

IBNR tartalékképzési módszerek megbízhatóságának összehasonlítása

MSc szakdolgozat

Juhász Jakab

Eötvös Loránd Tudományegyetem
Biztosítási és pénzügyi matematika MSc
Aktuárius szakirány

Témavezető:

Arató Miklós



Eötvös Loránd Tudományegyetem
Természettudományi Kar



Budapesti Corvinus Egyetem
Közgazdaságtudományi Kar

Budapest, 2016

Tartalomjegyzék

Bevezetés	3
1. A használt tartalékolási módszerek bemutatása	5
1.1 A szeparációs módszer	5
1.2 Egy másik, bővített szeparációs módszer	6
1.3 Stagnáló infláció módszere	7
1.4 Dupla lánclétra módszer	8
2. A munka leírása	9
2.1 A módszerek mellékprogramjai	9
2.1.1 Lánclétra módszer	9
2.1.2 Jéghegy módszer	10
2.1.3 Szeparációs módszer	13
2.2 Az adatok és a főprogram	16
2.2.1 A bootstrap	18
2.2.2 A tartalékképzési módszerek leprogramozása	21
3. Az eredmények	26
3.1 Az abszolút négyzetes hibák	26
3.2 A módszerek relatív hibái	29
3.3 Korrelációk	44
4. Összefoglalás	46
Irodalomjegyzék	48



Bevezetés

A lánclétra és a jéghegy módszerek régóta széles körben használatos tartalékolási módszerek az IBNR tartalék számításához az élet- és nem életbiztosításban egyaránt. Létezik azonban számos más módszer is, ami erre használható, mint például a főként a nem-életbiztosításban használatos szeparációs módszer. A dolgozatban ezen tartalékolási módszereket hasonlítom össze megbízhatósági szempontból egy nem-életbiztosítási ágból származó valós adatsoron, amit a jobb becslések érdekében bootstrappeléssel sokszorosítok. Középpontba helyezem a szeparációs módszert, de bemutatok ezen kívül több új módszert is, amivel kifutási háromszögekben tartalék számolható, majd ezeket több különböző módon összehasonlítom. A dolgozatban a kifutási háromszögekben a kifutási évek a kár bekövetkezése és a kár kifizetése közti időt jelentik, tehát az IBNR károk és a tételes függőkárok a dolgozatban külön nem szerepelnek. A dolgozatban az olvasó számára a következő fogalmakat ismertnek tételezem fel, és külön nem definiálom: kifutási háromszög, károk, kárszám, kárkifizetések, kumulált kifizetésszám, kumulált kifizetésösszeg, lánclétra módszer, jéghegy módszer.

Az első fejezet a szeparációs módszer leírásával kezdődik, majd bemutatja bővített szeparációs módszert a [2] cikket felhasználva. Ezután általam kitalált módszereket mutatok be a fejezet további részében, amiket stagnáló infláció módszere és dupla lánclétra módszer néven fogok emlegetni a dolgozatban, azonban fontos megjegyezni, hogy az itt szereplő dupla lánclétra módszer nem azonos a [6] cikkben szereplő Double Chain Ladder nevű módszerrel. A jéghegy és az egyszerű lánclétra módszert az olvasó által ismertnek feltételezem [1].

A második fejezet a dolgozat technikai része, itt az elemzés elkészítésének lépéseit írom le. A kezdeti adattömb egy 44735 kárkifizetés adatait tartalmazó táblázat, mely

kárkifizetések 43081 káreseményhez tartoznak. Ezekből a káreseményekből állítottam elő bootstrappelés segítségével 1000 darab forgatókönyvet, melyek mindegyikéből előállítottam a kifizetésszámokra és a kifizetésösszegekre vonatkozó kifutási háromszögeket, hogy a módszereket tesztelni tudjam. A megvalósításhoz Microsoft Excel-t használtam Visual Basic for Applications-el kiegészítve. A fejezet első felében bemutatom a használt tartalékszámítási módszerek programjait, ami segít az olvasónak a dolgozathoz tartozó Excel-fájl megértésében, ami a <https://sites.google.com/site/jakabjuhasz> weboldalon érhető el. A fejezet második felében a forgatókönyvek és az azokhoz tartozó kifutási háromszögek elkészítését mutatom be, ami a nagy mennyiségű adat miatt megfontolást igényelt.

A harmadik fejezet az eredmények ismertetése. Itt definiálom az abszolút és a relatív hibaérték fogalmát, melyek segítségével elemzem a módszereket. Az eredményeket ábrák teszik szemléletesebbé. Az elemzés eredményeként megmutatkozik, hogy nincsen a módszerek között olyan, ami egyértelműen a legjobb. Az eredeti, szakirodalmakból ismert, és széles körben használt módszerek mellett a stagnáló infláció módszere is viszonylag jó illeszkedést mutat az adatokra, és némely esetben a bővített szeparációs módszer is használható. Fontos megemlíteni, hogy léteznek olyan módszerek is az INBR tartalék becslésére, amelyek megát a kifutási években felmerülő kárkifizetések eloszlását becsülik, lásd [5], ez a dolgozat azonban ezt nem tárgyalja.

1. A felhasznált tartalékolási módszerek bemutatása

1.1 Szeperációs módszer

A szeperációs módszer azon a feltevésen alapul, hogy a kifizetésszámokat ismertnek tételezzük fel. Két inputja van a módszernek, az egyik az ismert kárszámok négyszöge, a másik a kárkifizetések kifutási háromszöge, és emellett még inflációs feltevéseink is vannak a következő $n - 1$ évre [1]. A módszernek az az alapötlete, hogy a kárkifizetéseket nem csak a kár évének és a kifizetés évének különbségéből becsülhetjük meg, hanem szerepet játszik a naptári évek közötti infláció is. A feltevés ésszerű, hiszen nem-életbiztosítási területen a biztosított tárgy vagy ingatlan átlagos értéke általában az üzletágbeli infláció szerint növekszik. A módszer a két hatás szétválasztásáról kapta a nevét. Ha a várható kárszámokat, tehát a kifutási háromszögen kívüli, de a négyszögön belüli értékeket nem ismerjük pontosan, akkor ezeket megbecsülhetjük lánclétra, vagy jéghegy módszerrel. Ezek alapján a módszer kimenete a várható kárkifizetések, a szükséges tartalékok, és a naptári évek közti üzletágbeli infláció lesz [3].

A szeperációs módszer alkalmazásának első lépése tehát, hogy megbecsüljük a kárszámokat a négyszögben. Erre lánclétra módszert használtunk. Miután megvannak az éves várható kárszámok, ezekkel leosztjuk a nem kumulált kifizetési háromszög (X) sorait. A kapott háromszög elemeit $P_{i,j}$ -vel jelöljük.

$$P_{i,j} = \frac{X_{i,j}}{n_i}$$

ahol n_i az i . év kárainak várható száma. Ezek a $P_{i,j}$ elemek azok, amiket szét akarunk választani kifizetési évtől függő és naptári évtől függő komponensekre, amik legyenek rend szerint r és λ , és mivel r egy arányossági tényező, ezért

$$\sum_{j=1}^t r_j = 1$$

aholt a maximális kifutási évek száma, ami a mi adatainkban 6. Így az adatainkat egy háromszögbe rendezhetjük, ami hasonlít egy nem kumulált kifutási háromszöghöz, azzal a különbséggel, hogy itt a háromszög sorai rendszerint le vannak osztva az adott bekövetkezési évre becsült kifizetések darabszámával. Ezek alapján ez a háromszög a következőképpen néz ki.

1. táblázat

kárév\kifutási év	1	2		t-1	t
1	$r_1 \lambda_1$	$r_2 \lambda_2$			$r_t \lambda_t$
2	$r_1 \lambda_2$	$r_2 \lambda_3$		$r_{t-1} \lambda_t$	
t-1		$r_2 \lambda_t$			
t	$r_1 \lambda_t$				

Látható, hogy $\hat{\lambda}_t$ -t könnyen megkaphatjuk, ha az alsó átló elemeit összeadjuk, mert

$$(r_1 + r_2 + \dots + r_t)\lambda_t = \sum_{j=1}^t P_{j,t+1-j} = \hat{\lambda}_t$$

valamint

$$\hat{r}_t = \frac{P_{1,t}}{\hat{\lambda}_t}$$

és

$$\hat{\lambda}_{t-1} = \frac{\sum_{j=1}^{t-1} P_{j,t-j}}{1 - \hat{r}_t}, \quad \hat{r}_{t-1} = \frac{P_{1,t-1} + P_{2,t-1}}{\hat{\lambda}_t + \hat{\lambda}_{t-1}}$$

Ezt az eljárást folytatva az összes együttható meghatározható.

$$\hat{\lambda}_{t-i} = \frac{\sum_{j=1}^{t-i} P_{j,t-j}}{1 - \hat{r}_t - \dots - \hat{r}_{t-i+1}}, \quad \hat{r}_{t-i} = \frac{P_{1,t-i} + P_{2,t-i} + \dots + P_{i+1,t-i}}{\hat{\lambda}_t + \hat{\lambda}_{t-1} + \dots + \hat{\lambda}_{t-i}}$$

Így már a $P_{i,j}$ háromszögben az összes λ -t ismerjük, azonban ahhoz, hogy a szükséges tartalékokat meg tudjuk határozni, ismernünk kell az egész négyyszöget. Ez úgy lehetséges, ha megadjuk $\lambda_{t+1}, \lambda_{t+2}, \dots, \lambda_{2t-1}$ értékét valamilyen inflációs előrejelzés segítségével. Az i . év és az $i + 1$. év közötti százalékos infláció $\left(\frac{\lambda_{i+1}}{\lambda_i} - 1\right) * 100\%$, tehát ha a t . és a $2t - 2$. évre vannak előrejelzéseink a tárgyév, és az azt követő év közötti inflációra, amik legyenek rendre $i_t, i_{t+1}, \dots, i_{2t-2}$, akkor a $\lambda_{t+1}, \lambda_{t+2}, \dots, \lambda_{2t-1}$ értékek a $\lambda_i = \lambda_{i-1} \left(1 + \frac{i_{i-1}}{100\%}\right)$ képlettel számolhatók. Ezek segítségével már az egész négyyszög előállítható, és ha a sorait visszaszorozzuk n_i -vel, akkor már egyszerűen megkapjuk a szükséges tartalékokat.

1.2 Egy másik, bővített szeparációs módszer

Ennek a módszernek azonban elképzelhető egy kicsivel bonyolultabb változata is, ugyanis érdemes megvizsgálni azt a kérdést, hogy mi van, ha a kárnagyságok nem csak a kifizetés évétől, hanem a károk bekövetkezésének évétől is függenek. Ezt a következőképpen írhatjuk fel.

$$X_{i,j} = n_i r_j q_i \nu_{i+j-1}$$

$$\sum_{j=1}^t r_j = 1, \quad \sum_{i=1}^t q_i = 1$$

ahol ν csak a naptári évtől függő inflációs komponens, q pedig csak a bekövetkezési évektől függő inflációs komponens [2].

2. táblázat

kárév\kifutási év	1	2	3		t-1	t
1	$r_1 q_1 \gamma_1$	$r_2 q_1 \gamma_2$	$r_3 q_1 \gamma_3$			$r_t q_1 \gamma_t$
2	$r_1 q_2 \gamma_2$	$r_2 q_2 \gamma_3$	$r_3 q_2 \gamma_4$		$r_{t-1} q_2 \gamma_t$	
3	$r_1 q_3 \gamma_3$	$r_2 q_3 \gamma_4$				
t-1		$r_2 q_{t-1} \gamma_t$				
t	$r_1 q_t \gamma_t$					

Induljunk el ugyanúgy, mint az előbb. Számoljuk ki szintén az $\hat{r}_1, \hat{r}_2, \dots, \hat{r}_j, \hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_t$ együtthatókat, azonban most elég ennyi. Az inflációs várakozásokkal most nem számolunk tovább. Nekünk most csak az r_j -k kellenek. Ezekkel leosztjuk a kezdeti $P_{i,j}$ háromszög oszlopait, majd a kapott háromszöget transzponáljuk.

$$\hat{R}_{i,j} := \left(\frac{P_{i,j}}{\hat{r}_j} \right)^T$$

Most $\hat{R}_{i,j}$ -re is elvégezzük az előbbi szeparációs módszert, legyen $\hat{R}_{i,j} = \hat{q}_j \hat{\gamma}_{i+j-1}$. Ez alapján kijönnek a \hat{q} és a $\hat{\gamma}$ együtthatók. Az $R_{i,j}$ négyzet előállításához már megint szükség van az inflációs várakozásainkra. Ezek alkalmazásával a nem kumulált kárkifizetések négyzöge

$$\hat{X}_{i,j} = n_i \hat{r}_j \hat{R}_{i,j}^T$$

1.3 Stagnáló infláció módszere

Egy szokásostól eltérő, általam alkotott módszert is lefuttattam az adatokon. Ennek egyik feltevése, hogy hosszú idő átlagában az infláció stagnál, vagyis állandó. Másik két feltevése, hogy a kárkifizetések és a kárszámok négyzetében a sorok mindkettőben függetlenek egymástól, és a sorokban az elemek aránya nem függ az adott sor számától. Bemeneti adataink a nem kumulált kárszámok és a nem kumulált kárkifizetések háromszöge. Ezek legyenek $N_{i,j}$ és $X_{i,j}$. Ezekből kell meghatároznunk az inflációs komponenseket a következőképpen.

$$\lambda_i = \frac{\sum_{k=1}^i X_{k,i-k+1}}{\sum_{k=1}^i N_{k,i-k+1}}$$

Ezzel a képlettel kiszámoljuk a $\lambda_1, \lambda_2, \dots, \lambda_t$ komponenseket, majd ezekből a $\hat{\lambda}_{t+1}, \hat{\lambda}_{t+2}, \dots, \hat{\lambda}_{2t-1}$ komponenseket az alábbi képlet szerint.

$$\hat{\lambda}_{t+i} = z \left(\frac{z}{y} \right)^{i-1}$$

aholy $z = \min_{i \leq t} \lambda_i$ és $y = \max_{i \leq t} \lambda_i$.

Most lánclétra vagy jéghegy módszerrel becsüljük meg a kárszám négyszöget, az $\hat{N}_{i,j}$ együtthatókat. Ezek használatával a kárkifizetéseket a $\hat{X}_{i,j} = \hat{N}_{i,j} \hat{\lambda}_{i+j-1}$ képlettel becsülhetjük.

A módszer úgy is módosítható, hogy $y = \lambda_1$ és $z = \lambda_t$, vannak esetek, amikor ez mutat jobb illeszkedést, és persze gyakran egyébként is $\lambda_1 = \min_{i \leq t} \lambda_i$ és $\lambda_t = \max_{i \leq t} \lambda_i$, tehát a két eset gyakran ugyanaz. Érdemes azonban megjegyeznünk, hogy bizonyos biztosítási ágazatoknál gyakori, hogy a nagyobb károkat később fizetik ki különböző okok miatt. Ilyen esetben ez a módszer nem ad pontos eredményt.

