EÖTVÖS LORÁND TUDOMÁNYEGYETEM
TERMÉSZETTUDOMÁNYI KAR

Wágner Beáta Eszter

ÉLETTARTAM MODELLEK

Szakdolgozat

Matematika BSc, elemző szakirány

Témavezető

Prőhle Tamás

Valószínűségelméleti és Statisztika Tanszék

Budapest, 2020.

NYILATKOZAT

Név: Wágner Beáta Eszter

ELTE Természettudományi Kar, szak: Matematika BSc.

NEPTUN azonosító: H8B684

Szakdolgozat címe:
Élettartam modellek

A szakdolgozat szerzőjeként fegyelmi felelősségem tudatában kijelentem, hogy a dolgozatom önálló szellemi alkotásom, abban a hivatkozások és idézések standard szabályait következetesen alkalmaztam, mások által írt részeket a megfelelő idézés nélkül nem használtam fel.

Budapest, 2020.12.28.

Wágner Beáta

a hallgató aláírása

Köszönetnyilvánítás

Köszönettel tartozom témavezetőmnek, Prőhle Tamásnak, akinek segítsége és türelme nagyban hozzájárult szakdolgozatom befejezéséhez.

Szeretnék köszönetet mondani családomnak, páromnak és barátaimnak, akik támogatása átsegített az egyetem nehezebb pillanatain. Külön kiemelném édesapámat, Wágner Péter Antalt, akinek a matematika szeretetét köszönhetem, és akinek segítsége nélkül most nem lennék itt.

Tartalomjegyzék

1. Bevezetés

Szakdolgozatom témájának mindenképpen valami olyat szerettem volna választani, ami a mai modern világban is releváns. Édesanyám biztosítási szférában dolgozik, sokat mesélt az életbiztosításokról, hogyan szabják meg a biztosítási díjakat, hogy dolgoznak a matematikusok a szakmában. Mindezek felkeltették az érdeklődésemet, majd mikor a témaválasztás aktuálissá vált, ezen szempontok alapján témavezetőm segítségével választottam témámként a népességstatisztikát.

Ez a demográfiának egyik olyan ágazata, mely a statikus és dinamikus népességi jelenségek megfigyelésével és elemzésével foglalkozik. Ebbe többek között a népesség összetételének változásai, születés, termékenység és halál tartozik.

Dolgozatom középpontjában a Lee-Carter modell áll, amit a mai napig használnak Magyarországon az átlagos várható élettartam becslésére. Az alapmodellnek számos továbbfejlesztése létezik. Ezek tipikusan abban különböznek az eredeti Lee-Carter féle modelltől, hogy másként közelítik kor-év specifikus demográfiai tábla adatait, és más modellt illesztenek az időfüggő komponensre.

Az eredeti Lee-Carter modell egy úgynevezett kétfaktoros modell, ami azt jelenti, hogy a demográfiai jelenségeket két tényezőt figyelembe véve modellezi. Mégpedig tipikusan az érintettek életkora és a jelenség (születés, elhalálozás, házasságkötés stb.) időpontja szerint. Az ilyen kétfaktoros modellezés esetén problémát jelent, hogy a modell eredményét egy harmadik faktorra, például a generációk modellezésére csak korlátozott mértékben használható. Egy-egy generáció sorsa a kor-év táblázatokban a diagonálissal párhuzamos cellák adataival közelíthető. Ennek közvetlen kezelésére használt klasszikus eszköz, az úgynevezett *Lexis-diagram*, mely az átlók mentén kettéosztott mezőkkel követi a generációk kor-év térben való mozgását.

A kapcsolódó analitikus módszer az úgynevezett APC (age-period-cohort) modell. E modelleknek alapvető problémája [9], hogy a faktorok becsülhetőségét, a faktorok közti nyilvánvaló függőség miatt csak a feltételezéseinek olyan kiegészítéseivel lehetséges, amelyek gyakorlati értelmezése nehézkes.

A dolgozatban a klasszikus módszer részleteivel foglalkozom. Bemutatom a működésének megértéséhez szükséges fogalmakat, modelleket, a szinguláris érték felbontástól egy lineáris idősor modellen, az ARIMA-n keresztül egészen a Lee-Carter modellig. Utóbbi ismertetése után végül az **R** programnyelv segítségével vizsgálatokat végzek a halálozási ráta felhasználásával.

2. Szinguláris érték felbontás

A matematikában többféle mátrixfelbontást ismerünk, ilyen például az LU-felbontás vagy a Cholesky-felbontás, melyek a Gauss-elimináción alapszanak.

Ezekkel ellentétben a következő fejezetben bemutatott szinguláris érték felbontás ortogonális mátrixok segítségével állítható elő.

2.1. A felbontás létezése és egyértelműsége

2.1.1. Definíció. Egy $A^{m\times n}$ mátrix olyan, $A = U\Sigma V^T$ alakban való felírását, ahol az $U^{m\times m}$ és a $V^{n\times n}$ ortonormált mátrixok, és a $\Sigma^{m\times n}$ egy diagonális nemnegatív mátrix, az A mátrix szinguláris felbontásának nevezzük. A Σ diagonális értékei az A mátrix szinguláris értékei.

2.1.2. Definíció. Egy W mátrixot ortonormált mátrixnak nevezünk, ha a transzponáltja egyben az inverze is, tehát ha

$$W^T W = WW^T = E$$

ahol az E egy megfelelő méretű identitás mátrixot jelöl.

2.1.3. Tétel. *Legyen A^{mxn} egy tetszőleges mátrix, ekkor létezik A-nak szinguláris felbontása. A Σ diagonális értékei egyértelmű halmazt alkotnak, és ha a Σ diagonális értékeire*

$$\sigma_1 > \sigma_2 > \cdots > \sigma_{min(m,n)} \geq 0\,,$$

akkor a felbontás is egyértelmű és az egyes szinguláris értékekhez tartozó U illetve V-beli vektorokat, az adott szinguláris értékhez tartozó szinguláris vektoroknak nevezzük. A nemnulla szinguláris értékek száma megegyezik a mátrix rangjával. Ha van a szinguláris értékek közt egyenlő, akkor az adott szinguláris értékhez nem egy vektorpár, hanem

egy altérpár tartozik. Ezeknek az altereknek a dimenziója megegyezik a hozzájuk tartozó szinguláris érték multiplicitásával, és ezek a szinguláris altérpárok egyértelműek.

Bizonyítás. A bizonyítás vázlata, abban a legegyszerűbb esetben, amikor a szinguláris értékek különbözőek, a következő. Keressük meg azt az 1 normájú v_1 vektort, amelyre az $||Av||$ maximális. Vagyis azt az egységvektort, amelyet az A transzformáció a legnagyobb mértékben nyújt meg. Legyen $\sigma_1 u_1 = Av_1$, ahol $||u_1|| = 1$. Ezután keressük meg azt a v_2 egységvektort, amelyet az A, a v_1-re merőleges egységvektorok közül a legnagyobb mértékben nyújt meg. Legyen $\sigma_2 u_2 = Av_2$, ahol $||u_2|| = 1$. Folytassuk az eljárást az eddig talált v_1, v_2-re merőleges egységvektorok körében. Ekkor kapjuk az (v_3, u_3) vektorpárt és a σ_3 nemnegatív szorzótényezőt. Folytassuk az eljárást mindaddig míg a nyert vektorpárok száma el nem éri a $\min(m,n)$ számot. Ezután, ha az m vagy az n nagyobb volna mint a $\min(m,n)$, akkor vegyünk hozzá a v vagy az u vektorokhoz még további annyi ortonormált vektort, hogy a megfelelő tér bázisát kapjuk. Az így nyert v vektorrendszer a konstrukció folytán ortonormált, az u pedig könnyen beláthatóan szintén ortonormált bázisa a neki megfelelő térnek. A szinguláriserték felbontás három eleme a szukcesszív maximalizálással nyert vektorok alapján a következő: V az a mátrix, amelynek oszlopai a v bázis vektorai. A U ugyanígy az u bázisból adódik. A Σ diagonális mátrix pedig a σ_k nyúlási (zsugorodási) értékekből képzett megfelelő méretű diagonális mátrix. Ugyanis az így definiált $B = U\Sigma V^T$ mátrixra tetszőleges $v_1, ..., v_k, ..., v_{\min(m,n)}$ esetén

$$Bv_k = U\Sigma V^T v_k = U\Sigma e_k = U\sigma_k e_k = \sigma_k u_k$$

ahol e_k azt az n elemű vektort jelenti, amelynek minden eleme 0 kivéve a k.-at. Az utóbbi egyenletek, ha $n > min(m,n)$ azaz, ha $m < n$, akkor kiegészítve azzal, hogy az olyan k-ra, amelyre $k > m$, a $Bv_k = 0$, mert $\Sigma e_k = 0$, azt is jelentik, hogy a $B = A$. Ugyanis a két mátrix (lineáris leképezés) bázison (a $v_1, ..., v_n$ bázisról van szó) egyenlő. Tehát az $U\Sigma V^T$ szorzat tényleg az A szinguláris felbontása.

Az u vektorok ortogonalitása a következő módon igazolható.

Tegyük fel, hogy $u_2 \neq u_1$, ekkor további megszorítás nélkül feltehetjük azt is, hogy $u_2^T u_1 > 0$. Megmutatjuk, hogy ekkor az u_1 nem lehet a maximálisan megnyúló vektor, mert egy elég kicsire választott $\varepsilon > 0$ esetén a

$$v_* = \frac{v_1 + \varepsilon v_2}{|v_1 + \varepsilon v_2|}$$

vektor A szerinti képének már az u_1 irányú komponense is hosszabb σ_1-nél.

