egyetem gondozásában. A program sikerét jelzi, hogy *Wellmann* [1998] az egyik legnagyobb hatású könyvként jelölte meg az SPSS 1970-es eredeti felhasználói kézikönyvét (*Nie et al.* [1970]).

A kezdetekben kizárólag társadalomtudományi területre koncentráló SPSS csak később, leglátványosabban a 2009-es IBM-felvásárláshoz kötött névváltoztatás – PASW (predictive analytics software –előrejelző analitikai szoftver) – során, illetve az SPSS átalakulásával (statistical product and service solutions – statisztikai termék- és szolgáltatásmegoldások) nyitott az egyéb tudományterületek világába.

Az eredetileg kizárólag parancssorból működő és lyukkártyák feldolgozására készített program az elmúlt 45 év során fokozatosan alakította ki saját fájlstruktúráját (sav), a grafikus felhasználói felületét (1985), a Java alapokra átépített, így platformfüggetlen programkódot (2007), és mára a „Base" csomagon kívül rengeteg további modul (add-on) is segíti a felhasználók munkáját a kérdőív szerkesztésétől és a minták meghatározásától az adatok kiértékeléséig.

Az inkább üzleti körökben ismert, de az SPSS-hez hasonlóan szintén igen elterjedt SAS (statistical analysis software – statisztikai elemző szoftver) megjelenése a North Carolina-i Állami Egyetemhez (North Carolina State University – NCSU) köthető (1968), és mára a „business intelligence" (BI – üzleti intelligencia) ágazat egyik legnagyobb kiszolgálójává vált a MicroStrategy, az IBM Cognos, az Oracle Hyperion, a Microsoft BI és az SPSS Modeler mellett.

A szoftver alapjainak kidolgozása az NCSU egy korábbi diákja nevéhez fűződik, aki előbb az ANOVA, majd a többváltozós lineáris regresszió implementálása (1966) után egy keretrendszer elkészítéséhez kezdett. A program elterjedését nagyban segítette, hogy az 1968-tól már többszerzős koprodukció képes volt hatékonyan kezelni az adathiányt.

A SAS fejlesztésében jelentős mérföldkövet jelentett előbb a platformfüggetlenség felé tett lépés az 1980-as évek elején különböző mini (azaz nem mainframe (nagy-)) számítógépek támogatásával, majd a FORTRAN és az IBM által fejlesztett PL/I, illetve gépi kódról a C nyelvre történő átállás. Mára szinte bármely számítógépen elérhető, sőt szerveroldali hosztolt, ún. „ondemand" (igény szerinti) szolgáltatást is nyújtanak.

Leeuw [2011] a statisztikai programcsomagok második generációjának megszületését 1985-höz köti, mikor is mind a három említett szoftver grafikus felülettel egészült ki, illetve a három nagy program mellett újabbak is megjelentek a piacon, elsődlegesen a felhasználói felület barátságosabbá tételén dolgozva.

A Data Desk 1986-ban debütált Macintosh számítógépekre, amelynek elsődleges célja az „exploratory data analysis" (feltáró adatelemzés) elősegítése volt, számos vizuális adatelemző eszköz felhasználásával. A program nagy előnyét a felhasználóbarát és interaktív kezelőfelülete jelentette, amely segítségével a kevésbé hozzáértő kezekben is látványos eredmények születhettek. 1997 óta elérhető Windows alól is, azonban napjainkban a program már nem áll állandó fejlesztés alatt.

Nem sokkal később, 1989-ben jelent meg a JMP (jump) program a SAS egyik társalapítójának felügyelete alatt, szintén Macintosh platformon. Ennek megfelelően a statisztikai programcsomagok korábban megszokott grafikus felületének további csiszolását tűzték ki elsődleges feladatukként a fejlesztők, amely eredményeképpen immáron interaktív grafikonok is segítették a feltáró jellegű adatelemzést.

A STATA (1985) máig tartó sikerét elsősorban a felhasználói aktivitásnak, és az azt lehetővé tevő programok vagy programrészek megoszthatóságának köszönheti (user contributed code – felhasználó által adott kód). Az interneten „ado" fájl formátumban közzétett STATA-kódok lehetővé teszik a felhasználóknak, hogy a mások (tehát nem a STATA fejlesztői) által kifejlesztett statisztikai eljárásokat adott licenc szerint felhasználják, illetve hivatkozzák.

A program ma is aktív fejlesztés alatt áll, 2003 óta grafikus felülettel is rendelkezik. Felhasználói bázisa meglehetősen nagy az ismertetett kiterjeszthetőségnek köszönhetően, levelező listája a korábban említett programokhoz és szinte bármely kereskedelmi szoftverhez képest kimagasló forgalommal (havi több, mint ezer levél) bír.

A STATA sikerét is felülmúló R program kialakulásáig vezető út bemutatása előtt előbb érdemes áttekinteni az azt megalapozó ún. S nyelvet és annak rövid történetét.

A Bell Laboratories belső hálózatában már az 1970-es évek második felétől használták a *John Chambers* által fejlesztett S programcsomagot. Nagy előnye volt a korábbi, egyedi feladatokra írt FORTRAN programokkal szemben, hogy egységes parancsok segítették az interaktív adatelemzést, illetve a különböző statisztikai módszerek elvégzéséért felelős függvények (programrészek) könnyen elérhetők voltak a fejlesztők számára.

A nagyszámítógépekre szánt General Comprehensive Operating Systemről UNIX-ra történő portolás (1980), majd a program (1981), illetve a programkód (1984) megnyitása a külvilág felé garantálhatta leszármazottjai hatalmas sikerét.

Az 1980-as évek végére az immáron több, mint tíz éves program többszöri átdolgozása után megjelent a „New S" nyelv, amely a korábbi makrók helyett már valódi függvényekre épített, újabb grafikus eszközök (X11 és PostScript) váltak használhatóvá, kialakul a napjainkban is használt „formula-notation" és az alapértelmezett S3, majd később az S4 metódusok.

Bár az S a mai napig elérhető, időközben újabb implementációi terjedtek el világszerte olyannyira, hogy mára a TIOBE-index (a programozási nyelvek népszerűségét számszerűsítő lista) szerint például az R már a leggyakrabban használt programozási nyelvek sorában bekerült az első 30 közé (2012 decemberében éppen 25. a listán), és az S kereskedelmi változata (S-PLUS) is többször szerepelt az első 100 között.