1.4 Dupla lánclétra módszer

Ez a módszer viszonylag egyszerű. A nem kumulált kárkifizetések háromszögének elemeit elosztjuk a nem kumulált kifizetésszámok háromszögének elemeivel, így megkapjuk az átlagos kifizetések háromszögét, ez legyen P .

$$P_{i,j} = \frac{X_{i,j}}{N_{i,j}}$$

Ezután lánclétra módszerrel kiszámoljuk a P és az N négyszöget, majd a P négyszög elemeit megszorozzuk N elemeivel, és így megkapjuk a becsült $X_{i,j}$ négyszöget, amiből megállapítjuk a szükséges tartalékot. Fontos megjegyezni, hogy a módszer nem azonos a [6] cikkben tárgyalt Double Chain Ladder módszerrel.

2. A munka leírása

2.1 A módszerek mellékprogramjai

Munkám célja a hagyományos módszerek, a szeparációs és a jéghegy módszer tesztelése, megbízhatóságának összevetése a szeparációs módszerrel, és két másik, általam kidolgozott módszerrel. Első lépésként ezeket a módszereket Visual Basic for Application-ben leprogramoztam. Először mindegyiket egy külön fájlba, majd ezeket átírtam a főfájlba, ahol mind az 1000 forgatókönyvre lefuttattam őket. A kódok a következők:

2.1.1 Lánclétra módszer

```
Sub chainladder()
```

```
t = Cells(2, 2)
```

```
For j = 1 To t
```

```
Range("b").Cells(1, j) = 0
```

```
For i = 1 To t - j + 1
```

```
Range("b").Cells(1, j) = Range("b").Cells(1, j) + Range("Y").Cells(i, j)
```

```
Next i
```

```
Next j
```

```
For j = 1 To t
```

```
Range("a").Cells(1, j) = 0
```

```
For i = 1 To t - j
```

```
Range("a").Cells(1, j) = Range("a").Cells(1, j) + Range("Y").Cells(i, j)
```

```
Next i
```

```
Next j
```

```
For j = 1 To t - 1
```

```
Range("f").Cells(1, j) = Range("b").Cells(1, j + 1) / Range("a").Cells(1, j)
```

```
Next j
```

```
For i = 2 To t
```

```
For j = t - i + 2 To t
```

```
Range("Y").Cells(i, j) = Range("Y").Cells(i, j - 1) * Range("f").Cells(1, j - 1)
```

```
Next j
```

```
Next i
```

```
End Sub
```

Y jelöli a kumulált kárkifizetéseket, f a szokásos f együtthatókat. Az f együtthatók előállításához használunk még két együtthatót. Ab_j együttható az f_{j-1} meghatározásához szükséges j . oszlop elemeinek az összege a mellékátlóig, az a_j pedig az f_j meghatározásához szükséges j . oszlop elemeinek az összege csak a mellékátló feletti elemeket belevéve, és a mellékátlót már nem belevéve. Ennek alapján $f_j = \frac{b_{j+1}}{a_j}$. A háromszög mérete a programban paraméterként szerepel (t), ami lehetővé teszi, hogy a programot tetszőleges méretű háromszögre futtassuk. Az alábbi ábrán kék háttérrel a meglévő és pirossal a becsült adatok szerepelnek.

3. táblázat

Y	1	2	3	4	5	6
1	126 747 199	727 975 742	1 025 379 121	1 190 577 574	1 870 992 542	2 276 089 908
2	62 428 642	513 566 656	974 260 455	1 129 906 975	1 452 807 589	1 767 361 770
3	53 710 018	636 235 705	1 012 527 448	1 275 164 718	1 826 511 906	2 221 978 561
4	92 060 095	679 626 274	1 208 495 840	1 442 591 711	2 066 329 862	2 513 720 627
5	64 411 976	829 810 084	1 369 493 474	1 634 775 950	2 341 609 436	2 848 602 272
6	90 307 344	765 955 346	1 264 109 546	1 508 978 262	2 161 420 186	2 629 399 403
b	489665274	3387214461	4220662864	3595649267	3323800131	2276089908
a	399357930	2557404377	3012167024	2320484549	1870992542	0
f	8,481650686	1,650369766	1,193708462	1,432373309	1,216514688	

2.1.2 Jéghegy módszer

A jéghegy módszerben a d együtthatóknak különböző súlyozást kell adni, én az átlagra és a minimumra terjesztettem ki a programot. Ezen kívül megadható egy m (memory) érték, ami megmondja, hogy hány korábbi év tapasztalatait használjuk fel a d együtthatók meghatározásához. Ezt az Excel fájl egyik cellájába kell megadni. Lehet 0 is.

4. táblázat

	1	2	3	4	5	6
-5	30020	45150	52300	56860	58740	58780
-4	33400	49070	56570	62070	65110	65300
-3	38400	58000	68000	74560	77540	77760
-2	43310	63110	72000	78700	83020	83480
-1	41000	61130	72450	77890	80100	80210
0	43200	64240	74050	80050	85650	85850
1	45350	67850	78650	86670	89870	90147
2	51670	85890	94820	103580	108477	108811
3	56450	86390	100890	109830	115023	115378
4	55600	78600	90491	98510	103168	103485
5	59100	88821	102259	111320	116584	116943

A fenti halványzöld háttérű négyszög tartalmazza a korábbi évek kumulált kifizetéseit, ez segítheti a becslésünket, ha számolunk vele. A kék háromszög a mostani kumulált kifizetési háromszög, és a piros háromszög a becsült adatok. Most 2-es m értékkel számoltunk, tehát a

lenti *d* tartományban felfelé a -2-ig lesznek meg az értékek. A *rate* sorban pedig ezeknek az együttthatóknak jelen esetben az átlaga szerepel.

5. táblázat

d	1	2	3	4	5	6
-5						
-4						
-3						
-2	0,518807	0,755989	0,862482	0,942741	0,99449	
-1	0,511158	0,762124	0,903254	0,971076	0,998629	
0	0,503203	0,748282	0,862551	0,93244	0,99767	
1	0,503068	0,752661	0,872466	0,961432		
2	0,474858	0,789347	0,871416			
3	0,489263	0,748759				
4	0,537274					
5						
rate	0,505376	0,759527	0,874434	0,951922	0,99693	1

Az Excel fájlban a zöld háttérű tartományt *N*-nel, az alatta lévő kék és piros tartomány pedig *Y*-nal jelöljük.

Sub iceberg()

t = Cells(2, 2)

m = Cells(3, 2)

For j = 1 To t - 1

For i = 6 - m To 6

Range("d").Cells(i, j) = Range("N").Cells(i, j) / Range("N").Cells(i, t)

Next i

Next j

If Cells(2, 9) = 1 Then

For j = 1 To t - 1

Range("rate").Cells(1, j) = 0

For i = 6 - m To 6

Range("rate").Cells(1, j) = Range("rate").Cells(1, j) + Range("d").Cells(i, j)

Next i

Range("rate").Cells(1, j) = Range("rate").Cells(1, j) / (m + 1)

Next j

End If

If Cells(2, 9) = 0 Then

For j = 1 To t - 1

Range("rate").Cells(1, j) = Range("d").Cells(1, j)

For i = 2 To m + 1

If Range("rate").Cells(1, j) > Range("d").Cells(i, j) Then

Range("rate").Cells(1, j) = Range("d").Cells(i, j)

Else

Range("rate").Cells(i, j) = Range("rate").Cells(i, j)

End If

Next i

Next j

End If

Range("rate").Cells(1, t) = 1

If Cells(2, 9) = 1 Then

For j = 1 To t - 1

Range("rate").Cells(1, t - j) = 0

For i = 6 - m To 5 + j

Range("rate").Cells(1, t - j) = Range("rate").Cells(1, t - j) + Range("d").Cells(i, t - j)

Next i

Range("rate").Cells(1, t - j) = Range("rate").Cells(1, t - j) / (m + j)

For i = 0 To j - 1

Range("Y").Cells(j, t - i) = Range("Y").Cells(j, t - j) * Range("rate").Cells(1, t - i) / Range("rate").Cells(1, t - j)

Next i

For i = 1 To t - 1 - j

Range("d").Cells(t + j, i) = Range("Y").Cells(j, i) / Range("Y").Cells(j, t)

Next i

Next j

End If

If Cells(2, 9) = 0 Then

For j = 1 To t - 1

Range("rate").Cells(1, t - j) = Range("d").Cells(6 - m, t - j)

```

For i = 7 - m To 5 + j
If Range("rate").Cells(1, t - j) > Range("d").Cells(i, t - j) Then
Range("rate").Cells(1, t - j) = Range("d").Cells(i, t - j)
Else
Range("rate").Cells(i, t - j) = Range("rate").Cells(i, t - j)
End If
Next i
For i = 0 To j - 1
Range("Y").Cells(j, t - i) = Range("Y").Cells(j, t - j) * Range("rate").Cells(1, t - i) / Range("rate").Cells(1, t - j)
Next i
For i = 1 To t - 1 - j
Range("d").Cells(t + j, i) = Range("Y").Cells(j, i) / Range("Y").Cells(j, t)
Next i
Next j
End If

For i = 1 To t - 1
Range("Y").Cells(i, t) = Range("Y").Cells(i, t - i) / Range("rate").Cells(1, t - i)
Next i

For i = 2 To t - 1
For j = t - i + 1 To t - 1
Range("Y").Cells(i, j) = Range("Y").Cells(i, t) * Range("rate").Cells(1, j)
Next j
Next i

End Sub

```

2.1.3 Szeparációs módszer

A háromszög mérete itt is paraméterként szerepel. Első dolgunk a P háromszög meghatározása.

```

Sub atlagkar()
t = Cells(2, 2)
For j = 1 To t
For i = 1 To t
Range("P").Cells(i, j) = Range("X").Cells(i, j) / Range("n").Cells(i, 1)

```

```
Next i
Next j
End Sub
```

X jelöli a nem kumulatív kárkifizetéseket, n pedig az adott naptári évben várható kárszámot, amit például lánclétrával becsülhetünk. Ezután kezdhetjük a paramétereket meghatározni.

```
Sub parameterek3()
```

```
t = Cells(2, 2)
```

```
For k = 1 To t + 1
```

```
For j = 1 To 8
```

```
Range("M").Cells(k, j) = 0
```

```
Next j
```

```
Next k
```

```
For k = 1 To t
```

```
For j = 1 To k
```

```
Range("lambda1").Cells(k, 1) = Range("lambda1").Cells(k, 1) + Range("P").Cells(j, 1 + k - j)
```

```
Next j
```

```
For j = 1 To k
```

```
Range("rate1").Cells(t + 1 - k, 1) = Range("rate1").Cells(t + 1 - k, 1) + Range("P").Cells(j, t + 1 - k)
```

```
Next j
```

```
Next k
```

```
Range("lambda").Cells(t, 1) = Range("lambda1").Cells(t, 1)
```

```
Range("rate2").Cells(t, 1) = Range("lambda").Cells(t, 1)
```

```
Range("lambda2").Cells(t, 1) = 1
```

```
Range("sumrate").Cells(t + 1, 1) = 0
```

```
Range("rate").Cells(t, 1) = Range("P").Cells(1, t) / Range("lambda1").Cells(t, 1)
```

```
Range("sumrate").Cells(t, 1) = Range("rate").Cells(t, 1)
```

```
For k = 1 To t - 1
```

```
Range("lambda2").Cells(t - k, 1) = 1 - Range("sumrate").Cells(t + 1 - k, 1)
```

```

Range("lambda").Cells(t - k, 1) = Range("lambda1").Cells(t - k, 1) / Range("lambda2").Cells(t - k, 1)
Range("rate2").Cells(t - k, 1) = Range("rate2").Cells(t + 1 - k, 1) + Range("lambda").Cells(t - k, 1)
Range("rate").Cells(t - k, 1) = Range("rate1").Cells(t - k, 1) / Range("rate2").Cells(t - k, 1)
Range("sumrate").Cells(t - k, 1) = Range("sumrate").Cells(t + 1 - k, 1) + Range("rate").Cells(t - k, 1)
Next k
End Sub

```

Kijelölünk egy M tartományt, ahová az együtthatókat fogjuk kiszámolni. A λ és az r együtthatókat hányadosokként fogjuk megkapni, ezért a könnyebb átláthatóság kedvéért ezeknek külön kiírjuk a számlálóját és a nevezőjét. Az M tartomány tehát a program futtatása után így néz ki:

6. táblázat

rate	lambda	rate1	rate2	lambda1	lambda2	sumrate
0,030951	575310,2	66206,95	2139071	17806,57	0,030951	1
0,262254	325896,5	410102,3	1563761	95554,51	0,293205	0,969049
0,194449	265550,7	240701,9	1237865	129497	0,487654	0,706795
0,090444	343236,7	87939,65	972313,9	198424,5	0,578098	0,512346
0,243125	310738,1	152944,3	629077,1	255185,3	0,821223	0,421902
0,178777	318339	56911,68	318339	318339	1	0,178777
	343806,1					
	357558,3					
	368285,1					
	375650,8					
	379407,3					

A $rate(r)$, és $lambda(\lambda)$ változók számlálói és nevezői rend szerint a $rate1$, $rate2$, $lambda1$ és $lambda2$ oszlopokban találhatóak, a $sumrate$ oszlop pedig az r -ek összegét mutatja az utolsótól visszafelé összeadva. A $lambda$ oszlop pirossal kiemelt részében találhatóak az inflációs előrejelzésünk alapján számolt λ értékek, amik már a P négyzet kiszámításához kellenek. Ezt állítja elő a $lambdaforecast()$ nevű alprogram. Az $infl()$ alprogram pedig az adatokból előállítható $lambda$ -kból számol inflációt a múltbeli évekre. Ezt a $deflator$ tartományba írja a program, és ennek a tartománynak az utolsó $t - 1$ darab cellájába kell beírniunk inflációs várakozásainkat. Fontos hangsúlyozni, hogy a múltbeli évekre a módszer becsüli meg az inflációs mutatókat, viszont a jövőbeli évekre való inflációs várakozásaink inputként szolgálnak a módszernek ahhoz, hogy a jövőbeli tartalékokat kiszámolja. Az $atlagkar2()$ alprogram, ami a Q tartományt tölti ki, a P négyzetet állítja elő, a $reserve()$ pedig már a becsült tartalékokat adja.

```
Sub infl()
```

```
t = Cells(2, 2)
```

```
For i = 1 To t - 1
```

```

Range("deflator").Cells(i, 1) = Range("lambda").Cells(i + 1, 1) / Range("lambda").Cells(i, 1) - 1
Next i
End Sub

```

```

Sub lambdaforecast()
t = Cells(2, 2)
For i = t + 1 To 2 * t - 1
Range("lambda").Cells(i, 1) = Range("lambda").Cells(i - 1, 1) * (1 + Range("deflator").Cells(i - 1, 1))
Next i
End Sub

```

```

Sub atlegkar2()
t = Cells(2, 2)
For j = 1 To t
For i = 1 To t
Range("Q").Cells(i, j) = Range("rate").Cells(j, 1) * Range("lambda").Cells(i + j - 1, 1)
Next i
Next j
End Sub

```

```

Sub reserve()
t = Cells(2, 2)
For j = 1 To t
For i = 1 To t
Range("res").Cells(i, j) = Range("Q").Cells(i, j) * Range("n").Cells(i, 1)
Next i
Next j
End Sub

```

2.2 Az adatok és a főprogram

A kiindulási táblázatunk 43081 káresemény adatait tartalmazza, minden kifizetéshez külön sora van, így összesen van 44735 kárkifizetésünk, ennyi sora van a táblázatunknak. Öt oszlopunk van, az elsőben a káresemények sorszáma szerepel, 1-től 43081-ig. Az azonos káreseményekhez tartozó sorok azonos sorszámmal szerepelnek. A *claim ID* a károk azonosítószámait tartalmazza, azonos károkhoz tartozó kifizetésekhez azonos azonosító tartozik. A *claimperiod* oszlopban a kár bekövetkezési éve található, jelen esetben 1-től 6-ig. A *reportperiod* oszlopban a kár bejelentési éve található, szintén 1-től 6-ig. Ezekben az oszlopokban egy azonosítóhoz egyféle érték tartozik. A *paymentperiod* oszlopban az adott

azonosítóhoz tartozó kifizetések évei vannak, a *payment* oszlopban pedig ezek összegei. Egy azonosítóhoz többféle érték is tartozhat ebből a két oszlopból, hiszen lehetséges, hogy némelyik kárt nem egyszerre fizette ki a biztosító, hanem több különböző összeget fizetett több különböző évben. Az alábbi ábrán ennek a hosszú táblázatnak csak a legeleje látható, az egészet csak a dolgozathoz mellékelt program tartalmazza.