Ugyanis

$$A(v_*) = \frac{1}{|v_1 + \varepsilon v_2|} A(v_1 + \varepsilon v_2) = \frac{\sigma_1 u_1 + \sigma_2 u_2}{\sqrt{1+\varepsilon^2}}$$

Tehát az $A(v_*)$ vektor u_1 irányú komponensének a hossza:

$$\begin{aligned} u_1^T A(v_*) &= u_1^T \left(\frac{\sigma_1 u_1 + \sigma_2 u_2}{\sqrt{1+\varepsilon^2}} \right) \\ &= (\sigma_1 + \varepsilon \sigma_2 u_1^T u_2)\big(1 - \frac{\varepsilon^2}{2} + \mathscr{O}(\varepsilon^4)\big) \\ &= \sigma_1 + \varepsilon \sigma_2 u_1^T u_2 - \mathscr{O}(\varepsilon^2) \\ &> \sigma_1 \end{aligned}$$

Amiből $|A(v_*)| > \sigma_1$, és ez ellentmond annak, hogy a konstrukció szerint σ_1 a legnagyobb megnyúlási mérték. □

2.2. A felbontás implementációja az R programnyelvben

A következő programrészlet azt mutatja meg, hogyan kapható meg az **R**-project 'base' csomagjának `svd()` parancsával egy véletlenszerűen generált 3×5 méretű mátrix szinguláris érték felbontása.

```
m <- 3
n <- 5
set.seed(123)
A <- matrix(rnorm(m*n),m,n)
```

```
A
#        [,1]  [,2]   [,3]   [,4]   [,5]
# [1,] -0.560 0.071  0.461 -0.446  0.401
# [2,] -0.230 0.129 -1.265  1.224  0.111
# [3,]  1.559 1.715 -0.687  0.360 -0.556

F <- svd(A,m,n)
str(F)
# List of 3
#  $ d: num [1:3] 2.663 1.703 0.601
#  $ u: num [1:3, 1:3] -0.262 0.322 0.91 0.163 ...
#  $ v: num [1:5, 1:5] 0.56 0.595 -0.433 0.315 ...

print(F)

# $d
# [1] 2.6631814 1.7027730 0.6006038
#
# $u
#         [,1]   [,2]  [,3]
# [1,] -0.262  0.163 0.951
# [2,]  0.322 -0.914 0.245
# [3,]  0.910  0.370 0.187
#
# $v
#         [,1]   [,2]   [,3]  [,4]   [,5]
# [1,]  0.560  0.409 -0.496 0.025  0.523
# [2,]  0.595  0.310  0.699 0.109 -0.222
# [3,] -0.433  0.574 -0.001 0.695 -0.020
# [4,]  0.315 -0.622 -0.094 0.711  0.027
# [5,] -0.216 -0.142  0.507 0.007  0.823
s <- F[[1]]
U <- F$u
V <- F$v
all(s>0) # TRUE
all.equal(diag(m),t(U)%*%U)          # TRUE
all.equal(diag(n),t(V)%*%V)          # TRUE
all.equal(A,U%*%diag(s,m,n)%*%t(V)) # TRUE
```

A programrészlet megjegyzésként szereplő részei a parancsok eredményeit mutatják. Az `svd()` parancs két méret-paraméterének explicit megadására azért van szükség, mert az $n > m$, azaz a leképezett tér a képtérnél magasabb dimenziójú. Az eljárás ugyanis csak a paraméterek explicit megadása mellett egészíti ki az egyértelműen adódó $min(m,n)$ elemű ortonormált rendszert a képtér és a leképezett tér közül magasabb dimenziójút a tér bázisává. Az utolsó sorban azért van szükség a `diag()` parancsban a diagonális mátrix méretének megadására, mert ez a mátrix nem feltétlen négyzetes.

A szinguláris felbontási tétel állításait az utolsó négy parancs igazolja. Nevezetesen, hogy egyrészt a kapott s szinguláris értékek tényleg nem negatívok, az U és V mátrixok ortonormáltak továbbá, hogy az $A = U\Sigma V^T$ egyenlőség érvényes.

2.2.1. Megjegyzés. A felbontás – természetesen – az ortogonális kiegészítés nélkül is érvényes:

```
F <- svd(A) # méret nincs megadva, mindkettő m=min(m,n)
s <- F[[1]]
U <- F$u
V <- F$v
all(s>0) # TRUE
all.equal(diag(m),t(U)%*%U)      # TRUE
all.equal(diag(m),t(V)%*%V)      # TRUE m=min(m,n)
all.equal(A,U%*%diag(s)%*%t(V)) # TRUE
```

2.3. Módszer alkalmazása közelítésre

A szinguláris érték felbontást fel tudjuk használni közelítésre is a következő módon.

Adott mxn-es A mátrix és $k \in \mathbb{Z}$. Legyen $A = U\Sigma V^T$ a mátrix szinguláris érték felbontása, ahol $\Sigma = diag(\sigma_1, ..., \sigma_m)$. Célunk, hogy találjunk egy megfelelő $\widehat{A}$ mátrix közelítést, úgy, hogy $rang(\widehat{A}) \leq k$. Ehhez minimalizálnunk kell a $A - \widehat{A}$ különbség Frobenius normáját:

$$\begin{aligned} \left\|A-\widehat{A}\right\|_F &= \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} X_{ij}^2} \\ min\left\|A-\widehat{A}\right\|_F &= \sqrt{\sigma_{k+1}^2+\ldots+\sigma_m^2} \end{aligned}$$

Legyen $\widehat{A}_k$ annak az U_k, V_k, Σ_k mátrixnak a $\widehat{A}_k = U_k\Sigma_k V_k^T$ szorzata, ahol U_k az U első k oszlopa, V_k az V első k oszlopa — feltételezve, hogy az U és a V oszlopai a megfelelő szinguláris értékek csökkenő sorrendjében vannak felsorolva — és Σ_k egy olyan $k \times k$ méretű diagonális mátrix, amelynek az átlójában a k legnagyobb szinguláris érték van.

2.3.1. Tétel. *Az Eckart-Young tétel azt mondja, hogy a vázolt módszer a lehető legkisebb Frobenius hibával rendelkező k-rangú mátrixot adja meg.*

A tétel bizonyítását elhagyjuk, ehelyett gyakorlatban mutatjuk meg a helyességét. Az alábbi programrészletben létrehozunk egy `Fr()` függvényt. Ezután az első két szinguláris vektorpárt, majd csak az elsőt felhasználva egy közelítését adjuk az eredeti mátrixnak.

```
Fr <- function(u) return(sqrt(sum(u^2)))

k <- 1:2 ; approx.A <-U[,k]%*%diag(s[k])%*%t(V[,k])
all.equal(Fr(s[-k]),Fr(A - approx.A)) # TRUE
k <- 1   ; approx.A <-U[,k]%*%diag(s[k],1,1)%*%t(V[,k])
all.equal(Fr(s[-k]),Fr(A - approx.A)) # TRUE
```

A két összehasonlítás azt mutatja, hogy ezzel a módszerrel olyan közelítését kapjuk az eredeti mátrixnak, amelyiknek a közelítés hibája a közelítéskor fel nem használt szinguláris értékekből képzett vektor Frobenius normájával egyenlő.

```
r.v <- rnorm(3)%*%A
r.u <- A%*%rnorm(5)
rand.approx.A <- r.u%*%r.v # egy véletlen diádszorzat
Fr(A - rand.approx.A)> s[1] # TRUE
```

Ebben az utasításban az A által leképezett tér és az A képterének egy-egy véletlenszerűen választott vektorát hívjuk meg. A mátrix közelítés pedig ezekből a vektorokból származó diádszorzat, egy véletlen, 1 rangú, 3×5 méretű mátrix. A `TRUE` eredmény mutatja, hogy ez rosszabb közelítése az A mátrixnak, mint a szinguláris felbontásából származtatott.

3. Lineáris idősor modellek

3.1. Az ARIMA modellek

Az idősorok modellezésének egyik alap modellje az ARMA modell, vagyis az autoregressziós mozgóátlag folyamat. Az ARMA folyamatot AR (autoregressziós folyamat) és MA (mozgóátlag folyamat) egyesítéseként hozták létre. Ha olyan a folyamat, hogy a valahányad rendű differencia folyamata egy ARMA folyamat, akkor az adott folyamatot ARIMA folyamatnak nevezzük.

3.1.1. Definíció. Az ε_t folyamatot zajfolyamatnak nevezzük, ha ε_t, $t = 1,...,T$ független, azonos eloszlású 0 várhatóértékű valószínűségi változók sorozata.

3.1.2. Definíció. Legyen ε_t egy zajfolyamat, 1 szórással, a $b_i \in \mathbb{R}$, $i = 1,...,p$ adott konstansok, és a $\sigma_\varepsilon > 0$ egy tetszőleges pozitív konstans. Az Y_t folyamat elégítse ki az

$$Y_t = b_1 Y_{t-1} + b_2 Y_{t-2} + ... + b_p Y_{t-p} + \sigma_\varepsilon \varepsilon_t$$

egyenletet. Ekkor az Y_t folyamatot p-edrendű autoregressziós folyamatnak nevezzük.
Jelölés: AR(p)

3.1.3. Definíció. Legyen ε_t egy zajfolyamat, $m_j \in \mathbb{R}$, $j = 0,...,q$ konstansok. Ekkor az

$$Y_t = m_0 \varepsilon_t + m_1 \varepsilon_{t-1} + .. + m_q \varepsilon_{t-q}$$

folyamatot q-rendű mozgóátlag folyamatnak nevezzük.
Jelölés: MA(q)

3.1.4. Definíció. Ha ε_t egy zajfolyamat és az Y_t az

$$b_0 Y_t + b_1 Y_{t-1} + ... + b_p Y_{t-p} = m_0 \varepsilon_t + ... + m_q \varepsilon_{t-q}$$

egyenlet megoldása, akkor az Y_t egy ARMA(p,q) folyamat.