A matematikus és statisztikus körökben mára szinte megkerülhetetlen R program fejlesztése – *Gerald Jay Sussman* SCHEME nyelvére és az S eredményeire, funkcióira épített (*Hornik* [2012]) azok újraírásával – 1993-ban indult az Auckland-i Egyetemen *Ross Ihaka* és *Robert Gentleman* vezetésével. A program sikerét talán jól jelzi, hogy Chambers, az S egykori ötletgazdája és fejlesztője is csatlakozott/felvételt nyert az R központi fejlesztőcsapatába (R Development Core Team).

A szoftver nyílt forráskódú: szabadon használható, terjeszthető és módosítható a GPL v2[1] licenc mellett. A Free Software Foundation által elismert program, a GNU része. Számos platformon ingyenesen elérhető a telepítésre kész változata (Windows, Macintosh, Linux), sőt, napjainkra sok grafikus felhasználói felület („frontend", „graphical user interface") segíti az R-t használók mindennapi munkáját a hagyományos parancssorok, megoldások és azok integrált környezetben (Eclipse/StatET, Emacs/ESS, Rstudio, TextMate, Notepad++ stb.) való futtathatósága mellett.

Az R sikerét az ingyenes és szabadon használható volta mellett (vagy talán inkább az alapján) elsődlegesen a CRAN (comprehensive R archive network) csomagtárolónak és a felhasználók által megosztható programkódoknak köszönheti. Mára a CRAN több mint 4500 R csomagot számlál, amelyek többnyire lefedik a kurrens statisztikai módszerek tárházát.

Bár a CRAN-re bárki beküldhet ún. „library-ket" (kiegészítő csomagokat), és azokon kizárólag automatikus teszteket futtatnak a hálózat üzemeltetői, a nagy számú felhasználó és az aktív közösség (GitHub, StackOverflow, [R-help] és egyéb levelezőlisták több mint havi 3000 üzenete stb.) állandó ellenőrzése és visszajelzése egyfajta garanciát jelent a programok karbantartására és további fejlesztésére. E mellett az R Core Development Team kezeskedik az alapcsomagok és néhány további

[1] http://gnu.hu/gpl.html

library hibamentes működéséért, illetve mára a valóban standard munkaeszközzé vált R többek között klinikai vizsgálatok esetében is megfelelő felülvizsgálattal és tanúsítványokkal bír (*The R foundation* [2012]).

Az itt bemutatott programokon kívül még számos egyéb üzleti programcsomag (MATLAB, Mathematica, Statistica stb.) elérhető az érdeklődők számára, azonban a tanulmány szempontjából kevéssé meghatározónak mondható jellegük miatt azok ismertetésétől itt eltekintünk.

2. Szövegközi R parancsok

Napjainkban a statisztikai programokról szóló online társalgások központi témáját adják a „megismételhető" („reproducible research"), ún. „annotált" jelentések készítése („literate programming") az R segítségével. Ennek az eljárásnak a lényege, hogy az elemzés folyószövegébe „csempészett" R kódot (ún. „chunk"-ok tartalmát) feldolgozva a kész anyag a szöveg között az eredményeket tartalmazza, ráadásul a szerző által meghatározott formátumban. A szerzők véleménye szerint ezzel egy új fejezet nyílt az elemzések világában: nem szükséges többé táblázatkezelő eszközökben finomítani a statisztikai programok outputját, hogy azt majd egy szövegszerkesztőbe átmásolva tudjuk végleges formába önteni, hiszen mindezt megtehetjük egy lépésben is – az adatokra és nem a segédeszközökre koncentrálva.

Természetesen a „reproducible" vagy „literate research"-nek megvannak a maga hagyományai, például az ún. „Noweb" fájlformátum már 1994 óta használatos (*Johnson* [1997]). Ehhez hasonlóan az R kód folyószövegbeli integrációja is régóta megoldott a Sweave[2] segítségével – az említett a Noweb szintaxisára építve –, azonban használata olykor körülményes, és kizárólag a pdf formátumot támogatja (*Leisch* [2002]).

Így az elmúlt 10 évben számos változat látott napvilágot, amelyek nagy számára való tekintettel saját „CRAN-feladatnézet" is készült „Reproducible Research" címmel (*Zeileis* [2005]). Itt találhatók többek között LaTeX, pdf, HTML, ODT, markup/markdown fájlformátumú kimenettel dolgozó csomagok is. Azonban még 2011-ben is, egymástól két független R csapat gondolta úgy, hogy a meglevő megoldások nem nyújtanak kielégítő eredményt.

A „knitr" csomag[3] (XIE 2012) célja a Sweave felváltása, amely testre szabható funkcionalitásával méltón nyeri el egyre több R felhasználó szimpátiáját. Immáron nem csak pdf, de markdown és HTML kimenet is előállítható ugyanazon kódsor

[2] Pdf formátumú riportok generálására használt R csomag.
[3] Általános célú, dinamikus jelentésgenerálásra képes R csomag.

alapján, ráadásul a „chunk”-ok (R kifejezéseket tartalmazó utasításdarabok) kezelése sokat egyszerűsödött a Sweave paramétereihez képest.

Daróczi Gergely és *Aleksandar Blagotić* [2012] hasonló csomag megírására tett kísérletet 2011-ben, amely eredménye többek között a „pander” és „rapport” csomagok. A „knitr”-rel ellentétben a programok célja nem az egyedi folyószövegben található R parancsok feldolgozása (bár a „pander” erre is lehetőséget nyújt), hanem különböző annotált statisztikai modulok elkészítése volt. Így tehát elkészíthető például egy ANOVA-modul, amely bármely adatbázis kiválasztott változóira futtatva formázott táblázatokkal, grafikonokkal és magyarázatokkal kiegészített riportot képes generálni teljesen automatikusan.

A következőkben ezeket a programcsomagokat és az azokra épülő webalkalmazásunkat fogjuk bemutatni. Az R forráskódot a folyószövegtől elkülönítve, az eredeti megjelenés szerint közöljük.

2.1. Pander: az R-től a Pandoc-ig

A „pander” csomag (*Daróczi* [2012]), amely eredetileg a „rapport” csomag részét képezte a 2012-es, a modularitás érdekében szükséges kiválásáig, arra nyújt lehetőséget, hogy szinte bármely R objektum – így akár egy táblázat vagy egy lineáris regresszió vagy mondjuk egy főkomponens-elemzés eredménye – leképezhető legyen a „pander” S3 „method” segítségével Pandoc (*MacFarlane* [2012]) nyelvjárásban. A részletekkel kapcsolatban lásd a program dokumentációját.

A „Pandoc’s markdown” egy továbbfejlesztett „markdown” nyelv, amelynek konvertáló programja képes a megfelelő szintaxis szerint felépített szövegfájlok több formátumba történő átalakítására – legyen az többek között pdf, HTML, MS Word docx, OpenDocument (odt) vagy valamilyen egyéb markdown formátum.