7. táblázat

Claim_ID	claimperiod	reportperiod	paymentperiod	payment
1	1	1	1	137666
2	1	1	1	104658
3	1	1	1	431662
4	1	1	1	125109
5	1	1	1	115653
6	1	1	1	42977
7	1	1	1	116028
8	1	1	1	100908
9	1	1	1	147519
10	1	1	1	228457
11	1	1	1	332772
12	1	1	2	130936
13	1	1	1	270189
14	1	1	3	483485
15	1	1	2	37896
16	1	1	1	2359754
17	1	1	2	36736

Az első feladatunk ezen adatok használhatóvá tétele, vagyis négyszögekbe rendezése. Ehhez elsőként a károkat bekövetkezési évük és a kifizetés évük szerint csoportosítanunk kell kárszám és kárösszeg szerint. Kifizetés évéből azonban több is van, tehát valójában a kárkifizetéseket kell csoportosítanunk. Ez egy alkalommal még könnyel elvégezhető makróírás nélkül is, azonban ahhoz, hogy a módszerek pontosságát tesztelni tudjuk, ez nem elég. Mivel más adataink nincsenek, ebből az adatsorból kell valahogy többet gyártnunk, hogy több mintát vehessünk a módszerek teszteléséhez. Célunk, hogy makróban leprogramozzuk a tartalékolási módszereket, majd az adatokból képzett háromszögekre lefuttassuk őket. a háromszögeket csak szimplán a négyszögekből képezzük a fölös elemek törlésével, majd az adott módszer által képzett négyszöget összehasonlítjuk az eredetivel, és kiszámoljuk a hiba várható értékét és szórását az összes mintaelemre, amiket valamilyen módon adatsorunkból képzünk. Csinálhatnánk azt is, hogy egyszerűen csak szétdaraboljuk az adatsort, és mindegyik darabra lefuttatjuk a módszereinket, ez azonban több szempontból sem egy elegáns megoldás. Egyrészt csökken a mintánkénti kárarabszám, így nő a hiba szórása, másrészt pedig nem lehetünk biztosak benne, hogy adataink a darabok között

homogének lesznek, ami tovább rontja módszereink pontosságát. Ezért ésszerű választás a következő eljárás.

2.2.1 A bootstrap

Visszatevéses húzással válasszunk ki az adatsorunkból a káresemények közül egy megadott számút. Ezt ismételjük meg kellően sokszor. Ilyen módon gyártható egy adatsorból több, amelyek rendelkeznek az eredeti adatsor tulajdonságaival. Ezt az Excelben a következő módon oldhatjuk meg. Generáljunk adott mennyiségű véletlen egész számot az [1;43081] intervallumon, majd a véletlen számokkal egyenlő sorszámú károkra vonatkozó kifizetéseket rendezzük négyeszőgekbe darabszám és összeg szerint, és ezt valahányszor ismételjük meg. Mivel a feladat nagy műveletigényű, az sem mindegy, hogy milyen programot, és hogyan használunk a megoldására. A problémát mégis Excelben oldottam meg, azonban a program beépített függvényei általában túl nagy memóriaigényűek, ezért nem minden esetben használhatóak egy ilyen méretű feladathoz. A probléma beépített függvénnyel például a következő módon oldható meg:

```
Sub bootstrap2()  
Dim oszlop(1 To 43081)  
For j = 1 To 1000  
For i = 1 To 43081  
oszlop(i) = Int(Rnd() * 43081 + 1)  
Range("ran").Cells(i, 1) = oszlop(i)  
Next i  
For i = 1 To 44735  
Range("coef").Cells(i, 1) = WorksheetFunction.CountIf(Range("ran"), Range("payments").Cells(i, 1))  
Next i  
For i = 1 To 44735  
Range("Z").Cells(i, j) = Range("payments").Cells(i, 5) * Range("coef").Cells(i, 1)  
Next i  
Next j  
End Sub
```

Az 5. sor generálja a véletlen számokat az [1;43081] intervallumon, a 8 – 10 sorok megnézik, hogy a sorszámok hányszor szerepelnek a kiválasztottak között, ez a szám lesz a coef oszlop. A 11-13. sorok az adott sorszámokhoz tartozó kifizetéseket megszorozzák az adott coef-beli együtthatóval. Ezt ismétli a program 1000-szer, így létrejön egy 1000 adatsorból álló minta. Egy ekkora mintán már kellően pontos vizsgálatot tudunk végezni.

Mielőtt egy ilyen hosszú összetett ciklusokból álló programot elindítunk, először érdemes rövidebb ciklusokkal futtatni, hogy lássuk, megfelelően működik-e, és hogy meg tudjuk becsülni a futási idejét. Ezzel a programmal pedig több probléma is van. A legfontosabb,

hogy futási ideje körülbelül 8 nap, ami érdemessé teszi más megoldás keresését. Emellett még az sem szerencsés, hogy a kárszámokkal nem foglalkozik, ezért így csak az alapmintára vonatkozó kárszámokat tudjuk használni, ami szintén a pontosság becslésének romlásához vezet.

Az következő ötlet az, hogy a *ran* oszlopot növekvő sorrendbe rendezzük. A kársorszámok is növekvő sorrendben vannak, tehát ebben az esetben elég, ha csak végigmegyünk a kársorszámokon, és ha találunk egyezést a *ran* oszloppal, akkor az adott kárkifizetést lejegyezzük, és annyiszor vesszük, ahányszor a sorszám szerepel a *ran* oszlopban, a kárkifizetésszám-négyszög adott celláját pedig növeljük 1-gyel. Ebből következik, hogy ha jól írjuk meg a programot, akkor az azonos kársorszámhoz tartozó kifizetések ugyanazzal az együtthatóval lesznek szorozva, és ha egy sorszám többször szerepel a *ran* oszlopban, akkor az ahhoz tartozó kifizetések annyival vannak szorozva, ahányszor szerepel. Mivel a *ran* oszlop most sorrendben van, ezért könnyebb lesz a szűrőpróba szerű ellenőrzés, mint az előbbi esetben. Ez a program a következőképpen néz ki:

```
Sub bootstrap4()
```

```
Dim x
```

```
Dim y
```

```
Dim k
```

```
Dim p
```

```
Dim q
```

```
Dim oszlop(1 To 43081)
```

```
For j = 1 To 1000
```

```
x = 1
```

```
For i = 1 To 43081
```

```
oszlop(i) = Int(Rnd() * 43081 + 1)
```

```
Range("ran").Cells(i, 1) = oszlop(i)
```

```
Next i
```

```
Dim oneRange As Range
```

```
Dim aCell As Range
```

```
Set oneRange = Range("ran")
```

```
Set aCell = Range("G12")
```

```
oneRange.Sort Key1:=aCell, Order1:=xlAscending, Header:=xlNo
```

```
For i = 1 To 43081
```

```
y = Range("ran").Cells(i, 1)
```

Do Until Range("payments").Cells(x, 1) >= y

x = x + 1

Loop

k = 0

Do While Range("payments").Cells(x + k, 1) = y

k = k + 1

Range("Z").Cells(x + k - 1, j) = Range("Z").Cells(x + k - 1, j) + Range("payments").Cells(x + k - 1, 5)

p = Range("payments").Cells(x + k - 1, 2)

q = Range("payments").Cells(x + k - 1, 4)

*Range("losses").Cells(1 + (j - 1) * 10 + p, 2 + q - p) = Range("losses").Cells(1 + (j - 1) * 10 + p, 2 + q - p) + 1*

Loop

Next i

Next j

A 13-17- sorok rendezik növekvő sorrendbe a *ran* oszlopot. Az *x* változó végig halad a kárkifizetéseken, és ha elérte az *y* változót, ami kezdetben a *ran* oszlop első eleme, akkor a *k* változó végig halad az *y*-nal egyenlő sorszámú kifizetéseken, és a hozzájuk tartozó összegeket hozzáadja a *Z* mező adott oszlopában ahhoz a kifizetéshez tartozó cellához, majd ha ez megtörtént, akkor az *y* változó a *ran* oszlop következő elemére ugrik, *x* pedig halad tovább eddig az elemig, és ez így megy, amíg *x* végig nem ér az egész kárkifizetés állományon. Ha ez megtörtént, akkor új mintaelemet kezd el készíteni, újra generál véletlen számokat a *ran* oszlopba, és a *Z* tartomány következő oszlopát tölti ki kifizetésekkel az előbbieket szerint, amíg végére ér. Eközben a *losses* tartományban minden mintaelem készítésekor egy új kifizetésszám-négyszöget tölt ki, így végül 1000 db kifizetésszám-négyszög lesz, amik a *Z* tartomány adott oszlopaihoz tartoznak.

Ez a program már futásidejét tekintve sokkal gazdaságosabb az előbbinél. Futásideje mindössze nagyjából 6 óra. A rendező algoritmus átlagosan $O(n \cdot \log n)$ idejű, ahol a jelen esetben $n = 43081$, tehát a program valamivel több, mint $(\log_2 43081) \cdot 43081 \cdot 44735 \cdot 1000$ művelet. Miután ez lefutott, megvannak a kifizetéseink és a kárszámaink mind az 1000 mintaelemre, azonban a kifizetések még nincsenek négyszögekbe rendezve. Erre a programot nem volt különösebben nehéz megírni, és a futási ideje sem túl hosszú, mindössze nagyjából 15 perc a DARABHATÖBB (SUMIFS) függvény használatával.

For k = 0 To 999

For j = 1 To 6

For i = 1 To 6

```
Range("sums").Cells(2 + 10 * k + i - 1, 1 + j) = WorksheetFunction.SumIfs(Range("Z").Columns(k + 1),  
Range("payments").Columns(2), Range("years").Cells(i, 1), Range("payments").Columns(4),  
Range("years").Cells(i, j))
```

Next i

Next j

Next k

A kifizetések négyszögeit a *sums* tartományba írja be, az adott mintaelemek kárszámnégyszögeivel egyező sorokba, ami javítja a munka áttekinthetőségét és ellenőrizhetőségét. Leprogramoztam a nem kumulált kárkifizetés-négyszögeket, és melléjük a háromszögeket is, hogy később könnyű legyen összehasonlítani a kitöltött háromszögeket a négyszögekkel, amit persze a program is elvégez majd. A *sums2* tartományban a kumulált kárkifizetésekkel csináltam meg ugyanezt, ezt azonban már ki lehet számolni a nem kumuláltakból, így a SUMIFS függvény újbóli használatára nincs szükség, tehát ezt a lépést már szinte egy pillanat alatt elvégzi a program. Fontos, hogy előre meglegyenek a kumulált háromszögek is, ugyanis a vizsgálni kívánt módszerekhez erre is szükség lesz.

2.2.2 A tartalékképzési módszerek leprogramozása

Most, hogy ezekkel megvagyunk, elkezdhetjük a módszereket leprogramozni. Először ezeket külön fájlba csináltam meg, hogy könnyebben átláthatóak legyenek. Most elkezdhetjük ezeket átírni úgy, hogy több mintaelemre is használhatóak legyenek a táblázat elrendezését figyelembe véve. A programoknak az összes mintaelemre való általánosításának egyik első lépése volt, hogy a fájlokban lévő változóinknak kijelölt tartományokat megfeleltessük a főfájlunk bizonyos tartományaival. Miután ezt megtettük, akkor kezdhettük el a programokat átmásolni a főfájlba, és átírni őket az új tartományokra.

A legegyszerűbb a **lánclétra módszernek** a makrója, mert ebből csak egyféle van, és nem is túl bonyolult. Ez a következő féleképpen néz ki a főfájlban:

```
Sub chainladder()
```

```
For k = 0 To 999
```

```
For j = 1 To 6
```

```
Range("sums2").Cells(9 + 10 * k, 11 + j) = 0
```

```
For i = 1 To 7 - j
```

```
Range("sums2").Cells(9 + 10 * k, 11 + j) = Range("sums2").Cells(9 + 10 * k, 11 + j) +
```

```
Range("sums2").Cells(1 + 10 * k + i, 11 + j)
```

```
Next i
```

```
Next j
```

```

For j = 1 To 6
Range("sums2").Cells(8 + 10 * k, 11 + j) = 0
For i = 1 To 6 - j
Range("sums2").Cells(8 + 10 * k, 11 + j) = Range("sums2").Cells(8 + 10 * k, 11 + j) +
Range("sums2").Cells(1 + 10 * k + i, 11 + j)
Next i
Next j

For j = 1 To 5
Range("sums2").Cells(10 + 10 * k, 11 + j) = Range("sums2").Cells(9 + 10 * k, 11 + j + 1) /
Range("sums2").Cells(8 + 10 * k, 11 + j)
Next j

For i = 2 To 6
For j = 6 - i + 2 To 6
Range("sums2").Cells(1 + 10 * k + i, 11 + j) = Range("sums2").Cells(1 + 10 * k + i, 11 + j - 1) *
Range("sums2").Cells(10 + 10 * k, 11 + j - 1)
Next j
Next i
Next k

For k = 0 To 999
For i = 1 To 6
Range("sums2").Cells(1 + 10 * k + i, 18) = (Range("sums2").Cells(1 + 10 * k + i, 7) -
Range("sums2").Cells(1 + 10 * k + i, 17)) ^ 2 / 10 ^ 12
Next i
Range("sums2").Cells(2 + 10 * k, 19) = 0
For i = 1 To 6
Range("sums2").Cells(2 + 10 * k, 19) = Range("sums2").Cells(2 + 10 * k, 19) + Range("sums2").Cells(1
+ 10 * k + i, 18)
Next i
Next k

Range("results").Cells(3, 1) = WorksheetFunction.Average(Range("sums2").Columns(19))
Range("results").Cells(3, 2) = WorksheetFunction.StDev(Range("sums2").Columns(19))
End Sub

```

k a főciklus, ez megy végig az 1000 mintaelemen. A háromszögeken belüli ciklusok mindig *i*-vel, vagy *j*-vel vannak jelölve, *i* mindig a sorokon, *j* pedig az oszlopokon megy végig, a

későbbiekben pár kivételtől eltekintve. Az utolsó két sor előtti tizenegy sor a négyzetes hiba várható értékét, és annak szórását számolja ki, az utolsó két sor pedig ezt kiírja a *results* mezőbe, ami jól látható helyen, a munkalap tetején található. Ezek a sorok minden program végén megtalálhatóak.