Egy folyamat akkor ARIMA(p,d,q) folyamat, ha a folyamat d. differenciája egy ARMA(p,q) folyamat. Azaz, hogy megfelelő $b_1,...,b_p$ és $m_1,...,m_q$ konstansokra:

$$\Delta^d Y_t + b_1\Delta^d Y_{t-1} + ... + b_p\Delta^d Y_{t-p} = \varepsilon_t + m_1\varepsilon_{t-1} + ... + m_q\varepsilon_{t-q}$$

egy ε_t, $t = ...,-1,0,1,...$.
Átrendezve az egyenletet a következő ekvivalens formát kapjuk:

$$\Delta^d Y_t = -b_1\Delta^d Y_{t-1} - ... - b_p\Delta^d Y_{t-p} + \varepsilon_t + m_1\varepsilon_{t-1} + ... + m_q\varepsilon_{t-q}$$

Itt $\Delta Y_t = Y_t - Y_{t-1}$ minden t-re és $\Delta^d = \Delta^{d-1}\cdot\Delta$ továbbá $\Delta^0 = I$.

Ugyanez a modell az L időbenvaló visszalépés operációval tömörebb formában is felírható. Az alkalmazandó L operáció egy tetszőleges ξ_t, $t = ...,-1,0,1,...$ időben lezajló folyamat esetén olyan, hogy

$$L\xi_t = \xi_{t-1} \quad \text{és} \quad L^{k+1} = L^k\cdot L \quad \text{valamint} \quad L^k\xi_t = \xi_{t-k}$$

Ezt az operációt alkalmazva, tekintsük a

$$B(z) = 1 + b_1 z + ... + b_p z^p \qquad M(z) = 1 + m_1 z + ... + m_q z^q \qquad D(z) = 1 - z$$

polinomokat, amelyekkel az általános ARIMA(p,d,q) modell, a

$$B(L)D^d(L)\,Y_t = M(L)\,\varepsilon_t$$

formában írható fel.

3.2. Az exponenciális simítás modellje

Az exponenciális simítás modellje szerint

$$\widehat{Y}_{t+1} = \widehat{Y}_t + \lambda(Y_t - \widehat{Y}_t)$$

Azaz a folyamat következő értékének a becslése megegyezik a legutóbbi megfigyelésnek a becslés legutóbb tapasztalt hibájának λ-szorosával vett módosításával.
Vagy ekvivalens módon:

$$\widehat{Y}_{t+1} = \lambda Y_t + (1-\lambda)\widehat{Y}_t$$

Azaz az új becslés a legutóbbi időpontra tippelt és mért értékek súlyozott átlaga.
E modell mellett a folyamat feltételezett trendje konstans.

Ha azt feltételezzük, hogy a folyamat trendje egy lineáris függvény, akkor ennek megfelel a fenti modell következő módosítása:

$$\begin{aligned} \widehat{Y}_{t+1} &= \lambda Y_t + (1-\lambda)(\widehat{Y}_t - b_t) \\ b_t &= \beta(\widehat{Y}_t - \widehat{Y}_{t-1}) + (1-\beta)b_{t-1} \end{aligned}$$

Ekkor ugyanis a h lépéses előrejelzésként a

$$\widehat{Y}_{t+h} = Y_t + h \cdot b_t$$

időben lineáris becslés adódik.

Könnyen látható, hogy érvényes a következő állítás.

3.2.1. Állítás. *Az, hogy egy Y_t folyamat az exponenciális simítás modellje szerinti, az ugyanazt jelenti, mint hogy a folyamat egy az előző alpontban bemutatott ARIMA folyamatok speciális esete szerinti. Azaz egy ARIMA$(0,1,1)$ folyamat.*

Bizonyítás. Az ARIMA$(0,1,1)$ modell szerint az Y_t folyamat elsőrendű differenciája egy $MA(1)$ folyamat:

$$(1-L)Y_t = \mu + (1-mL)\varepsilon_t$$

vagyis

$$Y_t = Y_{t-1} + \mu + \varepsilon_t - m\varepsilon_{t-1},$$

ahol m egy valamilyen valós együttható, μ az esetleges drift és ε_t egy zajfolyamat.

Vegyük μ-t 0-nak. Legyen az optimális egy lépéses előrejelzés hibája $\hat{\varepsilon}_t = Y_t - \hat{Y}_t$. Ekkor az optimális előrejelzés a hibák optimális becslései alapján, figyelembe véve, hogy a t időpontban már van Y_t, de még nincs $\hat{\varepsilon}_{t+1}$:

$$\hat{Y}_{t+1} = Y_t - m\hat{\varepsilon}_t$$

Ha ebbe a képletbe beleírjuk a $\hat{\varepsilon}_t$ definícióját, és átrendezzük a képletet, akkor a következő formulát kapjuk:

$$\hat{Y}_{t+1} = m\hat{Y}_t + (1-m)Y_t,$$

ami az exponenciális simítás képletével azonos. □

4. A Lee-Carter modell

4.1. A modell leírása

Legyen $m_{x,t}$ a halálozási ráta a t évben az x korúak körében.
Ez a mátrix nyilván jelentős ingadozást mutat a szomszédos cellákat vizsgálva.
Kérdés egyrészt, hogy hogyan lehetne modellezni az ismert (x,t) kor-év adatokat. Másrészt, hogy hogyan lehetne becslést (előrejelzést) adni azokra az évekre, korokra, amelyekre nem rendelkezünk adatokkal.

A Lee-Carter modell [2] feltételezése szerint a mortalitási ráta logaritmusa a kor és év bilineáris modellje a következő módon:

$$ln(m(x,t)) = a(x) + b(x)k(t) + \varepsilon(x,t)$$

Ahol az $a(x)$ és a $b(x)$ a kor függvényei, a $k(t)$ pedig az évtől való függést írja le. Vagyis, van az elhalálozás valószínűségének a kortól függő két komponense $a(x)$ és $b(x)$, de a $b(x)$ komponens súlya függ a vizsgált évtől. Ez a $k(t)$ súly modellezi azt, hogy a halandóság tipikus esetben az évek haladtával csökkenő tendenciát mutat, függetlenül a vizsgált személy korától. Az $\varepsilon(x,t)$ az $a(x)+b(x)k(t)$ modell hibája. A feltételezés szerint 0 várható értékű, normális eloszlású.

Ez a modell további megkötések nélkül nyilván nem egyértelmű. Ugyanis tetszőleges λ konstans mellett egyformán jó modellt adnak az $(a, \lambda b, k/\lambda)$ modellek és úgyszintén egyformán jó modellt adnak tetszőleges μ konstans mellett az $(a-\mu b, b, k+\mu)$ modellek.

Vagyis az eddigi feltételek szerint a b csak egy multiplikatív konstans, a k pedig csak egy lineáris transzformáció erejéig meghatározott. Ennek a bizonytalanságnak a megakadályozására két korlátozást vezetünk be.

Legyen az X a figyelembe vett korcsoportok, a T figyelembe vett évek halmaza. Ekkor megkötjük, hogy

$$\begin{aligned} \sum_{x \in X} b(x) &= 1 \\ \sum_{t \in T} k(t) &= 0 \end{aligned}$$

legyen.

E feltételeknek persze közvetlen következménye, hogy az $a(x)$, a mortalitás időtől független komponense az $ln\big(m(x,t)\big)$ logaritmikus mortalitás időbeli átlaga.

Tehát a $b(x)k(t)$ szorzat a $ln\big(m(x,t)\big) - \overline{m}_t(x)$ adjusztált mortalitás tábla közelítése. A Lee-Carter modell szerint azt a (b,k) vektorpárt kell választani, ami az adjusztált mortalitás tábla első szinguláris értékéhez tartozó vektorpár. Ennek a választásnak az indokát az előző fejezetben tárgyalt approximációs tétel adhatja. Miszerint az első szinguláris vektorpár diádszorzata adja azt az 1 rangú mátrixot, amelyik a közelíteni szándékozott adjusztált mortalitás táblának a legjobb közelítése a Frobenius-norma szerint.

Felmerül, hogy miként modellezzük az így nyert mortalitás modell $k(t)$, időtől függő komponensét, annak értekében hogy egyrészt becslésünk legyen az $\varepsilon(x,t)$ nélküli mortalitás értékekre. Másrészt, hogy $t \notin T$ időpontokra előrejelzéseket tudjunk készíteni. Két egyszerű modell adódik: az egyik általánosabb, egy tetszőleges ARIMA(p,d,q) modell, a másik az exponenciális simítás modellje. Ugyanis az exponenciális simítás, mint láttuk, egy speciális ARIMA modell $(0,1,1)$ paraméterekkel, noha az exponenciális simítást tipikusan nem ARIMA formában szokás értelmezni.

5. Elemzés a Lee-Carter modell segítségével

5.1. A ‘demography’ programcsomag bemutatása

Az **R**-project ‘demography’ programcsomagja több függvényt is tartalmaz, amik segítségével demográfiai elemzéseket végezhetünk. Országok népességadataival – mint születési, vagy halálozási ráta – dolgozhatunk, előrejelzéseket végezhetünk el a jövőre nézve.

A dolgozatban a ‘demography’ csomag `lca()` függvényével és a hozzá kapcsolódó metódusokkal dolgoztunk. Ezek az eljárások a Lee-Carter modell szerint egy modellt illesztenek a megadott adatokra. Az előrejelzéseket ‘forecast’ generikus függvény ‘demography’ csomagbeli `forecast.lca()` metódusát felhasználva állítottuk elő. A következőkben részletesen ismertetjük ezeknek az eljárásoknak a működési módját, tekintettel arra, hogy ennek pontos dokumentációja, a tapasztalatunk szerint sehol sem található.