Így a Pandoc és a „pander” csomagnak köszönhetően megnyílt a lehetősége annak, hogy bármely R eredmény „csatolható” legyen az általánosan használt szöveges dokumentumformátumokban a felhasználó különösebb beavatkozása nélkül, amely korábban csak igen körülményesen, és a kimeneti formátumok szerint korlátozott programcsomagok segítségével volt lehetséges (például xtable, Hmisc, ascii).

A továbbiakban ezen funkcionalitás bemutatására teszünk kísérletet egy mintaadatbázis, néhány észak-amerikai gépjármű adatainak (*Henderson–Velleman* [1981]) felhasználásával.

Itt szeretnénk felhívni az Olvasó figyelmét arra, hogy az „elemzés” során nem az eredményekre és az azok alapján történt következtetésekre helyezzük a hangsúlyt (hiszen az említett, historikus adatbázissal már egyébként is számos kutatás során foglalkoztak), hanem azt szeretnénk bemutatni, hogy a „nyers” R objektumok hogyan hasznosíthatók a riportok folyószövegében.

Az „*mtcars*” adatbázis az alap R programcsomag része, és a következő változókat tartalmazza 32 amerikai gépjármű esetében:

1. **mpg –** Miles/(US) gallon.
(*A gépjármű fogyasztása az egy mérföldre jutó, gallonban kifejezett üzemeanyag-felhasználás mértékét megadva. Tehát minél magasabb az érték, annál kevesebb üzemanyagot fogyaszt a jármű.*)
2. **cyl –** Number of cylinders.
(*A gépjárművekben található cilinderek száma: 4, 6 vagy 8.*)
3. **disp –** Displacement (cu.in.).
(*A motor hengerűrtartalma négyzethüvelykben kifejtve.*)
4. **hp –** Gross horsepower.
(*A gépjármú teljesítménye (lóerő).*)
5. **drat –** Rear axle ratio.
(*A nyomaték mértéke.*)
6. **wt** – Weight (lb/1000).
(*A gépjármű súlya 1000 fontokban kifejezve.*)
7. **qsec –** 1/4 mile time.
(*A gépjármű gyorsulása: mennyi idő alatt tesz meg negyed mérföldet. Tehát minél alacsonyabb az érték, annál jobban gyorsul a jármű.)*
8. **vs –** V/S.
(*A kormányzás, meghajtás típusa.*)
9. **am –** Transmission (0 = automatic, 1 = manual).
(*Kézi (manuális) vagy automata váltó van-e a gépjárműben?*)
10. **gear –** Number of forward gears.
(*A sebességek száma: 3, 4 vagy 5.*)
11. **carb** Number of carburetors.
(*A karburátorok száma: 1, 2, 3, 4, 6 vagy 8.*)

Az adatokat egyszerűen megjeleníthetünk markdown, MS Word, pdf vagy HTML dokumentumban is a „pander” csomag segítségével:

```
> pander(head(mtcars))
```

Ezt a parancsot bármely (telepített „pander” csomaggal ellátott) R konzolba beírva a következő eredményt kapjuk vissza:

Ez egy egyszerű, pusztán karakterekből felépített, Pandoc markdown ún. „multiline” táblázat, amelyet azonban a Pandoc képes MS Word, pdf vagy egyéb formátumokba is egyszerűen exportálni a „pander” csomag beépített parancsának felhasználásával:

```
> Pandoc.brew(text='<%=head(mtcars)%>', output=tempfile(), convert='docx')
```

A parancs egyetlen ún. „chunk”-ot tartalmazott, amely R kódot a program MS Word kompatibilis formátumban jelenített meg számunkra.

```
------------------------------------------------------------------------------------------
                   mpg   cyl   disp   hp   drat   wt     qsec   vs   am   gear   carb
----------------------- ----- ----- ------ ---- ------ ----- ------ ---- ---- ------ ------
     **Mazda RX4**      21     6    160    110  3.9    2.62  16.46  0    1    4      4

   **Mazda RX4 Wag**    21     6    160    110  3.9    2.875 17.02  0    1    4      4

    **Datsun 710**      22.8   4    108     93  3.85   2.32  18.61  1    1    4      1

  **Hornet 4 Drive**    21.4   6    258    110  3.08   3.215 19.44  1    0    3      1

**Hornet Sportabout**   18.7   8    360    175  3.15   3.44  17.02  0    0    3      2

     **Valiant**        18.1   6    225    105  2.76   3.46  20.22  1    0    3      1
------------------------------------------------------------------------------------------
```

1. táblázat

Az mtcars adatbázis első 6 sora

Autómárka	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21	6	160	110	3,9	2,62	16,46	0	1	4	4
Mazda RX4 Wag	21	6	160	110	3,9	2,87	17,02	0	1	4	4
Datsun 710	22,8	4	108	93	3,85	2,32	18,61	1	1	4	1
Hornet 4 Drive	21,4	6	258	110	3,08	3,21	19,44	1	0	3	1
Hornet Sportabout	18,7	8	360	175	3,15	3,44	17,02	0	0	3	2
Valiant	18,1	6	225	105	2,76	3,46	20,22	1	0	3	1

A továbbiakban az egyszerűség kedvéért kizárólag a nyers eredményekhez szükséges R parancsokat ismertetjük, és a markdown, illetve a MS Word kompatibilis formátumot nem mutatjuk be, amely ugyanakkor automatikusan lefut a *rapporter.net* rendszerben is (lásd később).

Amint látható, az adatbázis szinte kizárólag magas mérési szintű változókkal bír, így érdemes megnézni az azok között páronként lehetséges lineáris összefüggések erősségét:

```
> round(cor(mtcars), 1)
```

A parancs elkészíti az adatbázis 11 változója között meghatározható Pearson-féle korrelációs együtthatókat tartalmazó mátrixot, amelyben diagonálisában értelemszerűen csupa egyes értékek szerepelnek.