Ennél időigényesebb a **szeparációs módszer** átírása. Itt külön el kell helyezni a főfájlból a P háromszöget, az r és a λ együtthatókat, és a mellékegyütthatókat is, valamint a P négyszöget, aminek bal felső része nem egyezik meg teljesen a P háromszöggel. Ez lehetséges is, hiszen ha t a kifutási évek száma, akkor $2t$ darab együtthatót számolunk ki, és a háromszög elemszáma $\frac{t(t+1)}{2}$, ami jelen esetben több mint $2t$. Megfigyelhető, hogy az $r_1\lambda_1$ és az $r_t\lambda_t$ elemek mindig megegyeznek az eredeti P ugyanezek a helyeken szereplő értékeivel. Ezek után pedig a tartalékokat is ki kellett számolni valahová, ehhez azonban új tartományt nem hoztam létre, hanem a *sums* tartományba írtattam be ezeket az adatokat. A P háromszög, az együtthatók, és a P négyszög számára a *sums3* nevű tartományt hoztam létre, ami 15000*30-as méretű. Ezek szisztematikusan a tartomány 2., 12. és 22. oszlopánál és 2+15k. soránál kezdődnek, ahol k a mintaelem sorszáma mínusz egy.

A bővített szeparációs módszerhez még tartozik további egy további 15000*30-as méretű tartomány, a *sums32*. Ebben azonban 4 adattömb található. Ezek az eredeti R értékek, a q és γ paraméterek, és a hozzájuk tartozó segédparaméterek, a becsült R együtthatók, majd végül a bővített szeparációs módszerrel számolt $X_{i,j}$ értékek, amik a fentebb leírtak szerint állnak elő. Az R együtthatók transzponálással jönnek létre, amit úgy oldunk meg, hogy a $\frac{P}{r}$ négyszögeket bevisszük a *tomb* tartományba, majd ezt transzponálva átvisszük a *tomb2*-be. Ezt a becsült R értékek visszatranszponálásakor is megcsináljuk. Fontos megjegyezni, hogy a *tomb* és a *tomb2* tartomány mintaelemenként felülíródik, így végül nem foglal nagy helyet fájlunkban. Az $X_{i,j}$ értékek mellett pedig még a soronkénti összegük, és az eredeti értékektől való négyzetes eltérés is szerepel, majd az utolsó oszlopban ezek összege.

A Z tartomány felett található a *deflator* és a *deflator2* tartomány, ahol a jövőbeli évekre vonatkozó inflációs várakozásaink és a múltbeli évekre a módszer által számolt becsült inflációs ráták szerepelnek. A jövőbeli inflációs várakozások szintén a módszer egy inputja, amit meg kell adnunk. Ezt az egyszerű szeparációs módszer esetében az ANF14 cellába írhatjuk be, a bővített esetében pedig az ANJ14-be. Itt, az első sheeten csak minden jövőbeli évre azonos rátákat tudunk megadni.

-0,5071	-0,47532	-0,48865	-0,46625	-0,42627	-0,44351	-0,4492
-0,17965	-0,20422	-0,15954	-0,15206	-0,16181	-0,22716	-0,25402
0,331088	0,340237	0,409349	0,138705	0,616041	0,282681	0,442751
-0,08791	-0,22825	-0,08597	0,30927	-0,26332	0,291565	-0,17913
0,05612	0,258271	-0,13694	-0,41091	-0,07481	-0,39442	0,253487
0,5	0,5	0,5	0,5	0,5	0,5	0,5
0,5	0,5	0,5	0,5	0,5	0,5	0,5
0,5	0,5	0,5	0,5	0,5	0,5	0,5
0,5	0,5	0,5	0,5	0,5	0,5	0,5
0,5	0,5	0,5	0,5	0,5	0,5	0,5

A Z minden oszlopához a *deflator* és a *deflator2* tartományok felette lévő oszlopa tartozik. A *deflator* tartomány tartozik az egyszerűbb szeparációs módszerhez, és a *deflator2* a bővítetthez. Az oszlopok 10 sorosak, melyekben az 1. évtől a 10. évig (2t-2. évig) szerepelnek az inflációs ráták. Ezek az 1.-től az 5. sorig a módszer általi becslés eredményei, a 6.-tól a 10. sorig pedig mi adjuk meg őket inflációs várakozásaink szerint. Ez annak az oka, hogy a program által számolt $\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_t$ együtthatókból kijönnek az $\hat{i}_1, \hat{i}_2, \dots, \hat{i}_{t-1}$ inflációs becslések, az általunk megadott $i_t, i_{t+1}, \dots, i_{2t-2}$ várakozásokból pedig a program megkapja a $\lambda_{t+1}, \lambda_{t+2}, \dots, \lambda_{2t-1}$ együtthatókat, amiből számolja a jövőben várható kárfizetéseket. Lehetőségünk van évenként különböző inflációs várakozások megadására is. Ezt a második sheeten tehetjük meg és a ThisWorkbook helyen található makrókat kell hozzá futtatni. Fontos megjegyezni, hogy a program által a szeparációs módszert többféleképpen lehet futtatni, aszerint, hogy hogyan becsüljük meg előre a kifizetésszámokat. Ha az AND14 cellába 1-et írunk, akkor a program a kifizetésszámokat lánclétra módszerrel becsli, ha 0-t, akkor előre ismertnek tételezi fel és a valódi értékükkel számol, ez azonban kevésbé életszerű.

A **jéghegy módszer**hez tartozik a sums4 tartomány ami mindössze 12 oszlopból, és 10 000 sorból áll. Fontos megjegyezni, hogy adataink csak a 0 m-értékű jéghegy módszer tesztelésére alkalmasak, hiszen korábbi évek tapasztalatai nem álltak rendelkezésünkre. A háromszögekben a d együtthatók szerepelnek, alattuk a háromszög adott oszlopában lévő átlagos és minimális érték. A tartomány 8. és 10. oszlopában a adott sorbeli négyzetes hiba (a sorok a károk bekövetkezési éveit szerint vannak), a 9. és 11. oszlopban pedig ezeknek a mintaelemenkénti összege, tehát a hat vizsgált év négyzetes hibájának összege található.

8. táblázat

sums4									
jéghegy módszer d együtthatói					átlag	min			
0,069941	0,357737	0,491853	0,58279	0,916434	0	1997344	0	8131575	
0,042215	0,324815	0,616537	0,72032		57,51439		57,51439		
0,028801	0,320108	0,509767			11990,49		117339,1		
0,036099	0,29147				8617,084		266516,9		
0,023134					1956793		362270,6		
					19886,37		7385391		
0,040038	0,323532	0,539385	0,651555	0,916434	1				
0,017618	0,246392	0,455966	0,58279	0,916434	1				

A becsült kárkifizetéseket a *sums2* tartományba írja be a program, a 11. oszloptól kezdődő kumulált kárkifizetések háromszögének kiegészítéseként.

A **stagnáló infláció és a dupla lánclétra módszerek**hez a *sums5* tartomány tartozik. A felső világosabb háromszögben a dupla lánclétra módszer már fentebb említett $P_{i,j}$ értékei vannak, az alsó, sötétebb háromszögben pedig a négyszög többi része, amit lánclétra becsléssel kapunk meg. A négyszög alatti három sorban rendre a lánclétra módszer fentebb ismertett a, b és f tényezői szerepelnek. A stagnáló infláció módszeréhez az utolsó két oszlop tartozik, a 7. és a 8. oszlop. A 7. oszlopban a kifizetések átlóbeli összegei vannak, a 8.-ban pedig ezeknek az azonos átlókban lévő kifizetésszámok összegeivel vett hányadosa szerepel (λ_i), az utolsó hat darab cella már ezek becslései. Az output a nem kumulált kárkifizetések, amiket a *sums* tartományba ír be a program.

9. táblázat

sums5							
dupla lánclétra						stagnáló infláció	
134033,2	152665,2	218389,1	271743,8	2047663	656134,6	1,36E+08	134033,2
118179,5	176663,4	305649,7	298883,9	1148127	367895,5	6,26E+08	147992,7
159786,9	160446,4	235446,8	351815,8	1970338	631357,3	7,7E+08	187285,2
99938,51	162462	273230,2	331855,4	1858550	595537	1,3E+09	197397,9
119937,5	189795,4	300511,5	364990,3	2044121	654999,8	1,84E+09	278789,9
116142,3	154770,3	245054,6	297634,4	1666896	534125,1	2,18E+09	267219,3
631875,7	652237,1	759485,6	570627,7	2047663	0		306760,1
748018	842032,5	1032716	922443,5	3195790	656134,6		352151,8
1,332592	1,583344	1,214564	5,600481	0,320431			404260,2
							464079,1
							532749,5
							611581,2

Ezek mellett a program a 9. és 10 oszlopban még kiírja a fentebb említett x, y és z értékeket, a 9. oszlopba, ha $y = \min_{i \leq t} \lambda_i$ és $z = \max_{i \leq t} \lambda_i$ és a 10. oszlopba, ha $y = \lambda_1$ és $z = \lambda_t$. Az x a módszerben nem használandó, ezt csak későbbi statisztikákhoz fogjuk használni. x a $\lambda_1, \lambda_2, \dots, \lambda_t$ együtthatók mértani közepe.

$$x = \sqrt[t]{\lambda_1 \lambda_2 * \dots * \lambda_t}$$

3. Az eredmények

3.1 Az abszolút négyzetes hibák

A módszereket összehasonlítjuk a négyzetes hiba várható értéke és szórása szerint. Az 1000 darab bootstrappelt forgatókönyv szerinti háromszögeken futtatjuk le módszereinket, és az eredményt minden esetben összehasonlítjuk a forgatókönyvek szerinti négyzögekkel, amik a valós adatok. A becsült kumulált kárkifizetés-négyzögek minden sorára számolunk egy négyzetes hiba értéket, amiket néha a nem kumulált négyzögekből számolunk. A bekövetkezési évekre becsült kárösszeget összehasonlítjuk a ténylegessel olyan módon, hogy ezek különbségének négyzetét vesszük, és a nagy nagyságrend miatt ezt elosztjuk mondjuk 10^{12} -el. Az egy négyzöghöz tartozó hat darab ilyen adatot összeadjuk, így mind az 1000 mintaelemhez tartozó négyzögre kapunk egy ilyen adatot.

$$\hat{e} = \frac{\sum_{i=1}^t (Y_{i,t} - \hat{Y}_{i,t})}{10^{12}}$$

Ezek az adatok értelem szerűen minden módszernél másak. Tapasztalati várható értéküket és szórásukat kiíratjuk a programmal a results mezőbe, hogy összehasonlíthassuk a vizsgált módszereket.

10. táblázat

Módszerek	abszolút hibaérték	szórás
Jéghegy min	15048233	11974381
Jéghegy átlag	3376566	4364021
Lánclétra	3576045	4301464
Egyszerű szeparációs ismert	4043057	4722543
Egyszerű szeparációs CL	4219736	5120718
Stagnáló inf 0	5974202	5739791
Dupla lánclétra	5072299	4871182
Bővített szeparációs ismert	6731339	4271480
Bővített szeparációs CI	6495919	4069305
Stagnáló inf 1	4999390	5522388

A jéghegy módszer, ha a d együtthatók minimumát használjuk, rossz eredményt ad, mint ahogy az várható is. Ebben az esetben erősen túltartalekol a módszer. Ha a jéghegy módszernél súlyozásnak az átlagot választjuk, akkor viszont módszereink közül a legjobb eredményt kapjuk. A négyzetes hiba várható értéke csak ötöde annak, mintha a minimális együtthatót vennénk. Ezt szorosan követi a lánclétra módszer, melynek négyzetes hibája mindössze az előbbi 6 százalékával nagyobb. Az egyszerű szeparációs módszernél érdemes megjegyezni, hogy a négyzetes hiba nagymértékben függ az általunk megadott inflációs várakozásoktól. A módszert többféle inflációs rátára is lefuttatjuk, és megfigyeljük, hogy milyen rátákra hogyan alakul a módszer pontossága. Erre szolgál a második munkalap, és a hozzá tartozó program. Itt több tetszőleges inflációs rátát adhatunk meg. A fejléc alatti 6. és

7. sorban rend szerint a négyzetes hibák várható értékei és szórásai szerepelnek ismert kárszám esetén, a 8. és 9. sorban pedig rendre ugyanezek lánclétra módszerrel becsült kárszám esetén. Megfigyelhető, hogy ha a kárszám előre nem ismert, akkor többnyire valamivel rosszabb eredményeket kapunk, ami azt mutatja, hogy a lánclétra módszer becslési hibája kis mértékben növeli a szeparációs módszer hibáját.

11. táblázat

		deflators					
egyszerű		-0,2	-0,1	0	0,1	0,2	0,3
		-0,2	-0,1	0	0,1	0,2	0,3
		-0,2	-0,1	0	0,1	0,2	0,3
		-0,2	-0,1	0	0,1	0,2	0,3
		-0,2	-0,1	0	0,1	0,2	0,3
ismert X	$E(X-X')^2$	5675849	4385577	4043057	5716354	11185539	23307127
	$D(X-X')^2$	6342250	5559567	4722543	4758077	8155512	16731171
CL	$E(X-X')^2$	5863097	4606054	4268313	5880354	11159177	22860001
	$D(X-X')^2$	6479866	5736581	4934179	4883988	7957934	16100898

Több lépcsőben is közelíthetünk az optimális inflációs rátához a nagyobb pontosság érdekében. Megnézzük, melyik értékek között a legkisebb a négyzetes hiba várható értéke, és azon az intervallumon kisebb osztásközökkel újra becsülhetünk.

12. táblázat

-0,075	-0,05	-0,025	0,025	0,05	0,075
-0,075	-0,05	-0,025	0,025	0,05	0,075
-0,075	-0,05	-0,025	0,025	0,05	0,075
-0,075	-0,05	-0,025	0,025	0,05	0,075
-0,075	-0,05	-0,025	0,025	0,05	0,075
4178830	4042350	3991307	4217388	4536807	5026836
5343688	5125915	4914304	4572639	4497581	4542024
4404297	4270645	4219736	4435466	4742906	5215251
5531339	5323719	5120718	4783145	4696395	4713783

Ez alapján látható, hogy -2,5%-os inflációs ráta a legmegfelelőbb ezek közül, hiszen ott a legkisebb a várható érték. Ezzel a módszerrel kapcsolatban még az az ötlet is felmerülhet, hogy becsülhetnénk a jövőbeli inflációt a stagnáló infláció módszerében alkalmazott két eljárás egyikével, azonban az évek közti inflációs ráták nagy szórása miatt ez feltehetően ennél rosszabb eredményre vezetne. A módszer által kapott inflációs ráták nagy szórása feltételezhetően a kárnagyágok nagy szórására vezethető vissza. Azonban talán mégis érdemes foglalkozni a kérdéssel, hogy mi lenne, ha az inflációs várakozásainkat mégis valahogyan a módszer által becsült inflációs rátákból kapnánk meg, hiszen ezzel legalább azt

elérnénk, hogy a jövőbeli inflációs ráták minden forgatókönyv esetében függjenek az adott mintaelembeli becsült inflációs rátáktól, ami jó eséllyel növelné a pontosságot. A jövőbeli inflációs rátáknak választhatjuk például a becsült rátáknak a számtani vagy mértani közepét, vagy akár előre is jelezhetjük őket például AR(1) folyamattal, azonban ezeken az adatokon ez nem vezet jó eredményre. A számtani és mértani esetben majdnem olyan jó eredmények jöttek ki, mint az osztásközökkel való közelítéskor.