Az alkalmazott modell szerint az $r_{x,t}$ mortalitás ráta logaritmusa $ln(r_{x,t}) = a_x + b_x k_t + \varepsilon_{x,t}$ formában állítható elő, ahol x a korosztályt t pedig az időt jelöli. Az `lca()` függvény a rendelkezésre bocsátott adatok alapján megbecsüli az (a_x, b_x, k_t) komponenseket, és a becslési eredményeket egy `lca` osztályú objektumban adja vissza.

A modell szerint ahhoz, hogy tippünk legyen egy t_0 időpontban egy r_{x,t_0} ráta becslésére, az a_x és b_x ismeretében, mindössze a k_{t_0} becslésére van szükség. Egy ilyen becslést a `forecast()` függvényt alkalmazva kaphatunk.

Ha a `forecast()` függvényt egy `lca` osztályú objektummal paraméterezve hívjuk meg, és a ‘forecast’ kiegészítő csomag be van töltve, akkor a `forecast()` függvény érzékelve a kapott objektum osztályát, a `forecast::forecast.lca()` metódust hívja meg. Ez a metódus, — feltételezi, hogy a k_t egy driftes véletlen bolyongás,

ezért — a hívást tovább adja, a ‘`forecast`’ kiegészítő csomag `forecast::rwf()` függvényének.

Az `rwf()` függvény a hívást tovább adja a szintén a ‘`forecast`’ csomagból való `lagwalk()` függvénynek, amely egy `lagwalk` osztályú objektumban eltárolja a kapott k_t folyamatnak, mint driftes véletlen bolyongásnak a paramétereit és visszaadja a vezérlést az `rwf()` függvénynek. Ezután, szintén az `rwf()` függvény meghívja a `forecast()` függvényt (!), amely most, mivel egy `logwalk` osztályú objektumot kapott a `forecast.lagwalk()` metódussal készíttet előrejelzést. Végül ezt az előrejelzést kapja meg az `rwf()`-et hívó `forecast.lca()` metódus, amely az így kapott előrejelzéseket adja meg eredményként.

5.2. A magyarországi mortalitás adatok

A `www.mortality.org` tárhelyről rendelkezésünkre áll a `Population.txt` és az `Mx_1x1.txt` állomány. Ezen állományok az igényünknek megfelelően évenként, korosztályos bontásban tartalmazzák a magyarországi lakosság lélekszámát és korspecifikus halálozási rátáját. A vizsgálatban az évek száma 68, mivel a rendelkezésre álló lélekszám adatok 1950 és 2018 köztiek, míg a halálozási adatok a 1950 és 2017 közti évekre vonatkoznak, és ez utóbbi szűkebb az egymást követő évek számának tekintetében. A korosztályok száma 111, mivel a vizsgált adatok a `0-110+` életkoroknak felelnek meg. Az adatokból egy `hu.mort` nevű, `demogdata` osztályú adatobjektumot építünk, hogy az adatokat a '`demography`' programcsomag parancsaival kényelmesen kezelni tudjuk.

A `hu.mort` adatobjektum felépítése.

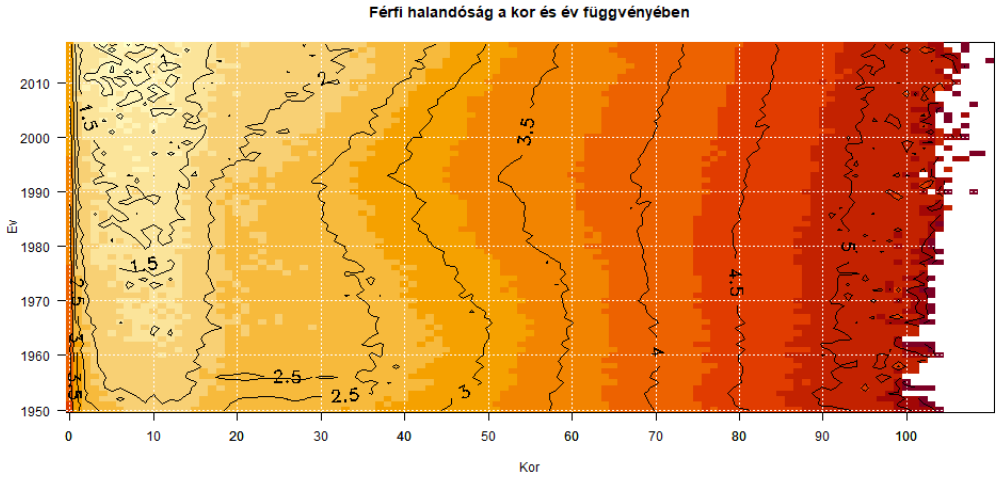
```
rm(list=ls())
library(demography)
mor.D <- read.table("Mx_1x1.txt",skip=2,header=TRUE,
                stringsAsFactors=FALSE, na.strings = ".")
dim(mor.D) # 7548 x 5 (7548=68*111)
str(mor.D)
# 'data.frame':   7548 obs. of  5 variables:
#  $ Year  : int  1950 1950 1950 1950 1950 1950 1950 ...
#  $ Age   : chr  "0" "1" "2" "3" ...
#  $ Female: num  0.08037 0.00716 0.00322 0.00206 ...
#  $ Male  : num  0.09971 0.00756 0.00348 0.00278 ...
#  $ Total : num  0.09030 0.00736 0.00335 0.00243 ...
pop.D <- read.table("Population.txt",skip=2,header=TRUE,
                stringsAsFactors=FALSE, na.strings = ".")
pop.D <- pop.D[pop.D$Year<2018,]
dim(pop.D) # 7548 x 5 (7548=68*111)


hu.mort <-list(type=NA, label=NA, lambda=NA,
               year=NA, age=NA, rate=NA, pop=NA)
```

```
hu.mort$type   <- "mortality"
hu.mort$label  <- "HUNGARY"
hu.mort$lambda <- 0 # Box-Cox transformation parameter
hu.mort$year   <- 1950:2017
hu.mort$age    <- 0:110

hu.mort$rate <- list(total = NA, female = NA, male = NA)
hu.mort$rate$female <- matrix(mor.D[,"Female"],111,68,
                              dimnames =list(0:110,1950:2017))
hu.mort$rate$male   <- matrix(mor.D[,"Male"],111,68,
                              dimnames =list(0:110,1950:2017))
hu.mort$rate$total  <- matrix(mor.D[,"Total"],111,68,
                              dimnames =list(0:110,1950:2017))

hu.mort$pop <- list(total = NA, female = NA, male = NA)
hu.mort$pop$female  <- matrix(pop.D[,"Female"],111,68,
                              dimnames =list(0:110,1950:2017))
hu.mort$pop$male    <- matrix(pop.D[,"Male"],111,68,
                              dimnames =list(0:110,1950:2017))
hu.mort$pop$total   <- matrix(pop.D[,"Total"],111,68,
                              dimnames =list(0:110,1950:2017))

class(hu.mort) <- "demogdata"
str(hu.mort)
hu.mort
# Mortality data for HUNGARY
#     Series: total female male
#     Years: 1950 - 2017
#     Ages:  0 - 110
```

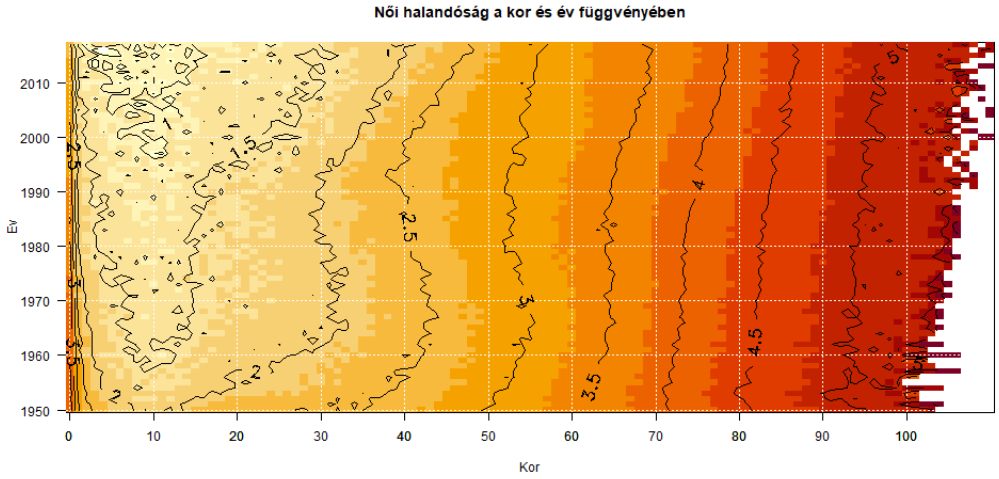
A fenti parancsok eredményeként tehát létrejött a `hu.mort` adatobjektum, melynek osztálya `demogdata`. Tehát a '`demography`' **R** kiegészítő csomag utasításaival (metódusaival) standard módon kezelhető.

Ez az objektum kétszer három 111×68 méretű táblázatot tartalmaz, 111 korosztályra az 1950 – 2017-ig tartó 68 évre. A táblák egyfelől a mortalitás ráta adatokat, másfelől a populációs lélekszámokat tartalmazzák, nők-férfiak és össz bontásban.

Az alábbi ábrák a férfi, illetve a női halandóság logaritmusát mutatják.