2. táblázat

Az mtcars adatbázis változóinak korrelációs mátrixa

Változó	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
mpg	1	–0,9	–0,8	–0,8	0,7	–0,9	0,4	0,7	0,6	0,5	–0,6
cyl	–0,9	1	0,9	0,8	–0,7	0,8	–0,6	–0,8	–0,5	–0,5	0,5
disp	–0,8	0,9	1	0,8	–0,7	0,9	–0,4	–0,7	–0,6	–0,6	0,4
hp	–0,8	0,8	0,8	1	–0,4	0,7	–0,7	–0,7	–0,2	–0,1	0,7
drat	0,7	–0,7	–0,7	–0,4	1	–0,7	0,1	0,4	0,7	0,7	–0,1
wt	–0,9	0,8	0,9	0,7	–0,7	1	–0,2	–0,6	–0,7	–0,6	0,4
qsec	0,4	–0,6	–0,4	–0,7	0,1	–0,2	1	0,7	–0,2	–0,2	–0,7
vs	0,7	–0,8	–0,7	–0,7	0,4	–0,6	0,7	1	0,2	0,2	–0,6
am	0,6	–0,5	–0,6	–0,2	0,7	–0,7	–0,2	0,2	1	0,8	0,1
gear	0,5	–0,5	–0,6	–0,1	0,7	–0,6	–0,2	0,2	0,8	1	0,3
carb	–0,6	0,5	0,4	0,7	–0,1	0,4	–0,7	–0,6	0,1	0,3	1

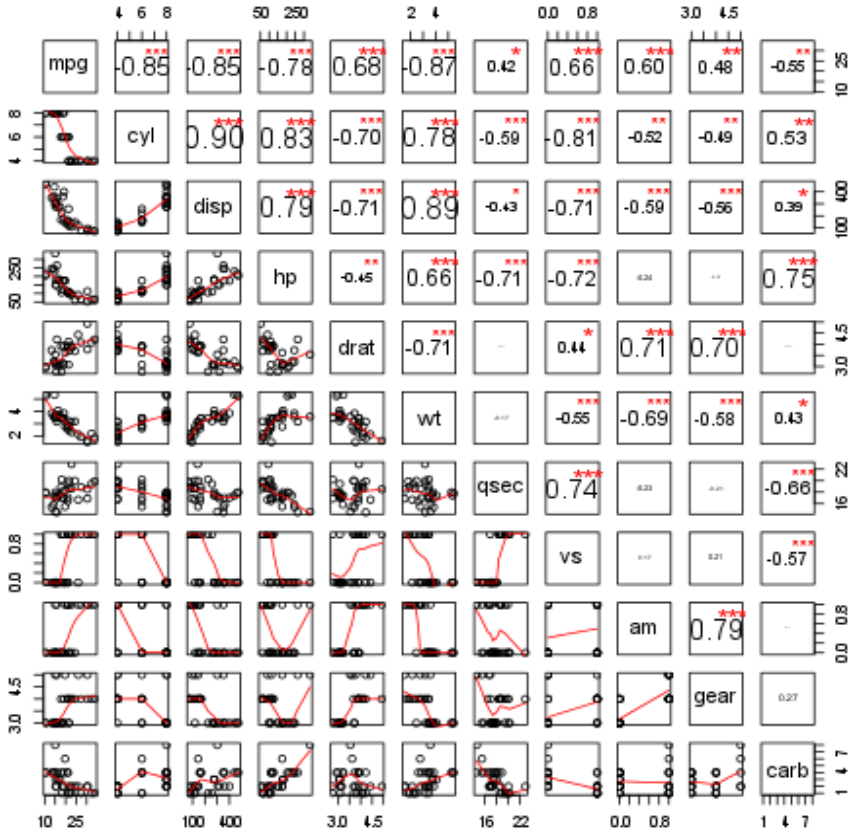
Természetesen e parancs módosításával (lásd „method” paramétert a „cor” függvénynél) ugyanilyen könnyen meghatározható a Spearman- vagy a Kendall-kovarianca vagy korrelációs együttható értéke is, illetve a jelen adatbázisban nem jelentkező, de olykor előforduló adathiányok kezelése is testre szabható (lásd a „na.rm” és a „use” paramétereket).

Az egyszerűségre törekedve a szignifikanciatesztek eredményeit itt nem tüntettük fel, azonban némileg összetettebb parancs segítségével még részletesebb elemzés készíthető:

```
> panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...) {
    par(usr = c(0, 1, 0, 1))
    r   <- cor(x, y, use = 'complete.obs')
    txt <- format(c(r, 0.123456789),
        digits = digits,
        decimal.mark = panderOptions('decimal.mark'))[1]
    txt <- paste(prefix, txt, sep = "")
    if(missing(cex.cor))
        cex <- 0.8/strwidth(txt)
    test <- cor.test(x,y)
    Signif <- symnum(test$p.value, corr = FALSE, na = FALSE,
        cutpoints = c(0, 0.001, 0.01, 0.05, 0.1, 1),
        symbols = c("***", "**", "*", ".", " "))
    text(0.5, 0.5, txt, cex = cex * abs(r) * 1.5)
    text(.8, .8, Signif, cex = cex, col = 2)
}
> pairs(mtcars, lower.panel = 'panel.smooth', upper.panel = 'panel.cor')
```

Ebben a parancsban meghatároztunk egy új „panelt” a felső háromszögben, ahol a kiszámolt korrelációs együtthatók alapján ábrázoljuk az értékeket (nagyság szerint kiemelve), illetve feltüntetjük a hozzájuk tartozó szignifikanciaszinteket:

1. ábra. Az mtcars adatbázis változóinak pontdiagramja és korrelációs mátrixa



E kódsort egy „chunk”-ban szerepeltetve nem a képernyőn jelenik meg az 1. ábra, hanem a meghatározott dokumentumtípusba ágyazva általában „png” formátumban. Tehát a „pander” csomag a folyószövegben előforduló R parancsoknál automatikusan érzékeli, ha grafikont, képet készít a felhasználó, azt fájlba rendezi (azaz létrehozza a kép fájlt), majd beépíti a kért dokumentumba.

De visszatérve a példánkhoz: jól látható, hogy a járművek teljesítményét („hp”) erősen és érthető módon meghatározza a cilinderek és karburátorok száma, a hengerűrtartalom, a jármű súlya és a gyorsulás, illetve a fogyasztásváltozókkal szintén erős,

ám fordított irányú összefüggést figyelhetünk meg az angolszász mértékegységeknek köszönhetően.