13. táblázat

valós kárszám, számtani közép		4577154	4885363
becsült kárszám, számtani közép		4636494	5064668
valós kárszám, mértani közép		4797355	4943407
becsült kárszám, mértani közép		4789844	5067088

Ugyanígy a bővített szeparációs módszerrel is elvégezhetjük ezt az eljárást itt, a második munkalapon. Ez a módszer rosszabbul illeszkedett az adatainkra, a négyzetes hibák várható értékei nagyobbak lettek.

14. táblázat

bővített		-0,2	-0,1	0	0,1	0,2	0,3	0,4
		-0,2	-0,1	0	0,1	0,2	0,3	0,4
		-0,2	-0,1	0	0,1	0,2	0,3	0,4
		-0,2	-0,1	0	0,1	0,2	0,3	0,4
		-0,2	-0,1	0	0,1	0,2	0,3	0,4
ismert X	$E(X-X')^2$	9825671	8991078	8149703	7343914	6630871	7273894	9425279
	$D(X-X')^2$	6762495	6342743	5866858	5341574	4792133	4557803	6876796
CL	$E(X-X')^2$	10262623	8971533	7759352	6829237	6494750	7219755	9669191
	$D(X-X')^2$	7060263	6411335	5621238	4737144	4070298	4595388	7241669

15. táblázat

	0,125	0,15	0,175	0,225	0,25	0,275
	0,125	0,15	0,175	0,225	0,25	0,275
	0,125	0,15	0,175	0,225	0,25	0,275
	0,125	0,15	0,175	0,225	0,25	0,275
	0,125	0,15	0,175	0,225	0,25	0,275
	6992194	6856691	6766834	6752932	6845082	7015371
	4781440	4581125	4406475	4199442	4208375	4321696
	6674573	6562608	6500168	6554577	6688649	6906799
	4525888	4334420	4176542	4037410	4101057	4282109

Itt az optimális inflációs ráta 20%, és a módszer 60%-kal rosszabbul illeszkedik, mint az egyszerűbb párja. Érdeemes megfigyelni, hogy itt a becsült kárszámokkal sokszor kicsit jobb az

illeszkedés, ez azt jelenti, hogy a lánclétra becslés hibája néha csökkenti ennek a módszernek a hibáját. Ha a becsült inflációk számtani vagy mértani közepeit használjuk mintaelemenként, az itt nem vezet nagyon jó eredményre, aminek oka, hogy itt a becsült inflációk még inkább ingadozóak. A hiba szórása azonban kicsit kisebb, mint az egyszerű szeparációs módszernél.

16. táblázat

valós kárszám, számtani közép	9321051	6715798
becsült kárszám, számtani közép	8963962	6515466
valós kárszám, mértani közép	8592157	6308132
becsült kárszám, mértani közép	8240558	6097751

A dupla lánclétra módszer pontossága a két szeparációs módszer között van. A stagnáló infláció módszer pontossága akkor jobb, ha $y = \lambda_1$ és $z = \lambda_t$. Ez szerepel a results mezőben a stagnáló inf 1 sorban. Ha $y = \min_{i \leq t} \lambda_i$ és $z = \max_{i \leq t} \lambda_i$, akkor a pontosság körülbelül az előbbi egy hatodával rosszabb. Ez a stagnáló inf 0 sorban látható. Elmondható, hogy minden általunk tesztelt nem szokványos módszer pontossága a kétféle szeparációs módszer pontossága között van.

3.2 A módszerek relatív hibái

A tartalékolási módszerek hibáinak becslésénél még egy dolgot mindenképpen figyelembe szokás venni. Nem mindegy ugyanis, hogy a hiba milyen irányú. Ha egy módszer alultartalékol, az sokkal nagyobb probléma, mint ha indokolatlanul sok tartalékot képez, ugyanis az alultartalékolás a biztosító csődkockázatát növeli. Az előbbi hibabecslési módszer alkalmazásával viszont ez nem derül ki egyértelműen, sőt a kapott hibaértékeket is csak egymáshoz képest tudjuk benne értelmezni. Ezért hasznos lenne egy olyan hibabecslési módszer, ami nem az abszolút hibát becsli, hanem minden mintaelemre a soronkénti becsült kárösszegek összegét osztja el a soronkénti tényleges kárösszegek összegével, és ezáltal egy relatív hibaértéket fogunk kapni. Ha $Y_{i,t}$ a tényleges kumulált kárkifizetések négyzöge és $\hat{Y}_{i,t}$ ennek a háromszögből való becslése, akkor most az $\hat{e}_r = \frac{\sum_{i=1}^t \hat{Y}_{i,t}}{\sum_{i=1}^t Y_{i,t}}$ értékeket tekintjük. A továbbiakban ezt relatív hibaértékként vagy csak röviden relatív hibaként fogjuk nevezni. Ezek az értékek az alábbi táblázatban találhatóak.

17. táblázat

Módszerek	relatív hibaérték	szórás
Jéghegy min	1,362351	0,236546
Jéghegy átlag	1,032055	0,137258
Lánclétra	1,055166	0,146318
Egyszerű szeparációs ismert	1,044513	0,140593
Egyszerű szeparációs CL	1,007642	0,131599
Stagnáló inf 0	0,994910	0,140897
Dupla lánclétra	1,143411	0,188789
Bővített szeparációs ismert	1,046034	0,155305
Bővített szeparációs CI	1,047889	0,155904
Stagnáló inf 1	0,925044	0,110073

A pirossal kiemelt két oszlop közül az elsőben ennek a fajta hibának az adatok alapján számított tapasztalati várható értéke, $E(\hat{e}_r)$, a másodikban pedig szórása, $D(\hat{e}_r)$ szerepel. Ha a várható érték 1-nél nagyobb, akkor a módszer várhatóan felültartalékol, ha kisebb, akkor várhatóan alültartalékol. Azt is jó, ha megnézzük még, hogy az 1000 esetből hányszor fordul elő az alültartalékolás. Ha túl magas tartalékot képzünk az is hátrányos. Az alábbi táblázat első oszlopában az adott módszereknél az alültartalékolás esetek száma látható, a második oszlopban pedig azon esetek száma, ahol a képzett tartalék meghaladta a szükséges másfélszeresét.

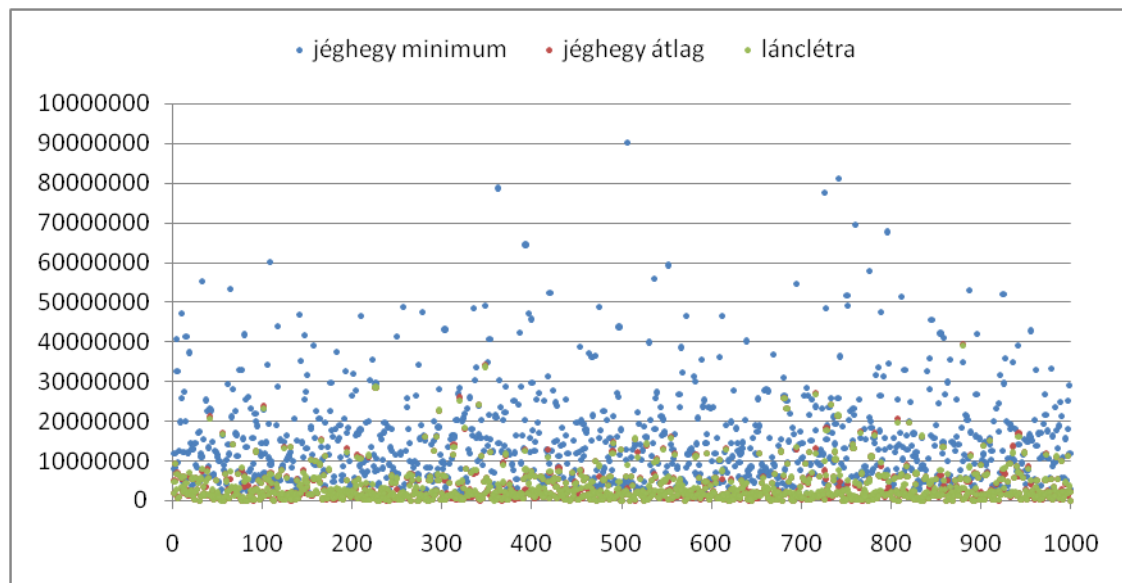
18. táblázat

Jéghegy min	48	272
Jéghegy átlag	404	0
Lánclétra	351	0
Szeparációs ismert	465	0
Szeparációs CL	479	0
Szeparációs 2 ismert	422	0
Szeparációs 2 CI	415	0
stagnáló inf 0	546	4
stagnáló inf 1	754	0
dupla lánclétra	237	40

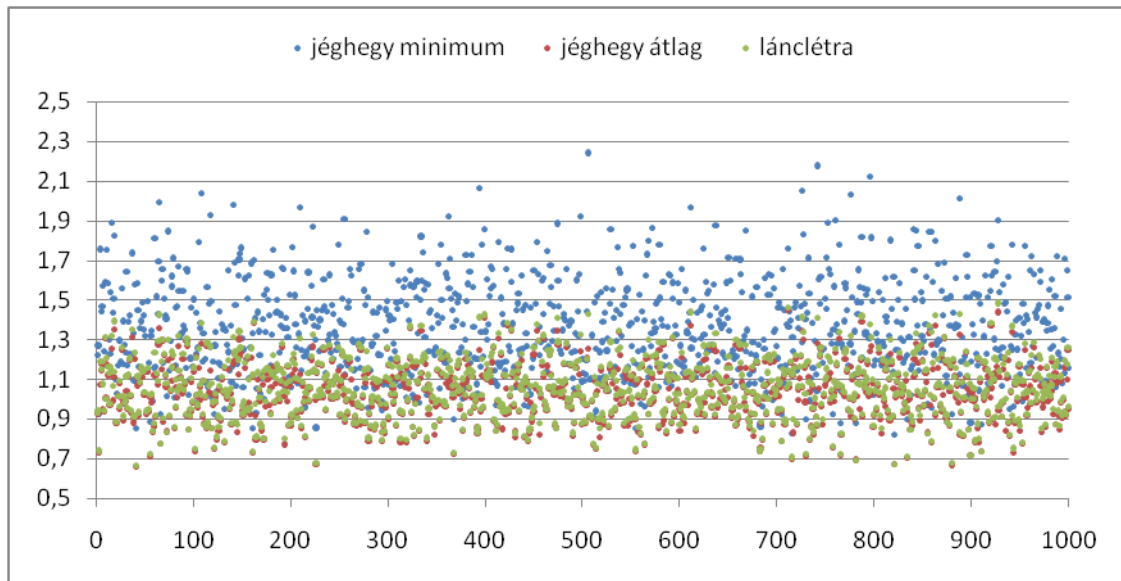
Ez alapján látható, hogy a jéghegy módszer a minimum súlyozással nagyon ritkán tartalékol alul, azonban gyakran indokolatlanul magas tartalékot képez. A stagnáló infláció 1 módszer, ahol $y = \lambda_1$ és $z = \lambda_t$, gyakran alültartalékol, a képzett tartalék várható értéke a szükségeshez képest lényegesen kevesebb, ezért nem célszerű alkalmazni. A stagnáló infláció 0, ahol $y = \min_{i \leq t} \lambda_i$ és $z = \max_{i \leq t} \lambda_i$, már nem ennyire rossz. Itt is nagy az alültartalékolás valószínűsége, azonban a képzett tartalék várható értéke nagyjából megegyezik a szükségessel (ami az 1 érték). A lánclétra módszer kevesebbszer tartalékol alul, és nem képez túl magas tartalékot sem. Átlagban 5,5%-kal tartalékol túl, ami igen jónak tűnik, és mint azt fentebb láttuk, a soronkénti abszolút négyzetes hibáinak összege sem túl magas. Emellett

még egyszerű is, ami megmagyarázza a módszer népszerűségét. Az átlaggal való súlyozással vett jéghegy módszerre is hasonlóak igazak, azonban itt már kicsivel nagyobb az alultartalékolás valószínűsége. A dupla lánclétra módszernél ritkább az alultartalékolás, mint a sima lánclétránál, azonban néha feleslegesen magas tartalékot képez. Fontos azonban megjegyezni, hogy egy módszer hibájának a relatív szórása legalább annyira fontos, mint hogy mekkora maga a hiba. Ha ugyanis rendelkezésünkre állnak egyéb káradatok az ágazaton belülről, például ha korábbi adatokkal is rendelkezünk a cégünktől, akkor egy kis hibaszórású módszer esetén egy korrekciós szorzó alkalmazásával tudunk javítani a módszer pontosságán, még ha a módszer relatív hibájának várható értéke nem is a legmegfelelőbb. A biztosító csőd kockázatát a kifizetések magas szórása növeli, a magas tartalék képzése azonban csökkenti a mérleg szerinti eredményt. Ezért ha egy tartalékképzési módszer által a kifizetések szórása alacsonyabb, akkor a kockázat elégséges fedezéséhez elég kevesebb tartalékot képezni, ezért a cég mérleg szerinti eredménye nagyobb lesz. Ennek szellemében is vizsgálhatjuk módszereinket. Az alábbi ábrák segítenek a módszerek hibáit szemléltetni.