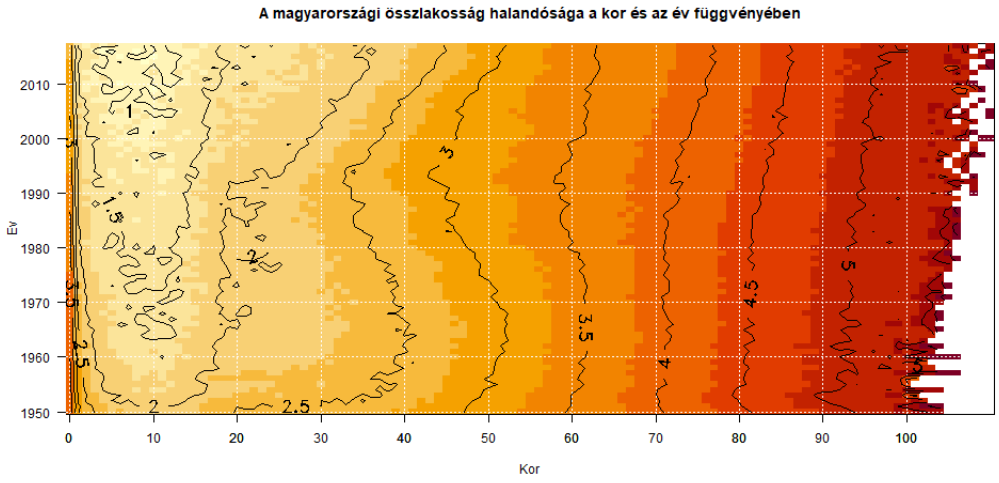
1. ábra. Férfi halandóság a kor és év függvényében



2. ábra. Női halandóság a kor és év függvényében

Az ábrákon jól megfigyelhető mindkét nem esetén az 5 éves kor alatti gyermek halandóság jelentős javulása. Ez a javulás kisebb mértékben a $10-24$-éves korokra is jellemző. Szembetűnő a $30-60$ éves férfiak halandóságának erőteljes romlása 1968

és 1994 közt. Gyenge visszaesés ugyanezekben az években a nők esetében is látható. De a nőknél törés csak a 40-es korosztály esetén mutatkozik, és ebben az esetben is csak sokkal kisebb mértékben, mint a férfiak esetén. A halandóság 2000-es években tapasztalható egyértelmű javulása mindkét nem esetén érvényes. Ugyanakkor a javulás az 50 év feletti nők esetén az egész vizsgált korszakra, azaz már az 50-es évektől kezdve jellemző.



3. ábra. A magyarországi halandóság a kor és év függvényében

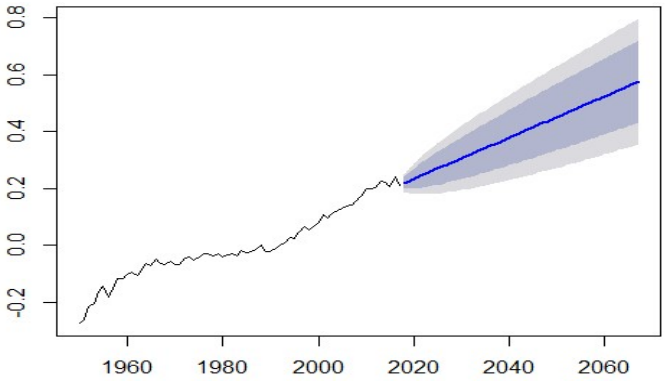
A hatvanas évek közepétől a kilencvenes évek elejéig tartó évek egyesített mortalitás adatain is jól megfigyelhető a mortalitási arányok jelentős romlása a 40-50 évesek körében.

5.3. Előrejelzés ARIMA és exponenciális simítás modell segítségével

A megismert modellek segítségével, a mortalitás tábla szinguláris érték felbontással nyert k_t időkomponens szerinti előrejelzéseket a következő fejezetben mutatjuk be. Minden előrejelzést 50 évre készítettünk.

Először az ARIMA$(0,1,0)$ modellre illesztjük rá, ehhez a '`forecast`' csomag `Arima()` parancsát alkalmazzuk. Vegyük az előző fejezetben létrehozott `hu.mort` adathalmazban szereplő együttes mortalitás táblát.

```
library(demography)
library(forecast)
mort.full <- hu.mort$rate$total
class(mort.full) #  "matrix"
dim(mort.full) #  111 68
rownames(mort.full) # "0"  "1"...  "109"  "110+"
colnames(mort.full) # "1950" "1951" ... "2016" "2017"
sum(is.na(mort.full)) # 242 hiányzo adat
sum(is.na(mort.full[1:101,])) # csak 0-100-ra teljes
mort <- mort.full[1:101,] # csak a teljes részt vesszük
ln.mort <- log(mort)
ln.mort.adj <- t(apply(ln.mort,1,function(u) u-mean(u)))
kt <- svd(ln.mort.adj)$v[,1] # első jobb-sziguláris vektor
kt <- ts(kt, start=as.numeric(colnames(ln.mort.adj)[1]))
M.arima010 <- Arima(kt,order=c(0,1,0),include.drift=TRUE)
plot(forecast(M.arima010,h=50))
```



4. ábra. ARIMA$(0,1,0)$ modellel nyert előrejelzés

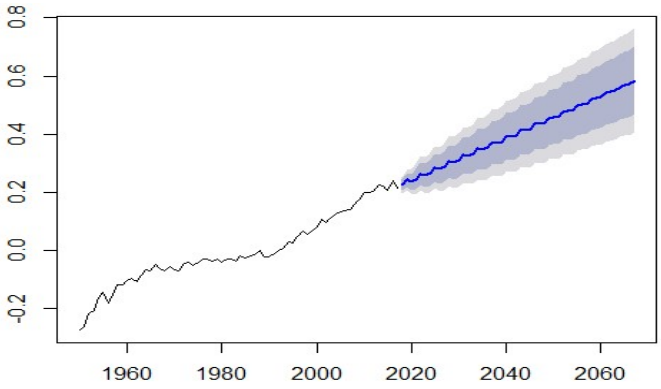
Ha a modell validitásának ellenőrzése képpen elvégezzük a Ljung-Box próbát, akkor kiderül, hogy ez a modell nem megfelelő. Esetünkben a zajfolyamat becsült autokorrelációinak súlyozott négyzetösszege – amit a Ljung-Box próba vizsgál – nagy, így a hozzátartozó p-érték nagyon kicsi. Vagyis a Ljung-Box próba null-hipotézise, mely szerint a modellhez tartozó hiba folyamat egy független folyamat, elvetendő. Ezért másik modellt érdemes választanunk.

Az `auto.arima()` függvény az Akaike-féle információs kritériuma szerint választ a megadott megfigyeléssorra optimális `ARIMA(p,d,q)` modellt. Itt elsődleges kérdés a $p+d+q$ össz-paraméterszám megválasztása. Az Akaike-kritérium a megfigyelések likelihood értékét az össz-paraméterszámmal módosítva egy olyan feltétel, amelyikben ellensúlyozva van, hogy egy modell a Frobenius kritérium szerint szükségszerűen annál jobb minél több paramétert tartalmaz.

Az `auto.arima()` eljárás az 'optimális' fokszámot — alapértelmezett esetben — a maximum $(5,2,5)$ fokszámú folyamatok közül választja ki, de nem az összes e feltételnek megfelelőt vizsgálva a klasszikus Box-Jenkins ajánlást követve. A differenciálás rendjét a Canova-Hansen teszt alapján határozza meg, a (p,q) fokszamokat pedig a [10] cikkben leírt, az AIC statisztikán alapuló lépésenkénti algoritmus szerint.

Esetünkben az `auto.arima()` eljárás a k_t folyamatra egy ARIMA$(2,1,2)$ modellt illeszt. E modell paramétereinek a becslése és az előrejelzés képe a következő:

```
M.arima212 <- auto.arima(kt,allowdrift=TRUE)
M.arima212
# Series: kt
# ARIMA(2,1,2) with drift
plot(forecast(M.arima212,h=50))
```

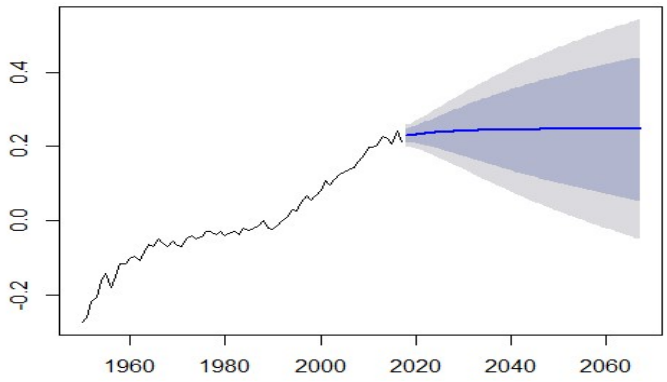


5. ábra. ARIMA$(2,1,2)$ modellel nyert előrejelzés

Ha erre a modellre elvégezzük a korábban is alkalmazott Ljung-Box próbát, akkor láthatjuk, ez a modell már sokkal jobb, mint az előző, hiszen az autokorrelációk súlyozott összege ebben az esetben, már nem tér el szignifikánsan a nullától.

Az exponenciális simítás modelljét, ami valójában egy ARIMA$(0,1,1)$ modell, a '`forecast`' csomag `ets()` utasításának segítségével nyerhetjük a következő módon.

```
M.ets1 <- ets(kt,model="AAN") # az idő komponens modellje
plot(forecast(M.ets1,h=50))
```



6. ábra. Exponenciális simítás modelljével nyert előrejelzés

Ha erre a modellre szintén elvégezzük a korábban is használt Ljung-Box próbát, akkor látszik, hogy ez a modell jobban illeszkedik, mint az ARIMA$(0,1,0)$, de rosszabbul, mint az Akaike-kritérium szerint optimálisnak választott ARIMA$(2,1,2)$ modell.