Ezek alapján érdekes lehet egy regressziós modellt építeni az említett változóknak megfelelően. Az eredeti R objektum alapértelmezetten csak egy rövid összefoglalót mutat a kért modellről:

```
> lm(hp ~ cyl + carb + disp + wt + mpg + qsec, data = mtcars)

Call:
lm(formula = hp ~ cyl + carb + disp + wt + mpg + qsec, data = mtcars)

Coefficients:
(Intercept)          cyl         carb         disp           wt          mpg         qsec
  112.75636      1.80445     21.69340      0.46082    -33.55501     -1.87325      0.05729
```

A nyers R objektumból a „pander” kinyeri az együtthatókon kívül a standard hiba mértékét és a *t*-próba eredményét, illetve automatikusan felcímkézi a táblázatot:[4]

```
> pander(lm(hp ~ cyl + carb + disp + wt + mpg + qsec, data = mtcars))
```

3. táblázat

Lineáris regressziós modell az mtcars adatbázis lóerő változójára

	Estimate	Std. error	t value	*Pr*(>|*t*|)
(Intercept)	112,8	125,4	0,8989	0,3773
cyl	1,804	7,367	0,2449	0,8085
carb	21,69	5,164	4,201	0,0002953
disp	0,4608	0,1415	3,258	0,003226
wt	–33,56	17,88	–1,876	0,07232
mpg	–1,873	1,895	–0,9886	0,3323
qsec	0,05729	5,785	0,009903	0,9922

Fitting linear model: hp ~ cyl + carb + disp + wt + mpg + qsec

Ehhez hasonlóan például egy főkomponens-elemzésnél sincs nehezebb dolgunk. A következőkben nézzük meg, hogy az „mtcars” adatbázis első négy változója alapján milyen PCA-modellt kapunk vissza a „pander” S3 eljárásán keresztül:

```
> pander(prcomp(mtcars[, 1:4]))
```

[4] Az R program által automatikusan generált táblázatokat és ábrát változtatás nélkül közöljük.

4. táblázat

Principal Components Analysis

	PC1	PC2	PC3	PC4
mpg	–0,03812	0,009204	0,997	–0,06658
cyl	0,01204	–0,003373	–0,06614	–0,9977
disp	0,8996	0,4355	0,03087	0,007334
hp	0,4348	–0,9001	0,02538	0,006606

5. táblázat

Principal Components Analysis

	PC1	PC2	PC3	PC4
Standard deviation	136,5	38,12	3,028	0,6594
Proportion of Variance	0,9272	0,07228	0,00046	2e–05
Cumulative Proportion	0,9272	0,9995	1	1

Végül soroljuk csoportokba az adatbázis elemeit minden előfeltevést nélkülözve, automatikusan meghatározva a létrehozandó klaszterek számát a „cluster” (*Maechler et al.* [2012]) és az „fpc” (*Henning* [2012]) csomagok segítségével:

```
> library(cluster)
> library(fpc)
> cn  <- pamk(mtcars)
> fit <- kmeans(mtcars, cn$nc)
```

Ahol az automatikusan meghatározott optimális klaszterszám a „cn” alapján 2, a klaszterek középpontja pedig:

```
> res <- fit$centers
> row.names(res) <- paste0(1:nrow(res), '.')
> res
```

6. táblázat

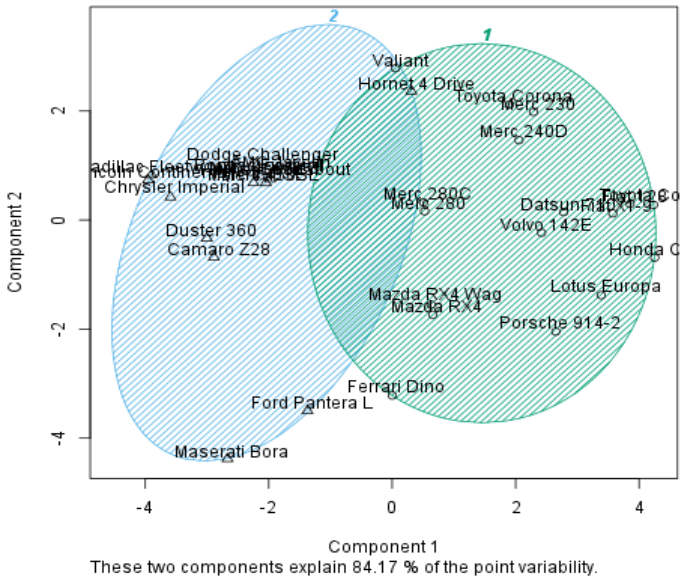
A két klaszterközéppont

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
1.	15,1	8	353,1	209,2	3,229	3,999	16,77	0	0,1429	3,286	3,5
2.	23,97	4,778	135,5	98,06	3,882	2,609	18,69	0,7778	0,6111	4	2,278

A két klaszter és az azokat tartalmazó esetek egyszerűen ábrázolhatók egy közös grafikonon:

```
clusplot(cn$pamobject,
    fit$cluster,
    color = TRUE,
    shade = TRUE,
    labels = 2,
    lines = 1,
    main = '',
    col.p = 'black',
    col.clus = panderOptions('graph.colors'))
```

2. ábra. A két klaszter ábrázolása



2.2. Rapport: annotált statisztikai modulok

A korábbiakban bemutatott R kódok alkalmasak az „*mtcars*” adatbázis felületes elemzésére, és apróbb módosításokkal könnyen alkalmazhatók egyéb adattömbökre is. Az R-ben dolgozó statisztikusok számára mindez a napi rutin része, és sokszor nagyon hasonló kódokat, parancsokat futtatnak hasonló struktúrájú adatbázisokon.

A „rapport” csomag ötlete pontosan ebből fakadt: a rutinszerű, standard eljárások futtatása, majd értelmezése és szavakba öntése valójában többnyire automatizálható feladat (*Daróczi–Blagotić* [2012]).

A cél tehát egy olyan R csomag összeállítása volt, amely segítségével ún. modulok (template) készíthetők, amelyek azután bármely szabványos adatbázison alkalmazhatók, a változónevektől függetlenül. Így például elkészíthető egy kiterjesztett ANOVA-modul, amelyben az általános ügyféligények alapján szövegesen ismertetjük a módszer lényegét és célját, bemutatjuk a felhasznált változókat (leíró statisztikák), majd különböző statisztikai tesztek futtatása után grafikonokkal, ábrákkal színesítjük a táblázatos statisztikai tesztek eredményeit.

Egy ilyen template megírása egy tapasztalt R programozó számára 1–4 óra, amely ráfordítás sokszorosan megtérülhet a későbbi munkák során.

Elérhetővé tettük néhány standard statisztikai eljárás angol nyelvű modulját a „rapport” R csomagban: https://github.com/Rapporter/rapport/tree/master/inst/templates

Természetesen a modulok tetszés szerinti nyelvre is átültethetők, illetve a bemeneti paraméterek (például a kiválasztott adatbázis mely változóit vegye figyelembe a modul, vagy milyen számszerű, illetve szöveges egyéb argumentumok alapján fusson a program) is szabadon meghatározhatók az ún. „template”-eken belül. A részletekkel kapcsolatban lásd a csomag dokumentációját.

2.3. Rapporter.net: statisztika a felhőben

A röviden bemutatott csomagok fejlesztésével elsődleges cél tehát – azon túl, hogy az R-t napi szinten jelentések készítésére és tanulmányok írására használó statisztikusok és elemzők munkáját segítsük – a hagyományos statisztikai szoftverek világának remélt megreformálása volt.