1. ábra

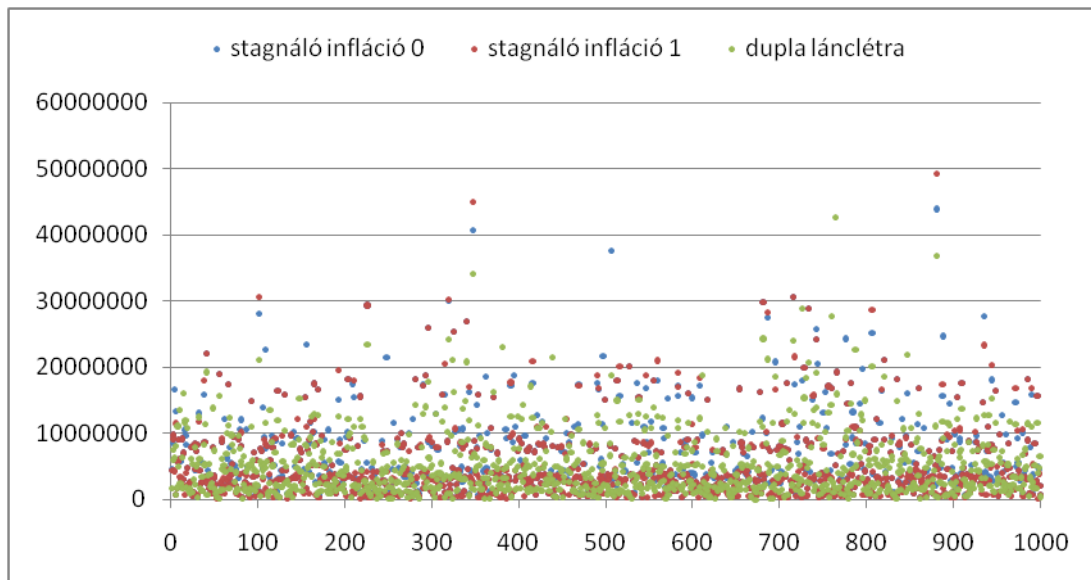


2. ábra

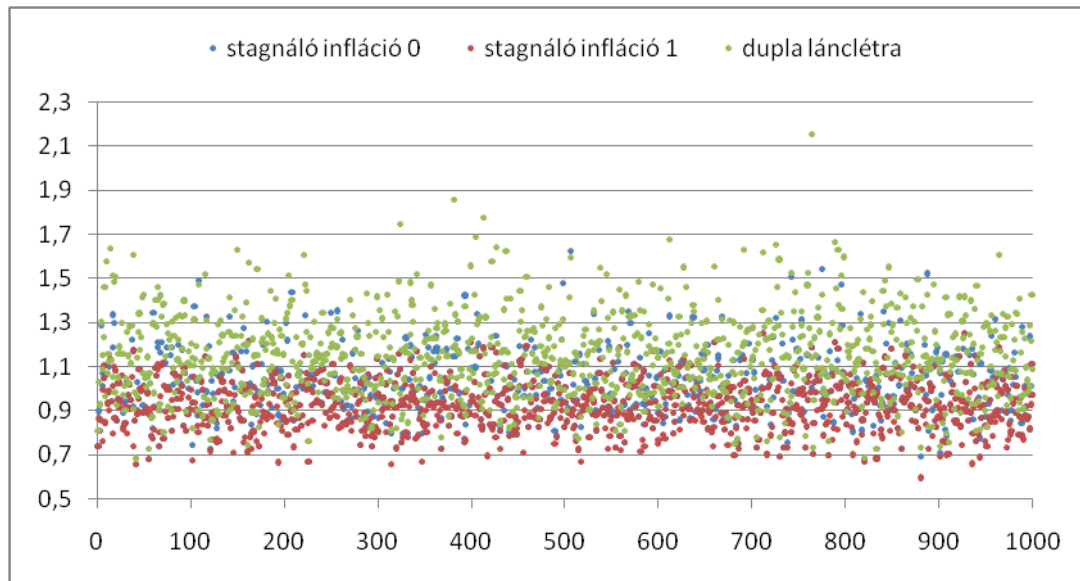


Az 1. ábra a soronkénti abszolút négyzetes hibák összegét ábrázolja minden mintaelemre. A 2. ábrán a relatív hibák ($\hat{\epsilon}_r$) láthatóak a mintaelemekre. Látható, hogy a **lánclétra és a jéghegy módszer** az átlaggal való súlyozással mintaelemenként is nagyon hasonló eredményt ad.

3. ábra



4. ábra



A 3. és 4. ábrán rendre a **stagnáló infláció** és a **dupla lánclétra módszer** fentebb bemutatott abszolút négyzetes és relatív hibái láthatóak a mintaelemekre. A 3. ábrán látható, hogy vannak mintaelemeink között bizonyos outlierok, amelyeknél mindhárom módszer rossz eredményt ad. A 4. ábrán látszik, hogy a stagnáló infláció 1 módszerhez tartozó pontok főleg az 1 alatt helyezkednek el, a stagnáló infláció 0 pontjai ennél feljebb, míg a dupla lánclétrához tartozók zömében az 1 fölött és némelyik egészen magasan, ami összhangban van a fentebb leírtakkal.

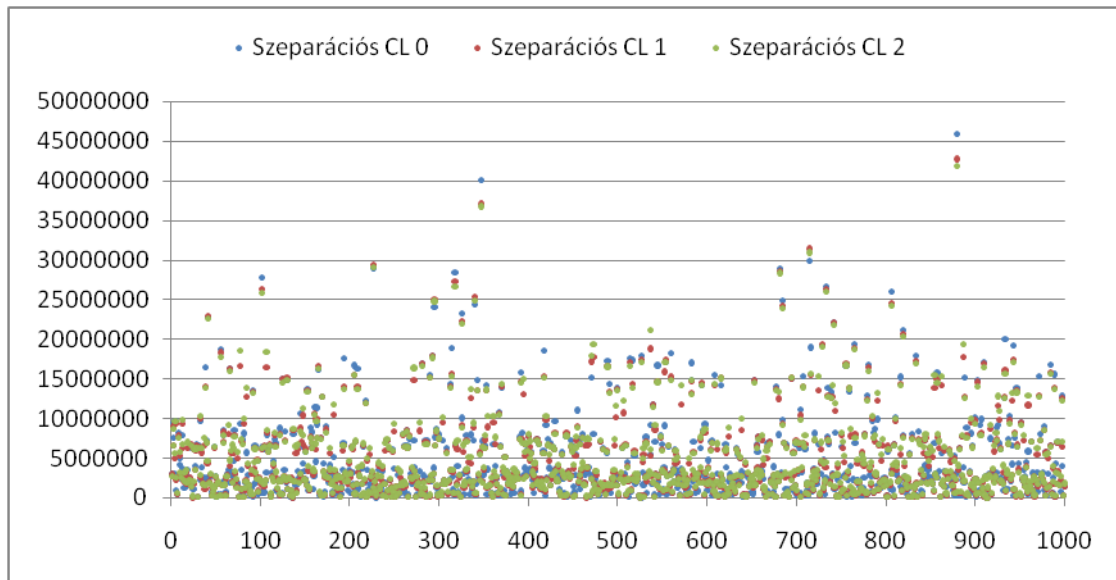
Az **egyszerű szeparációs módszer** lánclétrával becsült kifizetésszám esetén az 1000 esetből 479-ben tartalékol alul, ami logikus, hiszen a relatív hiba várható értéke csak nagyon kicsivel több 1-nél. Ismert kárszám esetén kicsit jobb az eredmény, azonban lévén, hogy ez a gyakorlatban kevésbé használható, ez kevésbé lényeges. Az alábbi ábrán a lánclétrával becsült kifizetésszám esetén szerepelnek az abszolút hibák a fentebb bemutatott három féle esetben. A jelmagyarázatban 0-val van jelölve a minden mintaelemre azonos inflációs várakozásokkal lefuttatott szeparációs módszer, 1-gyel a mintaelemenként becsült λ együtthatók számtani közepét inflációs várakozásokként használó, és 2-vel ugyanígy a mértani közép. Mint ahogyan főleg a 6. ábrán, a relatív hibák között látható, az utóbbi kettő eredményei között magas pozitív korreláció van. Az 1-es és 2-es esetben az alultartalékolások és a másfélszeresnél magasabb tartalékok képzésének száma az 1000 esetből a következő:

19. táblázat

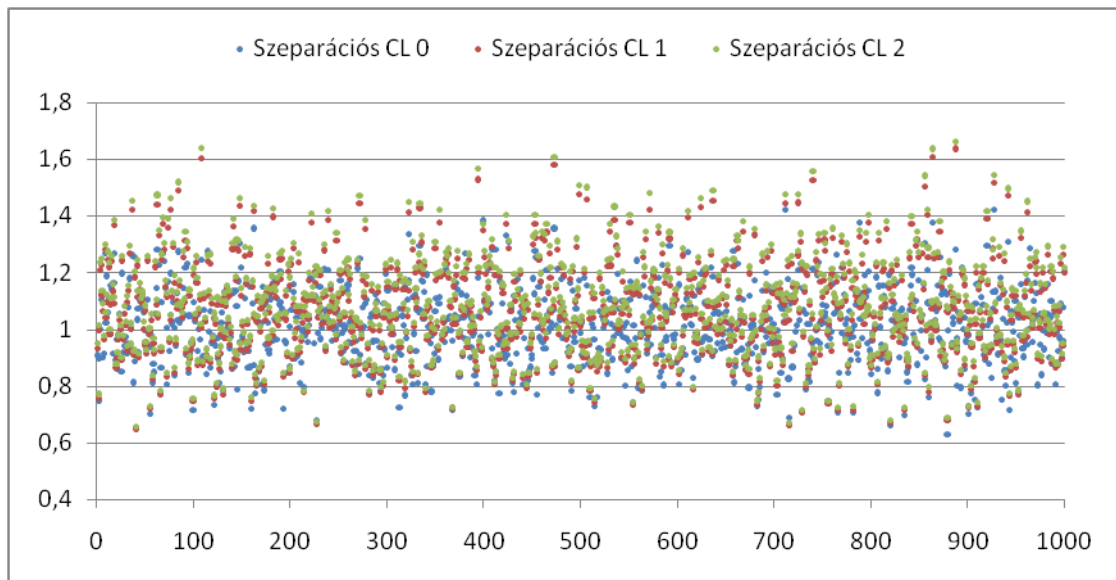
Szeparációs 1 1	308	14
Szeparációs 1 CL 1	355	8
Szeparációs 1 2	277	22
Szeparációs 1 CL 2	319	10

Annak ellenére, hogy az ismert kifizetésszámmal vett szeparációs módszernek kisebb a gyakorlati jelentősége, érdekes lehet megnézni, hogy ennek relatív hibái mennyire hasonlítanak a becsült kifizetésszámmal vett módszeréhez. A 7. ábrán az ismert kárszámmal vett relatív hibák szerepelnek. Látható, hogy a két ábra távolról nézve szinte teljesen ugyanaz, a kifizetésszámokra használt lánclétra módszer hibája pedig összességében nem rontja a szeparációs módszer pontosságát az ismert kifizetésszám-adatokhoz képest.

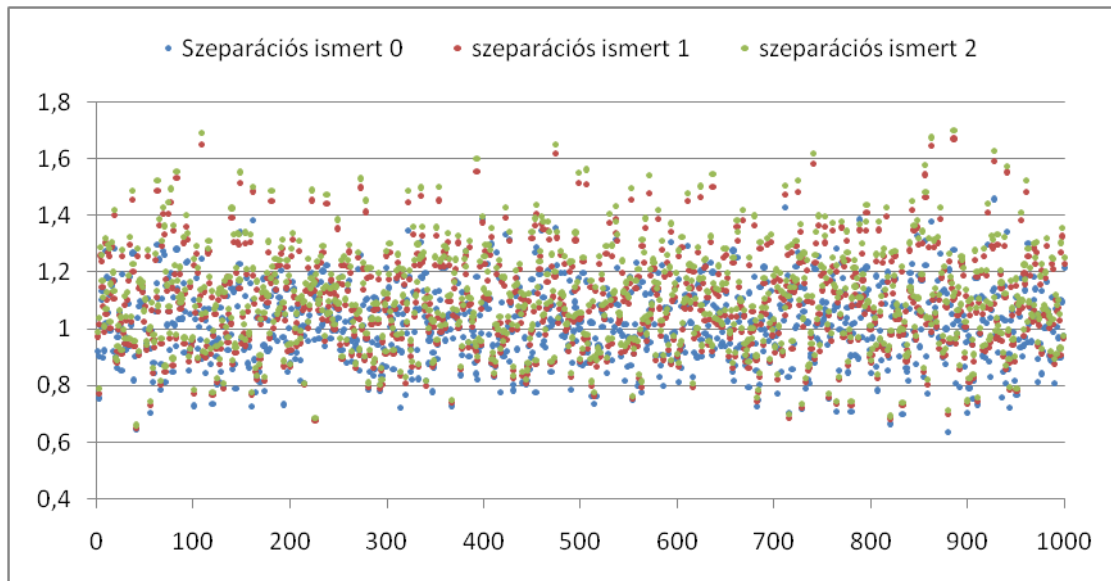
5. ábra



6. ábra

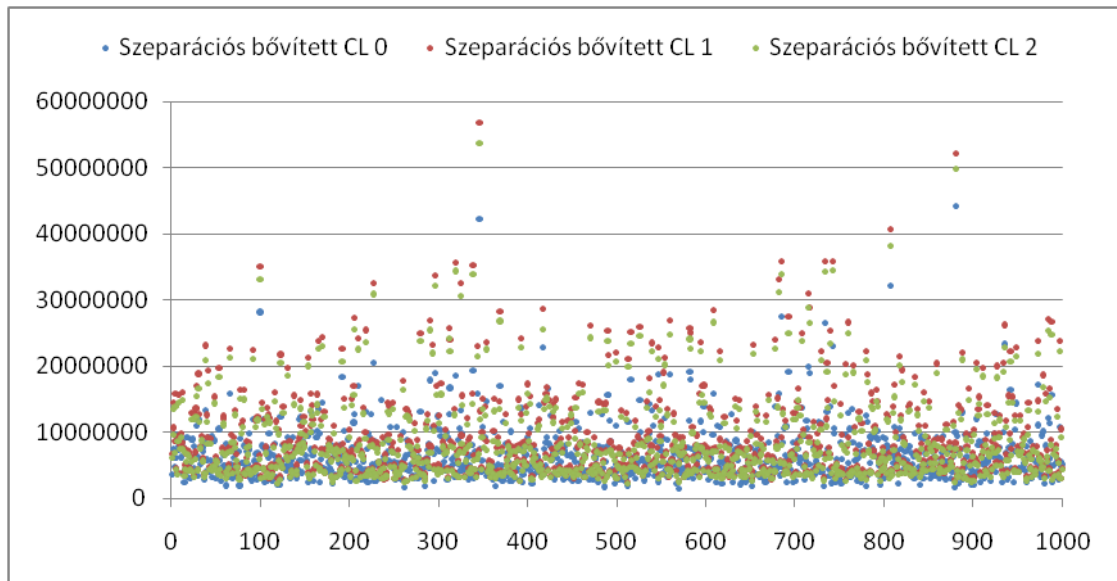


7. ábra

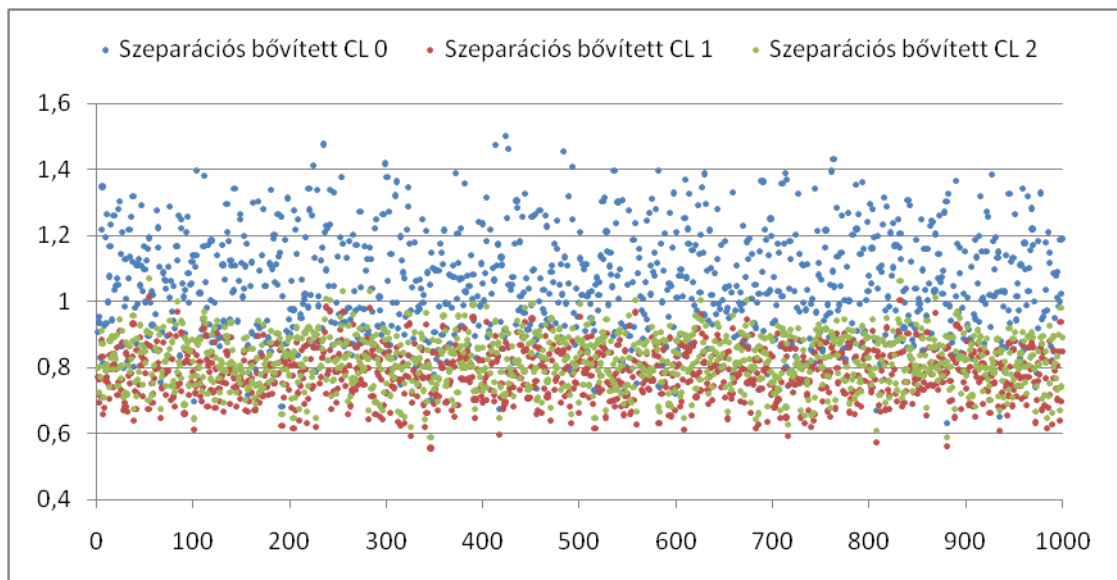


A **bővített szeparációs módszernél** a számtani és a mértani közeget használó 1-es és 2-es típus sajnos elég rossz eredményt ad a relatív hibák becslésénél is, mint ahogy az a 9. ábrán látható. Ezek a módszerek szinte minden esetben alultartalékolnak, ezért az adott adatsort adó termék esetében gyakorlati használatuk semmi esetre sem ajánlható. Látható, hogy a bővített módszer esetében a szórás kicsi, ami a hiba következetességére utal. Az alultartalékolás oka, hogy a q együttható túlságosan fejnehéz, azaz hogy az első évbeli értékei általában nagyok, későbbi években viszont nagyon kicsik lesznek. Ez a kifizetések nagyon nagy szórása miatt van így. Általánosságban elmondható, hogy a szeparációs módszer használata akkor eredményes, ha valóban megfigyelhető valamilyen inflációs hatás a kifizetésekben, és mint minden módszernél, ha kisebb a kifizetésenkénti összegek szórása. Ha nagy a kifizetések szórása, akkor általában a módszer hibájának szórása is nagy lesz. A mi adatsorunkban látható, hogy nagy az összegek szórása, mint ahogy a belőlük becsült inflációs együtthatóknak is, és ezért az abszolút hiba nagyobb lesz, és jelen esetben a relatív hiba is nagy lesz.