Nyilvánvaló a három modell közti különbség. Míg az előrejelzés átlagértéke az első két modell szerint az idővel arányosan lineárisan növekvő, addig a harmadik szerint lényegében konstans.

Ha pontosabban megnézzük, akkor látható, hogy az illesztett ARIMA$(0,1,0)$ és ARIMA$(2,1,2)$ modell két lényegében azonos driftes bolyongással írja le a megadott idősort. De a speciális ARIMA$(0,1,1)$ modellel azonos exponenciális simítás modell is egy driftes bolyongás! A különbség *csak* az, hogy míg az első két bolyongás driftje nem-nulla, addig a harmadiké nulla.

Ennek az az oka, hogy az exponenciális simítás a drift mértékét lokálisan becsüli meg, és a rendelkezésre álló adatsor utolsó elemeinek szakaszosan becsülhető várhatóértéke lényegében konstans. Ellentétben az első két modellel, amelyek a driftet globálisan becsülik meg. Azaz a várhatóérték becslést az egész megfigyeléssor alapján

végzik: a lényegesen korábbi időpontok alapján vehető becsléseket a legújabbakkal azonos súllyal veszik. Így ez a két modell érzékeli az idősor mért értékeiben tapasztalható globális lineáris növekedést.

Tehát elemzői szempontból a megfelelő modell kiválasztásakor az az alapkérdés, hogy a jövőbeli értékek becslésében megjelenő trend mértékét a lokális vagy a globális adatok alapján becsüljük-e meg.

5.4. Mortalitási ráta elemzése

A népességstatisztika egyik fontos eleme a mortalitási ráta, vagy magyarul halálozási arányszám. Ez mutatja meg egy adott népességen belül a halálozások arányát. Mérésére több mutatót létrehoztak.

Az ún. nyers halálozási arányszám az 1000 főre jutó halálozások száma egy évben. Ez a mutató viszont nem veszi figyelembe a népesség korösszetételét, így nem elég specifikus.

Ezért létrehozták az úgynevezett korspecifikus halálozási arányszámot, aminek segítségével már pontosabb képet láthatunk egy adott népesség halálozási arányáról.

Ebben a fejezetben a mortalitási ráta felhasználásával végzünk elemzéseket, a 5.1. fejezetben megismert ‘`demography`’ csomag `lca()` függvényének segítségével. Ebben az esetben viszont – ellentétben az előző fejezetben ismertetett előrejelzésekkel – a k_t időkomponenst nem közvetlenül a szinguláris érték felbontással kapjuk, hanem a ‘`demography`’ csomag `lagwalk()` függvényének használatával a következő módon.

A `lagwalk()` a `stats::lm()` függvénnyel konstans regresszió modellt illeszt a k_t megfigyelések differenciáira. Ha a

$$k_t = \sum_{s=0}^{t} \delta_s + \varepsilon_t,$$

ahol a δ_s független, azonos eloszlású, nem feltétlen 0 várható értékű sorozat és ε_t független, akkor a δ_t várható értéke maga a drift. Vagyis a k_t differencia folyamatának a várható érték becslése — ez a `stats::lm()` függvény regresszió becslésével adódik — a drift becslése.

Legyen a drift becslése $\hat{\delta}$, melynek a becsült szórása $\hat{\sigma}_\delta$. Az ε_t 0 várható értékű zaj becsült szórása pedig $\hat{\sigma}_\varepsilon$. Ezen becslések birtokában a h időtávra az x korosztály mor-

talitásának becslése

$$ln(r_{x,t+h}) = \underbrace{a_x + b_x k_t}_{\hat{r}_{x,t}} + h\hat{\delta},$$

ahol az $\hat{r}_{x,t}$ helyett $r_{x,t}$ is használható, amennyiben az az adott x és t értékekre ismert.

A becslés α szintű konfidencia tartományának sugara

$$z_{\alpha/2}\sqrt{h\sigma_\varepsilon^2 + h^2\sigma_\delta^2},$$

ahol a $z_{\alpha/2}$ a standard normális eloszlás $\alpha/2$ kvantilise.

Kétféle ábrát készítettünk el a 2017-es évre a ‘`demography`’ csomag `lca()` függvényével és a hozzá kapcsolódó metódusokkal, amik működéséről a 5.1. fejezetben írtunk részletesebben.

Az első ábrán a korspecifikus halálozási arány becslése látható 2017-re, ha az adatokat egy megadott évig vesszük figyelembe. Az elsőként közölt ábra azt az esetet mutatja, amikor az összes 2017-et megelőző adatot felhasználjuk. A második a 65 évesek mortalitási rátájának becslését mutatja a 2017-es évre, annak függvényében, hogy a rendelkezésre álló évek közül melyik az utolsó, a becsléskor még felhasznált év.

Az első ábrát a következő kódrészlet segítségével hoztuk létre.

```
rm(list=ls())
library(demography)
load("humor.R")
ls()

# a halálozási ráta a teljes (ff+nő)
# populációra együtt az 1-100+ korúakra
# az 1950-2017 évekre

W <- hu.mort$rate$total[1:101,]

# létrehozzuk a 100 fölötti korokra a mortalitási rátát
```

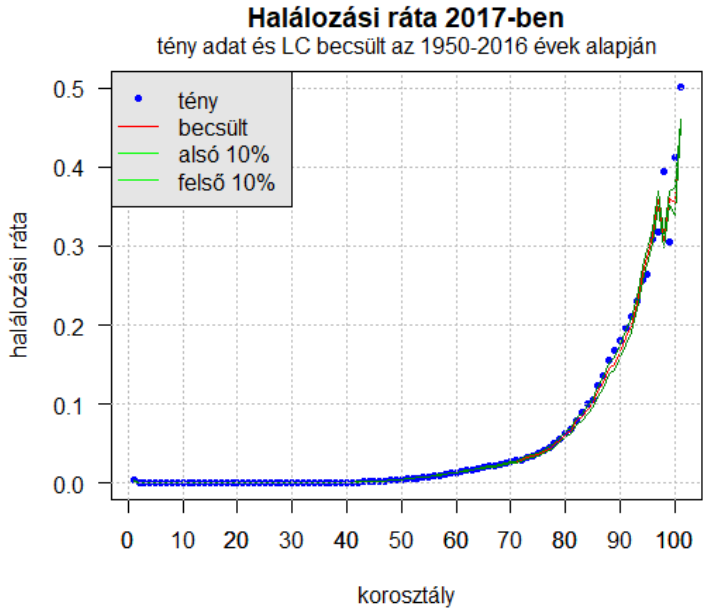
```
W[101,] <- colSums(hu.mort$rate$total[101:111,]*
           hu.mort$pop$total[101:111,],na.rm = TRUE)/
           colSums(hu.mort$pop$total[101:111,],
   na.rm=TRUE)

row.names(W)[101] <- "100+"
rate <- W
dim(rate)# 101 x 68
teny <- rate[,68]
# ez a 2017-es tény adat a 0-100+ korosztályokra

# ---
# 2017-es becslés az 1950-2016 adatok alapján

vago <- function(A,h) # "h" a levágott évek száma
  {
    vag <- function(o)
     return( if(is.matrix(o)) o[,1:(ncol(o)-h)]
               else           o[1:(length(o)-h)])
    D <- A
    D$year <- vag(A$year)
    D$rate <- lapply(A$rate,"vag")
    D$pop  <- lapply(A$pop,"vag")
    return(D)
  }

becsul <- function(A,h) # "h" az előrejelzési távolság
  {
    veg <- function(o) return(o[,ncol(o)])

    M <- lca(A, series ="total",max.age = 100,adjust="dt")
    F <- forecast(M,h=h,jumpchoice="actual")
    est <- F$rate # ez az átlag becsült értéke
              # és 80%-os konfidenciája
    est <- lapply(est,"veg")
    est <- as.matrix(data.frame(est))
    colnames(est) <- c("mean","lower","upper")
    return(est)
  }
```

```
rajzol <- function(mert,tipp,ev)
 {
    plot(mert,t="p",pch=20,las=1,col="blue",
     xlab="korosztály",ylab="halálozási ráta",
     main="Halálozási ráta 2017-ben")
    mtext(paste0("tény adat és LC becsült az 1950-",
              ev," évek alapján"),
              side = 3, line = .5)
    axis(1,at=seq(0,100,10))
    lines(tipp[,"mean"],col="red")
    lines(tipp[,"lower"],col="green4")
    lines(tipp[,"upper"],col="green4")
    abline(h=seq(0,.5,.1),
       v=seq(0,100,10),lty=3,col="gray")
    legend("topleft",c("tény", "becsült",
       "alsó 10%", "felső 10%"),
           col = c("blue","red","green","green"),
 text.col = "black", bg = "gray90", # szinek
           lty = c(-1, 1, 1, 1), pch = c(20, NA, NA, NA))
  }

hu.mort.2016 <- vago(hu.mort,1)
becs1 <- becsul(hu.mort.2016,1)
rajzol(teny,becs1,2016)
```

Először létrehoztunk egy `vago()` függvényt, mellyel az utolsó éveket tudjuk levágni annak megfelelően, hogy melyik évig szeretnénk felhasználni a meglévő adatainkat a becsléshez. Ebben az esetben az utolsó évet vágtuk le, így 2016-ig használjuk fel az éveket.