Ugyanis az első ökonometriai program megjelenése óta ezen szoftverek a felhasználó gépén futnak, az adatok lokális meglétét követelik meg, ráadásul erősen függnek a rendelkezésre álló erőforrásoktól, amely a napjainkban hódító mobil eszközök esetében egyre korlátozottabban érhetők el.

Másfelől a jelenlegi – mind az ingyenes, mind a kereskedelmi – statisztikai programokat és csomagokat többnyire csak a szakavatott hozzáértők használják, pedig az idő haladtával és az eddigi tapasztalatok szerint egyre nő a rendelkezésre álló (elemzésre váró) adatok köre. Ennek megfelelően gyakran az egyszerű táblázatkezelők nem feltétlenül körültekintően megtervezett eredményeit próbálják értelmezni a sokszor kevéssé felkészült felhasználók.

A *rapporter.net* rendszerrel ezeken az általunk fontosnak vélt problémákon szeretnénk segíteni úgy, hogy egyfelől az elemzésekhez opcionálisan biztosítjuk a számítási kapacitást szerverparkunkban, másfelől az eddig megszokott statisztikai „outputok” helyett testre szabható, szöveges jelentéseket nyújtunk.

E mellett ahol lehetséges, ott a napi rutin részévé vált döntéseket a felhasználó helyett a program is meg tudja hozni, így a különböző „statisztikai varázslók" szintén elkészíthetők. Ilyen lehet például, ha a felhasználó kiválaszt két változót egy feltöltött adatbázisból, amelyek között lehetséges összefüggést keres úgy, hogy a változók eloszlásai alapján a rapporter.net rendszere képes megtalálni a megfelelő algoritmust és statisztikai tesztet a vizsgálathoz (például *t*-próba, korrelációs együtthatók vagy kereszttáblás elemzés).

Így talán valóra válhat *Chambers* [1980] jóslata, miszerint „az olcsóbb személyi számítógépek, és azok elosztott hálózata" minőségbeli változásokat eredményezhet a statisztikai számítások területén.

Az itt bemutatott csomagok, az alap R program, illetve a számtalan importálható CRAN-könyvtár mellett a rapporter.net kialakítása során számos egyéb technológiát használunk a teljesítmény maximalizálása és a biztonság garantálása érdekében.

A frontend és kezelőfelület alapvetően Ruby on Rails nyelvben íródott, amelyet a kliens oldalon JavaScript kódok segítenek. Háttéradatbázisként ún. NoSQL tárolókat használunk (CouchDB és MongoDB), és a Pandoc végzi a dokumentumok több formátumba történő exportálását.

A biztonságról alapvetően az AppArmor kernelmodul, illetve az R kódok futtatása során dinamikusan alkalmazott profilok gondoskodnak az „RAppArmor" (OOMS 2012) és a „pander" csomag vonatkozó fejlesztői ága segítségével. Ezeknek köszönhetően a felhasználók korlátozás nélkül, bármilyen R kódot futtathatnak a szervereken, de azok közvetlenül nem férnek hozzá a szerverek adatbázisához és merevlemezéhez, továbbá tiltott bárminemű nem R program hívása is.

Természetesen ez fejlesztői oldalról sok problémát eredményezett, így a felhasználók által futtatott programok számára egyedi R környezetet („environment") alakítottunk ki, ahova az adatbázisaik és a modulok futtatásához szükséges CRAN-könyvtárak már előzetesen betöltésre kerülnek. E mellett az említett, szigorú AppArmor profilokat tovább finomítottuk, például úgy, hogy az OpenBUGS meghívható legyen összetettebb bayes-i számításokra.

Egy másik R csomagunk, a „sandboxR" (*Daróczi* [2012]) gondoskodik arról, hogy a futtatott kódok az adott R munkamenetben se tudjanak kárt tenni. A csomag lényege, hogy a futtatás előtt feldolgoz („parse") minden R parancsot, és tiltott függvények (például alapvető beállítások vagy az R környezet módosítása) esetén megakadályozza a kódsor futását.

A rapporter.net rövid bemutatásakor érdemes kitérnünk arra is, hogy a rendszer szabadon skálázható, és nagy rendelkezésre állás biztosítható a többszerveres kialakításnak köszönhetően. Ez annyit jelent, hogy a szerverparkba korlátlan számú R számításokat végző gép beemelhető, illetve bármely R, adatbázis vagy frontend gép kiesése automatikusan pótolható a rendszer leállítása nélkül. E mellett a „pander" csomag automatikus „cache" (gyorsítótár) eljárása is garantálja, hogy a korábban már le-

futott és időigényes számításokat a rendszer észrevétlenül újrahasznosítsa, ezzel is elősegítve az R reszponzív, tehát minimális várakozással járó, gördülékeny és hatékony használatát.

A következőkben egy, a rendszerben magyar nyelven kialakított modul segítségével mutatjuk be a rapporter.net lehetséges felhasználási módjait.

2.3.1. Kistérségi adatokat elemző modul bemutatása

Annak érdekében, hogy bemutathassuk az előzőkben felvázolt statisztikai elemző és riportkészítő rendszer hasznosságát, készítettünk egy bemutatót a Rapporter rendszeren belül, amely kistérségi adatok leíró elemzését, térképezését és statisztikai jellegű elemzését képes elvégezni.

A sablont úgy alkottuk meg, hogy bármilyen kistérségi szintű, folytonos változókat tartalmazó adatbázis elemzését el tudja készíteni. Bemutatóként, jelen tanulmány számára a 2008-as KSH által publikált településsoros adatokból továbbszámított kistérségi adatsort választottuk ki és töltöttük fel rendszerünkbe.[5] Ezen adatbázis több száz különböző változót tartalmaz, amelyekből mi néhány kiemelten fontos változó (munkanélküliségi és demográfiai mutatók) elemzését végeztük el a Függelékben elérhető módon.

Ugyanakkor szeretnénk felhívni a figyelmet arra, hogy az általunk bemutatni kívánt rendszernek pont az a lényege, hogy bárki számára könnyen elérhetővé tegye a komplex elemzések készítését, tehát nemcsak az itt bemutatott néhány változó, hanem az adatbázis bármelyik változójának elemzését el lehet végezni általa.

Annak érdekében, hogy ezt bizonyítsuk, létrehoztunk egy bármely, internetre kapcsolt böngészőből elérhető hivatkozást, amelyre kattintva regisztráció nélkül ezt bárki megteheti (lásd a Függeléket).

Ezen a hivatkozáson ugyanakkor csak az általunk feltöltött 2008-as adatok kérdezhetők le, miközben fontos hangsúlyozni, hogy nem csak a példaadatok elemzésére alkalmas a rendszer, hanem a regisztrált felhasználók akár saját kistérségi szintű adataikat is feltölthetik és azokra is futtathatják saját elemzésüket.