8. ábra



9. ábra



A relatív hiba értékei számokban a következők rendszerint várható értékkel és szórással:

20. táblázat

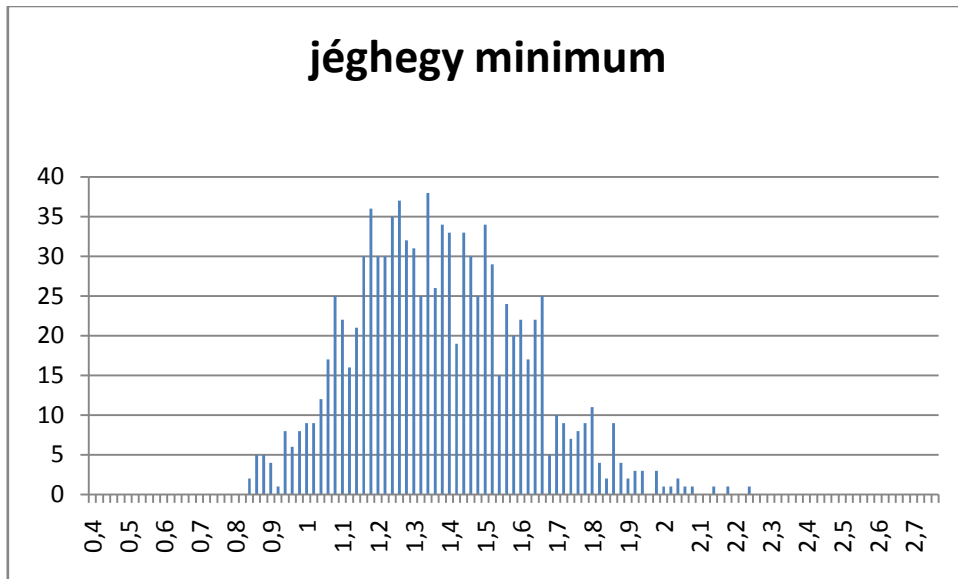
	relatív hibaérték	szórás
Egyszerű, ismert, számtani	1,088069	0,166496
Egyszerű, becsült, számtani	1,061571	0,159698
Egyszerű, ismert, mértani	1,109229	0,171433
Egyszerű, becsült, mértani	1,080465	0,163959
Bővített, ismert, számtani	0,782920	0,076073
Bővített, becsült, számtani	0,779885	0,076747
Bővített, ismert, mértani	0,821238	0,078303
Bővített, becsült, mértani	0,818031	0,078830

Látható a bővített szeparációs módszer erős alultartalékolása, amennyiben a jövőbeli inflációt a múltbeli számolt infláció számtani vagy mértani átlagának becsüljük.

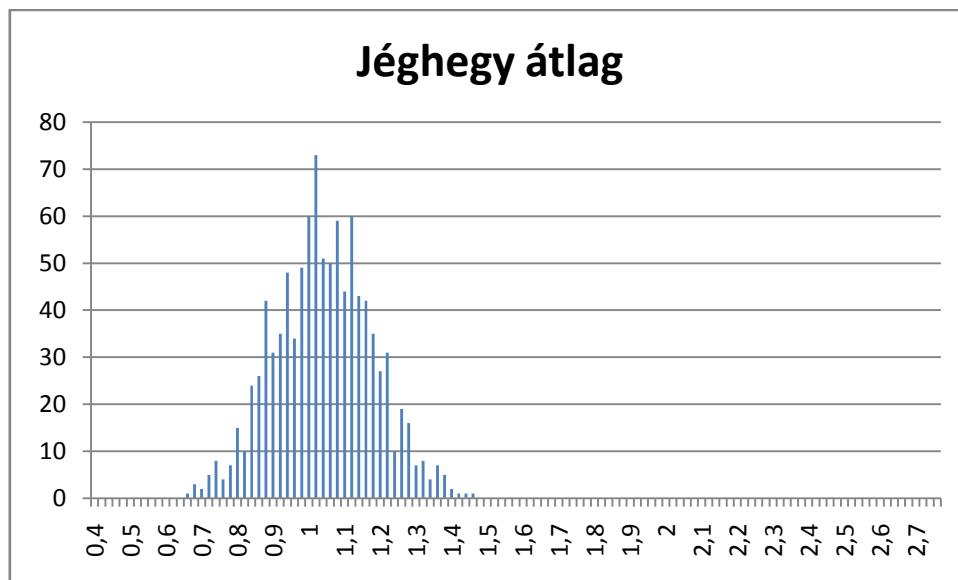
Pontosabb képet kaphatunk azonban, ha a különböző módszerek általi hibák nagyságának eloszlásait több mérőszámmal is elemezzük. Az alábbi táblázatban ezek szerepelnek. Nem mindegy ugyanis, hogy az esetek mekkora részében esnek a hibák az átlag alá vagy fölé, valamint a hibaadatok mediánja is fontos, ami jelen esetben, ha nagyság szerint sorba rendezzük a relatív hibákat, a két középső átlaga lesz. Az alsó és felső kvartilis rendszerint megmutatja, hogy mi az az érték, ami alá illetve fölé az adatok 25%-a esik. A ferdeség megmutatja, hogy az adott módszer hibaeloszlása jobbra vagy balra nyúlik el egy szimmetrikus eloszláshoz képest. Ha jobbra nyúlik el, vagyis az eloszlás jobb oldali farka a jelentősebb, akkor az érték pozitív, ha balra, akkor negatív, ha pedig az eloszlás szimmetrikus, akkor a ferdesége zérus. A normális eloszlás csúcossága 0. Szemléletesen, ha egy eloszlás sűrűségfüggvénye laposabb a normális eloszlásénál, akkor csúcossága negatív, ha csúcsosabb, akkor pozitív. A ferdeség és a csúcosság diszkrét eloszlások esetén is értelmezhető, így használhatjuk őket adataink elemzéséhez.

A jéghegy módszer a minimum súlyozással az esetek több, mint egy negyedében több, mint másfélszeresét képi a szükséges tartaléknak, sűrűségfüggvénye jobbra nyúlik el, előfordulnak kirívóan magas értékek. A jéghegy módszer az átlag súlyozással és a lánclétra módszer legtöbb tulajdonságában hasonló, jó eredményt ad, közepes szórással. Egyedül a csúcosságukban van említésre méltó különbség. Ez alapján a jéghegy módszernél van egy olyan kicsi relatív hibaérték intervallum, amibe aránylag sok mintaelem esik. Ez azonban esetünkben nem befolyásolja nagyban a módszer használhatóságát.

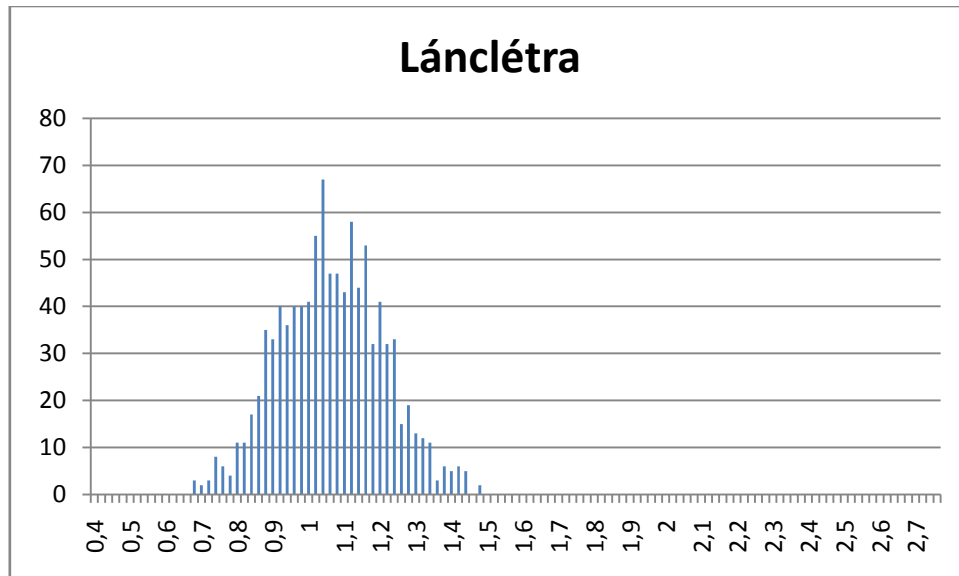
10. ábra



11. ábra

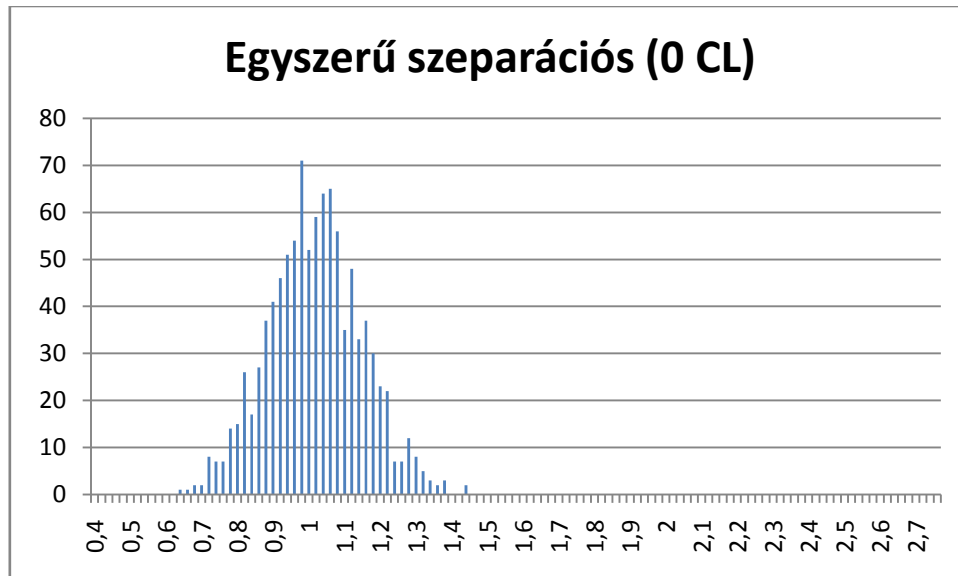


12. ábra

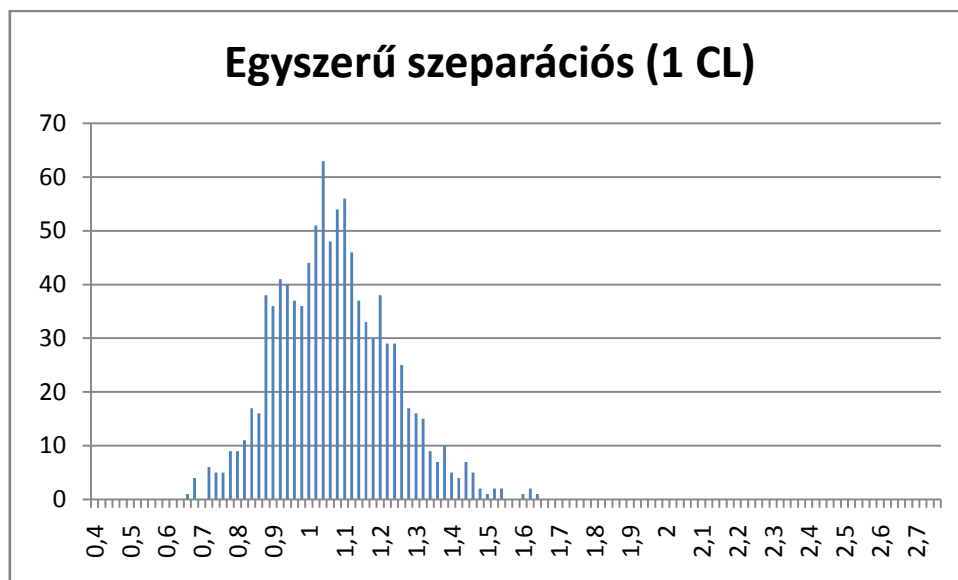


Az egyszerű szeparációs módszer abban az esetben a legjobb a három közül, amikor az inflációs paraméter előrejelzését magunk optimalizáltuk úgy, hogy a módszer a lehető legjobb illeszkedést mutassa, azonban ne feledjük, hogy erre a valóságban ebben a formában nincs lehetőség. A valóságban valamilyen előrejelzés vagy várakozás alapján választunk inflációs paramétert. Ennek ellenére van az optimalizációs módszernek létjogosultsága, hiszen ha vannak hasonló adataink előbbi évekből, lehetőleg olyan korból, amikor az infláció a valóságban is hasonlóan alakult a mostanihoz, akkor az azokon legoptimálisabb inflációs paramétert választva valószínűleg jó eredményt kapunk a mostani adatainkkal is. Ehhez képest az egyszerű szeparációs módszer másik két módosítása nem kér inflációs paramétereket bemenő adatként, ugyanis azokat a számolt paraméterekből állítja elő azok számtani illetve mértani közepeként. Ezeknél a szórás kicsit nagyobb, és erősen jobbra ferdek a súlyfüggvényeik, tehát előfordulnak nagyobb értékek is, mint az előző módszernél.