Ezután a `becsul()` függvénnyel elvégezzük a becslést a 2017-es évre. Ennél a résznél használjuk a megismert `lca()` és `forecast()` függvényeket. Végül létrehoztunk egy `rajzol()` függvényt, amellyel a kirajzolást végezzük el. Ezzel a módszerrel sikeresen megadhatjuk a mortalitási rátának becslését bármelyik évre, annak függvényében, meddig vesszük figyelembe az előző évek adatait.

A következő ábra a 2017-es évre mutatja az egyes korosztályok halálozási arány becslését, az 1950 – 2016-os évek alapján.



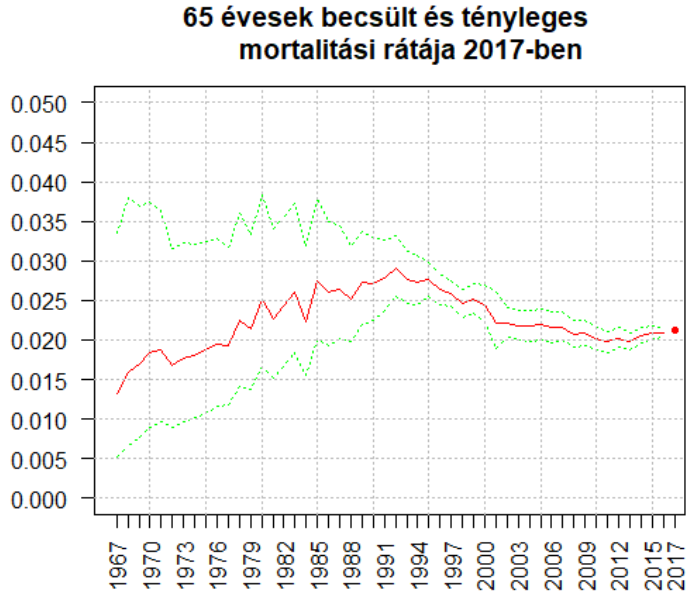
7. ábra. Korspecifikus halálozási arány 2017-re

A második ábrát a következő kódrészletben leírtakkal hoztuk létre.

```
mort65 <- matrix(NA,50,3)
rownames(mort65) <- 1967:2016
colnames(mort65) <- c("mean","lower","upper")

for (k in 50:1)
    mort65[51-k,] <- becsul(vago(hu.mort,k),k)[65+1,]

matplot(mort65,ylim=c(0,.05),ylab="",
        t="l",lty=c(1,3,3),col=c("red","green","green"),
axes=FALSE,frame=TRUE)
axis(1,at=1:50,label=rownames(mort65),las=3)
axis(2,at=seq(0,.05,.005),las=1)
```

```
abline(h=seq(0,.05,.005),v=seq(4,54,5),lty=3,col="gray")
points(51,teny[65+1],pch=20,col="red")
axis(1,at=51,label=2017,las=3)
title("65 évesek becsült és
       tényleges mortalitási rátája 2017-ben")
```

Ezen ábra elkészítéséhez szintén felhasználjuk az előzőekben létrehozott és megismert `becsül()` és `vago()` függvényeket. Majd a `matplot()` függvénnyel kirajzoljuk a megkapott becslést és a 2017-es év tényadatát.



8. ábra. 65 évesek becsült és tényleges mortalitási rátája 2017-ben

6. Összefoglalás

A bevezetésben leírtak alapján a dolgozatban bemutattam a Lee-Carter modell működését, megértéséhez és használatához szükséges fogalmakat. Részletesen bemutattam a szinguláris érték felbontást, mellyel a Lee-Carter modell k_t időkomponensét nyertük. Erre a paraméterre az **R** programnyelv '`forecast`' csomagjának segítségével különböző idősor modelleket illesztettünk, ezzel előrejelzéseket kaptunk a jövőre nézve a rendelkezésre álló adatokból.

Ezután a magyarországi halálozási ráta mélyebb elemzésére tértünk rá. Az **R**-project '`demography`' csomag `lca()` függvényének alkalmazásával két ábrát készítettünk el. Az első ábrán a korspecifikus halálozási arány becslése látható 2017-re, ha az adatokat $1950-2016$-ig vesszük figyelembe, a második ábrán pedig a 65 évesek mortalitási rátájának becslését, és a pontos adatot mutatja szintén a 2017-es évre.

A 7. ábrán feltüntettük a halálozási ráta becsült és tény-adatait is. Jól látszik, hogy körülbelül az 50-es korosztályig 0 közelében vannak az adatok. Ez a 70-es korig lassan emelkedik, majd ezt követően exponenciálisan növekszik meg. 90-től a becsült adatok már nagyobb eltérést mutatnak a tényadatokhoz képest, mint a korábbi korosztályoknál. Ennek oka a viszonylag alacsony esetszám.

A 8. ábrán a 65 évesek mortalitási rátájának 2017-es becsült és tényadata látszik, annak függvényében, hogy melyik évig vesszük figyelembe az ismert adatokat. Az ábrán a következő érdekes jelenséget figyelhetjük meg. Ha az $\widehat{r}_{x,t}(s)$ jelöli az x évesek t évbeli mortalitási rátájának a becslését az $1950-s$. évig tartó adatok alapján (ennek a függvénynek a görbéje a piros vonal), akkor e függvényértéke előbb közel megkétszereződik, majd egy szelíd csökkenést mutat. Az ábra alapján a 2000-es évektől elég nagy pontossággal sikerült megbecsülnünk a 65 évesek 2017-es halálozási arányát. Jól látszik, hogy minél több adat áll rendelkezésünkre a becslés elkészítéséhez, annál ki-

sebb szórással kapjuk meg azt. Érdekes, hogy konfidencia tartományok szélességének s függvényében vizsgált csökkenésében egy nyilvánvaló törés látható az 1992-es év körül.

Az ábrákon bemutatott konfidencia tartományok alapján egyértelműen látszik, hogy minél több adat áll a rendelkezésünkre, — értendő ez úgy is mint kronologikus mennyiség, azaz, hogy 'hány évnyi' adat, de úgy is mint vizsgált, rizikónak kitett populációs méret — annál pontosabb előrejelzést vagyunk képesek végezni a Lee-Carter modell alkalmazásával.

7. Függelék

Az 1., 2. és 3. ábrák elkészítésének menete az **R** programnyelvben:

```
 D<-read.delim("Mx_1x1_red.t",sep=";")
# Year Age   Female     Male    Total

names(D)<-c("Ev","Kor","NO","FF","TT")

Kor <- unique(D[,"Kor"]) # 0:110
Ev <- unique(D[,"Ev"]) # 1950:2017

mFF<-matrix(D[,"FF"],111,68)
mNO<-matrix(D[,"NO"],111,68)
mTT<-matrix(D[,"TT"],111,68)

k <- 13 # szintvonalak száma

# -
# a férfiak halandósága 1950-2017-közt
image(Kor,Ev,log(mFF),las=1,
      main="Férfi halandóság
a kor és év függvényében")
abline(v=seq(0,100,by=10),
 h=seq(1950,2010,by=10),
 col="white",lty=3)
contour(Kor,Ev,log(mFF),add=TRUE,
        vfont = c("sans serif", "bold"),
                nlevels=k,
labels=pretty(mFF, n=k),
labcex=1.5)
axis(1,seq(0,100,by=10))

# -
# a nők halandósága 1950-2017-közt
image(Kor,Ev,log(mNO),las=1,
      main="Női halandóság a
kor és év függvényében")
abline(v=seq(0,100,by=10),
       h=seq(1950,2010,by=10),
```

```
 col="white",lty=3)
contour(Kor,Ev,log(mNO),add=TRUE,
        vfont = c("sans serif", "bold"),
                nlevels=k,
labels=pretty(mNO, n=k),
labcex=1.5)
axis(1,seq(0,100,by=10))

# -
# az összlakosság halandósága 1950-2017

image(Kor,Ev,log(mTT),las=1,
      main="A magyarországi összlakosság
halandósága a kor és az év függvényében")
abline(v=seq(0,100,by=10),h=seq(1950,2010,by=10),
       col="white",lty=3)
contour(Kor,Ev,log(mTT),add=TRUE,
        vfont = c("sans serif", "bold"),
                nlevels=k,
labels=pretty(mTT, n=k),
labcex=1.5)
axis(1,seq(0,100,by=10))

# ----
# fine

rm(list=ls())

z <- outer(-9:25, -9:25)
levs <-

contour(z,col=1:4)

contour(z,levels=levs[-c(1,length(levs))],
        col=1:5,lwd=1:3*1.5,lty=1:3)

contor> par(op)

contor> ## Persian Rug Art:
contor> x <- y <- seq(-4*pi,4*pi,len=27)
```

```
contor> r <- sqrt(outer(x^2,y^2,"+"))

contor> opar <- par(mfrow=c(2,2),mar=rep(0,4))

contor> for(f in pi^(0:3))
contor+   contour(cos(r^2)*exp(-r/f),
contor+           drawlabels=FALSE,
                  axes=FALSE,frame=TRUE)
```