Jelen tanulmányban hat darab 2008. évi munkanélküliségre vonatkozó, az aktív korú populáció számával arányosított változó, valamint három 2008-as demográfiai változó elemzését csatoltuk a rendszerből generálva:

1. nyilvántartott álláskeresők száma,
2. 180 napon túli nyilvántartott álláskeresők száma,
3. általános iskola 8 osztályánál kevesebb végzettséggel rendelkező nyilvántartott álláskeresők száma,
4. szakmunkás végzettségű nyilvántartott álláskeresők száma,

[5] *Forrás*: http://statinfo.ksh.hu/Statinfo/themeSelector.jsp?&lang=hu

5. egyetemi végzettségű nyilvántartott álláskeresők száma,
6. nyilvántartott pályakezdő álláskeresők száma,
7. állandó népességből a 0–2 évesek száma,
8. állandó népességből a 18–59 évesek száma,
9. állandó népességből a 60–x évesek száma.

A sablon megalkotásánál arra törekedtünk, hogy közvetlenül is felhasználható legyen az általa készíthető elemzés eredménye, tehát akár kész jelentésként is ki lehessen nyomtatni. A sablon által készített jelentés több részre tagolódik, amely részek minden elemezni kívánt változónál külön-külön megtalálhatók.

Az első részben az adatok leíró statisztikáját láthatjuk – hagyományos statisztikai eszközök alkalmazásával (középértékek, eloszlás kiemelt értékei) –, amelyet kiegészítettünk egy tematikus térképpel.

Ezen eljárásokat elméletileg bármelyik GIS szoftver képes elvégezni. Ugyanakkor azonban a térképek és az eloszlás bemutatása önmagában sokszor hiányérzetet kelt, hiszen nem olvashatók le róla, hogy mely kistérségek emelkednek ki leginkább.

Ezért a leíró táblázatot és térképet táblázatokkal és szöveges elemzéssel is kiegészítettük, ahol is a legalacsonyabb és a legmagasabb értékű kistérségek felsorolását és értékeiknek összehasonlítását végezzük el.

Itt külön szeretnénk felhívni a figyelmet arra, hogy ez az az első pont, amely talán leginkább képes rámutatni az általunk bemutatni kívánt rendszer újszerűségére, hiszen szöveges elemzések ilyen mértékben könnyed automatikus készítésére az általunk ismert statisztikai vagy akár GIS szoftverekben egyáltalán nem volt lehetőség.

Az elemzés második részében az adatok belső struktúrájának egyenlőtlenségi szempontú elemzését végezzük el. Az alapmutatót a közismert Gini-index jelenti, amely mellé négy egyéb indexet is kiszámítunk (*Cowell* [2000]).

Az egyenlőtlenség klasszikus grafikus elemzési lehetősége a Lorenz-görbe alkalmazása, amely egy koordináta-rendszerben felrajzolt görbe vonal segítségével képes bemutatni az értékek eloszlásának egyenlőtlenségét (*Arnold* [1987]).

Mivel az egyenlőtlenségi értékek értelmezése nem mindenki számára kézenfekvő, ezért a sablont úgy hoztuk létre, hogy minden elemzésben – kizárólag csak egyszer, az első változó esetén – legyen egy „Emlékeztető” megjegyzés, ahol röviden kitérünk a mutatók magyarázatára.

A harmadik részben egy talán kevéssé ismert eljárásnak, a térbeli autokorrelációnak a kiszámítását végezzük el a Moran-féle I-mutató felhasználásának segítségével (*Cliff–Ord* [1981]). A térbeli autokorreláció lényegében annak fokmérője, hogy a vizsgált jelenség területi eloszlásában felfedezhető-e valamilyen szabályszerűség (*Dusek* [2004]). Először globálisan – tehát egész Magyarországra – számítjuk ki a mutatót, és mutatjuk meg egy grafikon segítségével a hozzá tartozó szomszédsági értékek eloszlását.

A Moran-féle I-mutató a hagyományos korrelációs értékekhez hasonló módon értelmezendő, azaz minél inkább közelít az egyhez, annál erősebb a térbeli autokorreláció értéke, tehát annál erősebb a térbeli elrendezettség. A mutató szignifikanciaszintjét Monte-Carlo-szimuláció segítségével ellenőrizzük.

A globális autokorrelációs mutató kiszámításán túl további lehetőség az ún. lokális Moran-féle I-értékek kiszámítása, amely minden egyes kistérség esetén a szomszédsági értékek felhasználása által képes megmutatni, hogy mely kistérségek azok, amelyek nemcsak önmagukban, hanem környezetükkel együtt mozogva speciális jellegűnek minősíthetők. A lokális értékek kiszámítása révén tehát statisztikai értelemben is képesek vagyunk ún. hotspotok (magas értékű térségek) és coldspotok (alacsony értékű térségek) azonosítására. Ezen speciális régiók azonosítását mind térkép készítésével, mind pedig szöveges felsorolással végezzük el.

Az elemzés utolsó, összehasonlító fázisában – amely természetesen csak többváltozós esetben készül el – az összes elemzésbe vont változó értékeit táblázatosan és szövegesen is áttekinthetővé tettük, ez által képet alkothatunk arról, mely változók és kistérségek azok, amelyek a vizsgálati szempontok szerint leginkább kiemelkedőnek/speciálisnak számítanak.

Fontos megjegyezni, hogy a sablon készítésénél gondoltunk arra is, hogy az elemzők számára sok esetben az adatok térképi megjelenítése is elégséges, ezért meghagytuk annak is a lehetőségét, hogy az alkalmazás használatával kizárólag térképek generálását végezzék el.[6] Ilyen módon tehát az általunk készített sablonnal egy közel teljes értékű kistérségi térképezőt is készítettünk.[7]

Függelék

A következő internetes oldal a „Kistérségi adatok térképezése és elemzése” *rapporter.net* modul forráskódját tartalmazza:

https://github.com/Rapporter/templates/blob/master/KSH_NUTS4.tpl

Amely kipróbálható az alábbi link segítségével:

http://goo.gl/P9BkI

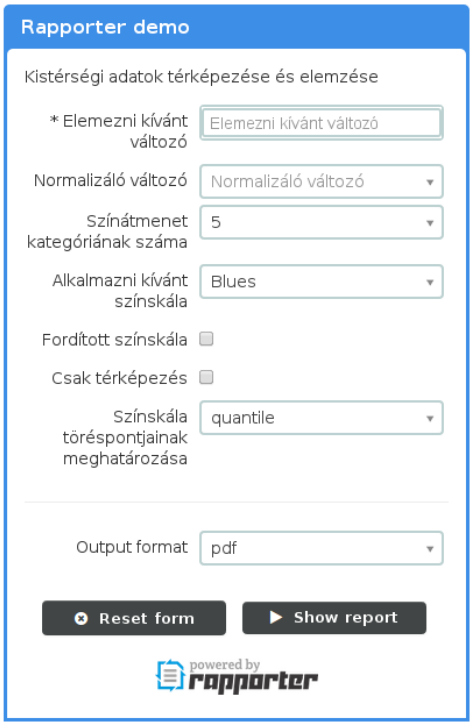
Az oldal a következő linkre mutat:

https://rapporter.net/api/form/
46bd5c2c7a4fefe8ca941bd356e80ef4dad182d26ffce3709edd0fdc87c8ee97

[6] Ehhez csak egy jelölőt kell tenni a „Csak térképezés” opciónál a paraméterek bevitele során.