A ‘demography’ csomag `lca()` és `forecast()` függvényének struktúrája **R**-ben:

```
M <- lca(hu.mort,series="total",max.age=100,adjust="dt")
class(M) # lca
# ====================================================
str(M)
# List of 15
# $ label    : chr "HUNGARY"
# $ age      : korcsoportok 0-101
# $ year     : evek 1950-2017
# $ total    : egy 101x68 meretű tábla == az input
#              LÁSD=>[1]
# $ ax       : az "ax" korcsoport konstans
#              LÁSD=>[2]
# $ bx       : a "bx" korcsoport szorzo
#              LÁSD=>[2]
# $ kt       : a "kt" ido fuggveny
#              LÁSD=>[2]
# $ residuals: "Residuals mortality rate"
#                 == ln(az input) - $fitted
#   ..$ y    : a 101x68 méretű tábla (mert max.age=100)
#              LÁSD=>[4]
# $ fitted   : "Fitted mortality rate" == ax+bx*kt
#              LÁSD=>[3]
#   ..$ y    : a 101x68 méretű tábla
# $ varprop  : num 0.772==megmagyarázott variancia hányad
#              LÁSD=>[5]
# $ y        : "Mortality" == exp(M$fitted+M$residuals$y)
```

```
#               LÁSD=>[1]
#   ..$ y     : a 101x68 méretű tábla
# $ mdev      : Named num [1:2] 21.6 24.2
#               a célfüggvény minimuma
#               LÁSD=>[6]
#   ..- attr(*,"names")=chr[1:2]
#                "Mean deviance base" "Mean deviance total"
# $ call      : language lca(data=hu.mort,
#                            series="total",adjust="dt")
# $ adjust    : chr "dt"
# $ type      : chr "mortality"
# - attr(*, "class")= chr "lca"

# ----
# [1] "$total" és a "$y" a feldolgozott adat
dim(hu.mort$rate$total) # 111x68 ezt adtuk meg
dim(M$y$y)              # 101 68 a feldolgozott rész
                        #        az eredmény objektumban
dim(M$total)            # 101 68 a feldolgozott rész
                        #        az eredmény objektumban
EQ(M$total,M$y$y)       # TRUE az M objektumban ugyanaz
                        #      az adat kétszer...

# az első 100 sor ugyanaz, mint az input
EQ(M$total[1:100,],hu.mort$rate$total[1:100,]) # TRUE
# a 101. sor eltér az inputtol, mert az 100+ összesített
W <- hu.mort$rate$total[1:101,]
W[101,] <- colSums(hu.mort$rate$total[101:111,]
                   *hu.mort$pop$total[101:111,],
                   na.rm = TRUE)/
           colSums(hu.mort$pop$total[101:111,],na.rm=TRUE)
rate <- W
EQ(M$total,rate) # TRUE

# ----
# [2] az "$ax" "$bx" es "$kt"
W <- log(W)
ax <- apply(W,1,mean, na.rm=TRUE)
EQ(M$ax,ax) # TRUE
W <- sweep(W,1,ax)
```

```
bx <- svd(W)$u[,1] / sum(svd(W)$u[,1])
EQ(M$bx,bx) # TRUE
kt <- svd(W)$v[,1]* svd(W)$d[1] * sum(svd(W)$u[,1])
# piros az illesztett "kt", kek az svd-bol szamolt
plot(M$kt,t="b",ylim=c(-60,60),col="blue",pch=4)
points(1950:2017,kt,col="red",t="b",pch=20)

# ----
# [3] a "$fitted" erteke
EQ(M$fitted$y,M$ax+M$bx%*%matrix(M$kt,1)) # TRUE

# ----
# [4] a "$residuals" erteke
EQ(M$residuals$y,log(M$total)-M$fitted$y) # TRUE
EQ(M$total,      exp(M$fitted$y+M$residuals$y)) # TRUE

# ----
# [5] a "$varprop" megmagyarázott variancia hanyad
w <- svd(W)$d
c(M$varprop,w[1]^2/sum(w^2)) # 77.2%

# ----
# [6] a "$mdev" értéke: "Mean deviance base" és
                        "Mean deviance total"
pop<-hu.mort$pop$total[1:101,]
pop[101,] <- colSums(hu.mort$pop$total[101:111,])
m <- length(M$year)
n <- length(M$age)
deaths <- pop*rate

M.kt <- matrix(M$kt,1) # a modell szerinti "kt"
deaths.M <- exp(M$ax+M$bx%*%M.kt)*pop
            # death, ha a "kt" a modell szerinti
mdev.M <- sum(deaths*log(deaths/deaths.M)-
              (deaths-deaths.M))*2/((m-2)*(n-1))

L.kt <- matrix(mean(M.kt)+mean(diff(M$kt))
                *(1:m-(m+1)/2),1) # linearis "kt"
deaths.L <- exp(M$ax+M$bx%*%L.kt)*pop
            # death, ha a "kt" linearis volna
```

```
mdev.L <- sum(deaths*log(deaths/deaths.L)-
              (deaths-deaths.L))*2/((m-2)*n)

mdev <- c(mdev.M,mdev.L)
rbind(prog=M$mdev,calc=mdev)

#      Mean deviance base Mean deviance total
# prog           21.61064            24.15022
# calc           21.61064            24.15022

# ================================================

F <- forecast(M,jumpchoice="actual")
str(F)
#  $ label : chr "HUNGARY"
#  $ age   : korcsoportok 0-101
#  $ year  : elorejelzesi evek 1:50
#  $ rate  :List of 3
#   ..$ total: 101x50 méretű tábla
#   ..$ lower: 101x50 méretű tábla
#   ..$ upper: 101x50 méretű tábla
#  $ fitted:"Fitted mortality rate"
#            ugyanaz mint ami az lca eedmenye
#   ..$ y    : 101x68 méretű tábla
#  $ e0    : Time-Series [1:50] e0
#             Forecasts of life expectancies
#   (including lower and upper bounds)
#  $ kt.f  :List of 6
#   ..$ mean  : a kt forecastja 1:50
#   ..$ lower : a kt.min forecastja 1:50
#   ..$ upper : a kt.max forecastja 1:50
#   ..$ level : 80 a konfidencia szintje
#   ..$ x     : Time-Series [1:68] 1950 to 2017: 100.3
#   ..$ method: chr "Random walk with drift"
#   ..- attr(*, "class")= chr "forecast"
#  $ type  : chr "mortality"
#  $ lambda: num 0
#  $ model : az input modell adatai
#   ..$ label     : chr "HUNGARY"
#   ..$ age       : int [1:101] 0 1 2 3 4 5 6 7 8 ...
```

```
#   ..$ year      : int [1:68] 1950 1951 1952 1953 ...
#   ..$ total     : 101x68 méretű tábla
#   ..$ ax        : Named num [1:101] -3.98 -6.75 ...
#   ..$ bx        : Named num [1:101] 0.036 0.0341 ...
#   ..$ kt        : !!! ez eltér az inputtol !!! ???
#   ..$ residuals :"Residuals mortality rate"
#   .. ..$ y      : 101x68 méretű tábla
#   ..$ fitted    :"Fitted mortality rate"
#   .. ..$ y      : 101x68 méretű tábla
#   ..$ varprop   : num 0.772
#   ..$ y         :"Mortality"
#   .. ..$ y      : 101x68 méretű tábla
#   ..$ mdev      : Named num [1:2] 21.6 24.2
#   .. ..- attr(*, "names")= chr [1:2]
#                "Mean deviance base" "Mean deviance total"
#   ..$ call      : language lca(data=hu.mort,
#                     ser="total", max.age=100,adjust="dt")
#   ..$ adjust    : chr "dt"
#   ..$ type      : chr "mortality"
#   ..$ jumpchoice: chr "actual"
#   ..$ jumprates : Named num [1:101] 0.003491 ...
#  $ call  : language forecast.lca(object=M,
#                                      jumpchoice="actual")
#  $ name  : chr "total"
#  - attr(*, "class")= chr [1:2] "fmforecast" "demogdata"

EQ(F$fitted$y,M$fitted$y)# TRUE ugyanaz mint az inputban
EQ(F$fitted$y,F$model$fitted$y)# TRUE mégegyszer

EQ(M$total        , F$model$total      ) # TRUE
EQ(M$ax           , F$model$ax         ) # TRUE
EQ(M$bx           , F$model$bx         ) # TRUE
EQ(M$kt           , F$model$kt         ) # különbözö
EQ(M$residuals$y  , F$model$residuals$y) # TRUE
EQ(M$fitted$y     , F$model$fitted$y   ) # TRUE
EQ(M$varprop      , F$model$varprop    ) # TRUE
EQ(M$y$y          , F$model$y$y        ) # TRUE
EQ(M$mdev         , F$model$mdev       ) # TRUE

cbind(M$kt , F$model$kt ) # különbozo
```

8. Irodalom

[1] C. Eckart, G. Young, The approximation of one matrix by another of lower rank. Psychometrika, Vol. 1, pp 211–218, 1936.

[2] Ronald D. Lee and Lawrence R. Carter, Modeling and Forecasting U.S. Mortality, JASA, Vol. 87, No. 419, pp 659-671, 1992.

[3] R: A Language and Environment for Statistical Computing, R Core Team, R Foundation for Statistical Computing, https://www.R-project.org/, Vienna, Austria, 2020.

[4] demography: Forecasting Mortality, Fertility, Migration and Population Data, Rob J Hyndman with contributions from H. Booth and L. Tickle and J. Maindonald, R package version 1.22, https://CRAN.R-project.org/package=demography, 2019.

[5] GEP. Box, GM. Jenkins, GC. Reinsel, Time Series Analysis: Forecasting and Control, 4th Edition, Wiley, 2013.

[6] KSH Népességtudományi Kutatóintézet, www.demografia.hu, 2020.

[7] Human Mortality Database. University of California, Berkeley (USA), and Max Planck Institute for Demographic Research (Germany). Available at www.mortality.org or www.humanmortality.de 2020. november.

[8] R. Rau, Ch. Bohk-Ewald, MM. Muszyńska, JW. Vaupel: Visualizing Mortality Dynamics in the Lexis Diagram, Springer, 2018.

[9] ThR. Holford: The Estimation of Age, Period and Cohort Effects for Vital Rates Biometrics Vol. 39, No. 2., pp.311-324, 1983.

[10] RJ. Hyndman, Y. Khandakar: Automatic Time Series Forecasting: The forecast Package for R JSS Vol. 27, Iss. 3., pp.1-22, 2008.