[7] Ezen lehetőséggel egyben tudatosan fel is kívánjuk vetni az általában igen magas költségszinten megvalósított, jellemzően csak megjelenítésre használható külön GIS-szerverek telepítésének és üzemeltetésének indokoltságát.

A program kezelőfelületének képe:



Irodalom

ARNOLD, B. C. [1987]: *Majorization and the Lorenz Order: A Brief Introduction*. Springer. Berlin.

CHAMBERS, J. M. [1980]: Statistical Computing: History and Trends. *The American Statistician*. Vol. 34. No. 4. pp. 238–243.

CLIFF, A. D. – ORD, J. K. [1981]: *Spatial Processes*. Pion. London.

COWELL, F. A. [2000]: Measurement of Inequality. In: *Atkinson, A. B. – Bourguignon, F.* (eds.): *Handbook of Income Distribution*. Elsevier Science. Amsterdam.

DARÓCZI, G. [2012]: *pander: an R Pandoc Writer*. CRAN. http://rapporter.github.com/pander/

DARÓCZI, G. – BLAGOTIĆ, A. [2012]: *rapport: an R Templating System*. CRAN. http://rapport-package.info/

DARÓCZI, G. [2012]: *sandboxR: filtering malicious R calls*. https://github.com/Rapporter/sandboxR

Development Core Team R. [2010]: *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna. http://www.r-project.org

DUSEK T. [2004]: *A területi elemzések alapjai*. Regionális Tudományi Tanulmányok 10. ELTE TTK Regionális Földrajzi Tanszék. Budapest.

FRANCIS, I. [1981]: *Statistical Software: A Comparative Review*. Elsiever. New York.

LEISCH, F. [2002]: Sweave: Dynamic Generation of Statistical Reports Using Literate Data Analysis. In: *Härdle, W. – Rönz, B.* (eds.): *Proceedings in Computational Statistics.* Physica Verlag. Heidelberg. pp. 575–580.

HENDERSON, V. [1981]: Building Multiple Regression Models Interactively. *Biometrics*. Vol. 37. No. 2. pp. 391–411.

HENNIG, C. [2012]: *fpc: Flexible Procedures for Clustering.* CRAN. http://cran.r-project.org/package=fpc

HORNIK, K. [2012]: *The R FAQ*. http://cran.r-project.org/doc/FAQ/R-FAQ.html

JOHNSON, A. L. – JOHNSON, B. C. [1997]: Literate Programming Using Noweb. *Linux Journal*. Issue 42. pp. 64–69. http://www.linuxjournal.com/issue/42

JONG, V. J. DE. [1989]: *A Specification System for Statistical Software*. Centrum voor Wiskunde en Informatica. Amsterdam.

LEEUW, J. [2011]: Statistical Software: An Overview. In: *Lovric, M.* (ed.): *International Encyclopedia of Statistical Science.* Springer. Berlin. pp. 1470–1473.

MACFARLANE, J. [2012]: *Pandoc: A Universal Document Converter*. http://johnmacfarlane.net/pandoc/

MAECHLER, M. – ROUSSEEUW, P. – STRUYF, A. – HUBERT, M. – HORNIK, K. [2012]: *cluster: Cluster Analysis Basics and Extensions.* CRAN. http://cran.r-project.org/package=cluster

NIE, N. H. – BENT, D. H. – HULL, C. H. [1970]: *SPSS: Statistical Package for the Social Sciences*. McGraw-Hill. New York.

OOMS, J. [2012]: *The RAppArmor Package: Enforcing Security Policies in R Using Dynamic Sandboxing on Linux.* http://cran.r-project.org/web/packages/RAppArmor/RAppArmor.pdf

RENFRO, C. G. [2004]: *Computational Econometrics: Its Impact on the Development of Quantitative Economics*. IOS Press. Amsterdam.

RENFRO, C. G. [2009]: *The Practice of Econometric Theory: An Examination of the Characteristics of Econometric Computation*. Springer. Berlin.

ROUTH, D. A. [2007]: Statistical Software Review. *British Journal of Mathematical and Statistical Psychology*. Vol. 60. No. 2. pp. 429–432.

STERN, N. B. [1981]: *From Eniac to Univac: Appraisal of the Eckert-Mauchly Computers*. Digital Press. Bedford.

The R Foundation for Statistical Computing [2012]: *R: Regulatory Compliance and Validation Issues. A Guidance Document for the Use of R in Regulated Clinical Trial Environments.* p. 25. http://www.r-project.org/doc/R-FDA.pdf

VALERO-MORA, P. M. – LEDESMA, R. [2012]: Graphical User Interfaces for R. *Journal of Statistical Software*. Vol. 49. No. 1. pp. 1–8. http://www.jstatsoft.org/v49/i01

VON NEUMANN, J. [1993]: First Draft of a Report on the EDVAC. *IEEE Annals of the History of Computing.* Vol. 15. No. 4. pp. 27–75.

WELLMAN, B. [1998]: Doing It Ourselves: The SPSS Manual as Sociology's Most Influential Recent Book. In: *Clawson, D.* (ed.): *Required Reading: Sociology's Most Influential Books*. University of Massachusetts Press. Amherst. pp. 71–78.

XIE, Y. [2012]: *knitr: A General-Purpose Package for Dynamic Report Generation in R.* http://cran.r-project.org/package=knitr

ZEILEIS, A. [2005]: CRAN Task Views. *R News*. Vol. 5. No. 1. pp. 39–40.

Summary

The paper gives a brief summary on the history of computer-aided data analysis in the past century, and presents an alternative solution to the traditional statistical software methods by means of the cloud-based and R-driven data analysis and reporting platform of a Hungarian startup company. The features of this innovative application are presented by a use-case of analyzing spatial data.