13. ábra

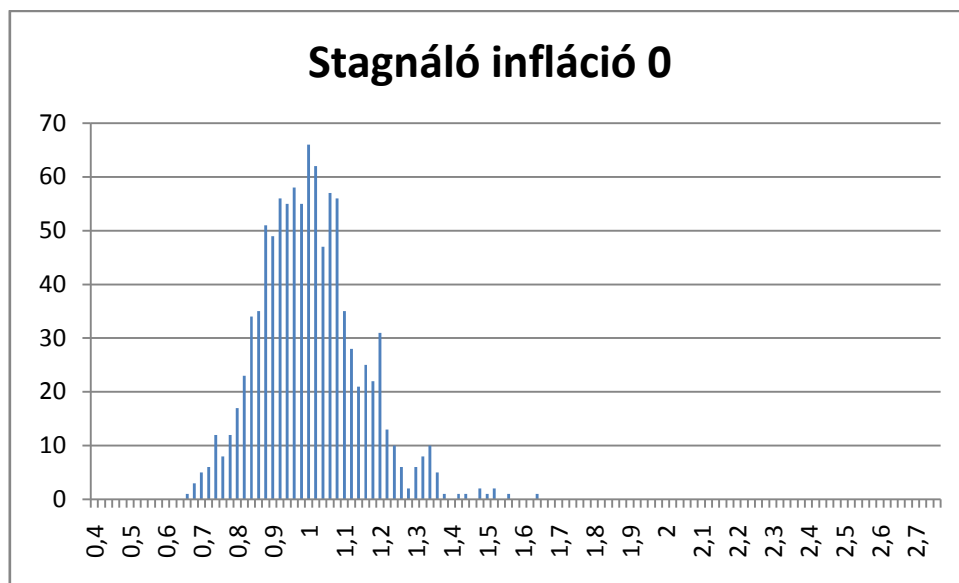


14. ábra

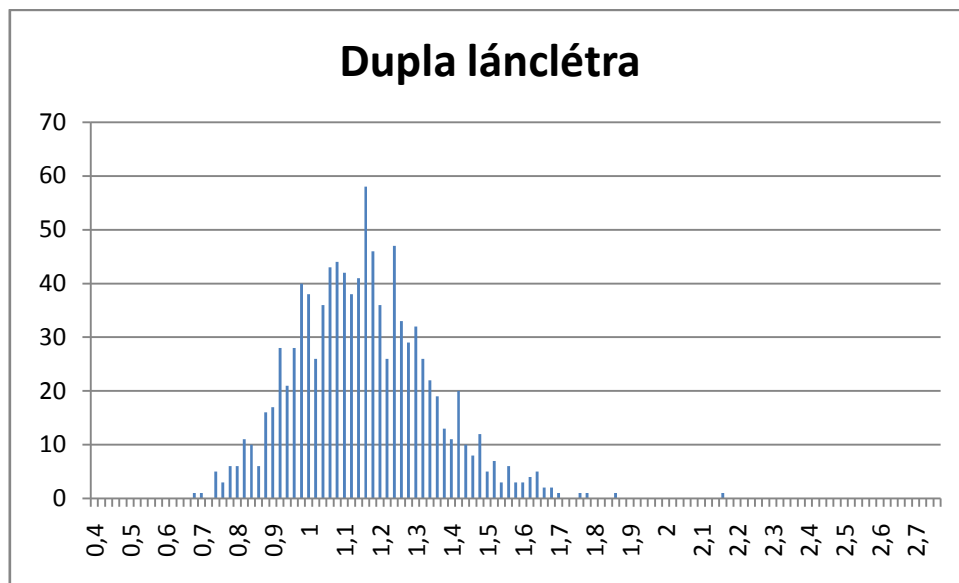


A stagnáló infláció 0 módszer várható értéke megfelelő, de lévén, hogy erősen jobbra ferde, gyakran képez túl alacsony tartalékokat, ezért csak korrekciós szorzóval célszerű alkalmazni. A stagnáló infláció 1 alultartalékol, viszont korábbi gondolatmenetünk eredményeként fontos az, hogy a szórása viszont alacsony. Ezért ha rendelkezésünkre állnak a vizsgálandóhoz hasonló ismert adatok, azokból meghatározhatunk egy megfelelő korrekciós szorzót, és így már akár használhatjuk is a módszert. Azonban ha ilyen adatokkal nem rendelkezünk, semmiképpen sem ajánlott a használata. Az általunk dupla lánclétraként emlegetett módszer a magas szórása miatt kevésbé hasznos, és ráadásul előfordulnak nagyon nagy hibaértékek is, ami logikus, hiszen a két lánclétra módszer együttes hibája jelenik meg benne.

16. ábra



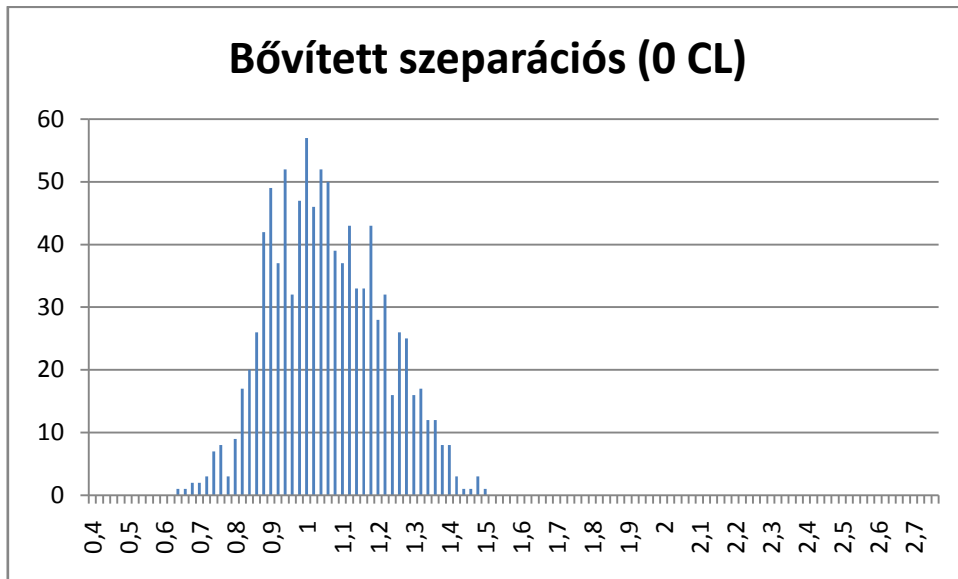
17. ábra



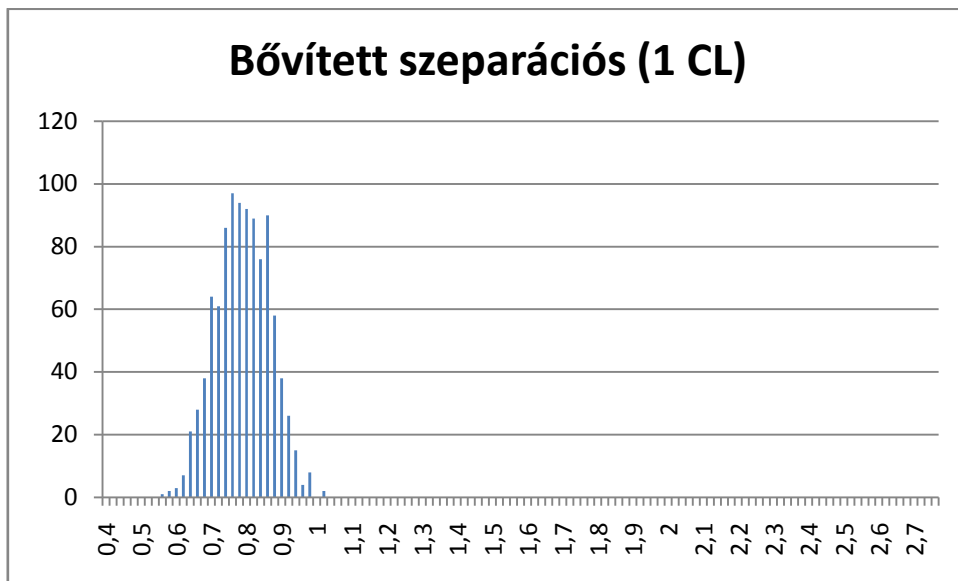
A bővített szeparációs módszer első látásra semmiképp nem tűnik jó választásnak, ami részben igaznak is mondható. Fontos megjegyezni, hogy amikor az inflációt optimalizáljuk (Szeparációs 2 CL 0), 20% jön ki az optimális inflációs paraméterre az évek között, ez azonban annyira irreális, hogy egyéb előrejelzés alapján valószínűleg nem jutna eszünkbe a paramétert hasonlóan választani, már pedig a módszer hibája érzékeny erre a paraméterre. A táblázatban szereplő értékeknél tehát egy reális inflációs várakozással dolgozva sokkal rosszabbat kapnánk. [] A másik kétféle bővített szeparációs módszer viszont látszólag rossznak tűnik, azonban fontos kiemelni, hogy az összes módszer közül ezeknél a legkisebb a szórás, és ez az állítás még akkor is igaz marad, ha egy 1,3 értékű korrekciós tényezővel

állítjuk be a tartalékot, így a hiba javítható, persze csak akkor, ha valahonnan tudjuk a megfelelő szorzó értékét.

18. ábra



19. ábra



21. táblázat

	Jéghegy minimum	Jéghegy átlag	Lánclétra	Szeperációs 1 0	Szeperációs 1 CL 0	Szeperációs 1 1	Szeperációs 1 CL 1	Szeperációs 1 2	Szeperációs 1 CL 2
Alsó kvartilis	1,1881	0,9356	0,9485	0,9494	0,9196	0,9659	0,9479	0,9856	0,9605
Medián	1,3458	1,0302	1,0514	1,0426	1,0055	1,0775	1,0523	1,0986	1,0699
Felső kvartilis	1,5146	1,1236	1,1554	1,1358	1,0939	1,1940	1,1663	1,2166	1,1887
Minimum	0,8202	0,6583	0,6662	0,6508	0,6290	0,6486	0,6467	0,6570	0,6551
Átlag	1,3624	1,0321	1,0552	1,0445	1,0076	1,0881	1,0616	1,1092	1,0805
Maximum	2,2353	1,4487	1,4787	1,5157	1,4216	1,6667	1,6327	1,7004	1,6632
Szórás	0,2365	0,1373	0,1463	0,1406	0,1316	0,1665	0,1597	0,1714	0,1640
Ferdeség	0,3702	0,0539	0,0710	0,1235	0,0843	0,3683	0,3434	0,3793	0,3526
Csúcsosság	-0,0022	-0,1508	-0,2297	-0,0258	-0,0337	0,1989	0,2001	0,2127	0,2130
	stagnáló infláció 0	stagnáló infláció 1	Dupla CL	Szeperációs 2 0	Szeperációs 2 CL 0	Szeperációs 2 1	Szeperációs 2 CL 1	Szeperációs 2 2	Szeperációs 2 CL 2
Alsó kvartilis	0,8985	0,8548	1,0111	0,9270	0,9266	0,7294	0,7254	0,7687	0,7628
Medián	0,9875	0,9223	1,1372	1,0321	1,0351	0,7832	0,7798	0,8193	0,8163
Felső kvartilis	1,0731	0,9993	1,2584	1,1566	1,1605	0,8413	0,8377	0,8839	0,8805
Minimum	0,6578	0,5905	0,6774	0,6307	0,6319	0,5514	0,5522	0,5868	0,5875
Átlag	0,9949	0,9250	1,1434	1,0460	1,0479	0,7829	0,7799	0,8212	0,8180
Maximum	1,6256	1,2500	2,1544	1,4907	1,5000	0,9975	1,0105	1,0549	1,0692
Szórás	0,1409	0,1101	0,1888	0,1553	0,1559	0,0761	0,0767	0,0783	0,0788
Ferdeség	0,5779	0,0740	0,4999	0,2350	0,2268	-0,0535	-0,0158	-0,0990	-0,0572
Csúcsosság	0,9035	-0,1562	0,8399	-0,4236	-0,4250	-0,3733	-0,3398	-0,3732	-0,3242

3.3 Korrelációk

A következő táblázatok a szeparációs módszerek hibáinak korrelációit tartalmazzák. A fejlécekben szeparációs 1 az egyszerű, a Szeparációs 2 a bővített szeparációs módszert jelenti. Ha van az adott fejlécben CL jelölés, akkor az adott sor vagy oszlop a lánclétrával becsült kifizetésszámra vonatkozik, ha nincsen, akkor ismert kifizetésszámra. Az utolsó karakterek (0, 1 vagy 2) a fentebbi grafikonok jelöléseivel ekvivalensek. A táblázatok celláiban a piros színnel szereplő érték a módszerek abszolút hibái ($\hat{\epsilon}$) közti korrelációs együttható, a kézzel szereplő érték pedig ugyanez a relatív hibákkal ($\hat{\epsilon}_r$). Megfigyelhető, hogy a módszerek becsült és ismert kifizetésszámmal vett esetei között az eredményekben nincs nagy különbség, hiszen a halvány barna háttérű négyzetekben minden korreláció nagyon közel van 1-hez. Az utolsó táblázatból látható, hogy az egyszerű és a bővített szeparációs módszerrel becsült adatok jelentősen eltérnek egymástól, mint már tudjuk, az egyszerű módszer javára.

22. táblázat

	Szeparációs 1 0	Szeparációs 1 CL 0	Szeparációs 1 1	Szeparációs 1 CL 1	Szeparációs 1 2	Szeparációs 1 CL 2
Szeparációs 1 0	1,0000 1,0000	0,9942 0,9982	0,9309 0,8750	0,9573 0,8780	0,9003 0,8711	0,9400 0,8750
Szeparációs 1 CL 0	0,9942 0,9982	1,0000 1,0000	0,9132 0,8753	0,9496 0,8817	0,8772 0,8712	0,9279 0,8787
Szeparációs 1 1	0,9309 0,8750	0,9132 0,8753	1,0000 1,0000	0,9920 0,9978	0,9960 0,9996	0,9965 0,9974
Szeparációs 1 CL 1	0,9573 0,8780	0,9496 0,8817	0,9920 0,9978	1,0000 1,0000	0,9787 0,9972	0,9974 0,9996
Szeparációs 1 2	0,9003 0,8711	0,8772 0,8712	0,9960 0,9996	0,9787 0,9972	1,0000 1,0000	0,9895 0,9975
Szeparációs 1 CL 2	0,9400 0,8750	0,9279 0,8787	0,9965 0,9974	0,9974 0,9996	0,9895 0,9975	1,0000 1,0000

23. táblázat

	Szeparációs 2 0		Szeparációs 2 Cl 0		Szeparációs 2 1		Szeparációs 2 Cl 1		Szeparációs 2 2		Szeparációs 2 Cl 2	
Szeparációs 2 0	1,0000	1,0000	0,9946	0,9982	0,8277	0,6086	0,8311	0,6121	0,8352	0,5773	0,8389	0,5844
Szeparációs 2 Cl 0	0,9946	0,9982	1,0000	1,0000	0,7821	0,6151	0,7866	0,6222	0,7913	0,5824	0,7964	0,5934
Szeparációs 2 1	0,8277	0,6086	0,7821	0,6151	1,0000	1,0000	0,9998	0,9979	0,9989	0,9937	0,9984	0,9937
Szeparációs 2 Cl 1	0,8311	0,6121	0,7866	0,6222	0,9998	0,9979	1,0000	1,0000	0,9990	0,9893	0,9989	0,9940
Szeparációs 2 2	0,8352	0,5773	0,7913	0,5824	0,9989	0,9937	0,9990	0,9893	1,0000	1,0000	0,9998	0,9975
Szeparációs 2 Cl 2	0,8389	0,5844	0,7964	0,5934	0,9984	0,9937	0,9989	0,9940	0,9998	0,9975	1,0000	1,0000

24. táblázat

	Szeparációs 2 0		Szeparációs 2 Cl 0		Szeparációs 2 1		Szeparációs 2 Cl 1		Szeparációs 2 2		Szeparációs 2 Cl 2	
Szeparációs 1 0	0,8255	0,6086	0,7863	0,5884	0,9329	0,4529	0,9334	0,4325	0,9341	0,4759	0,9344	0,4570
Szeparációs 1 Cl 0	0,8180	0,6093	0,7771	0,5921	0,9499	0,4750	0,9502	0,4575	0,9508	0,4988	0,9509	0,4830
Szeparációs 1 1	0,7001	0,2855	0,6566	0,2619	0,8541	0,2762	0,8538	0,2478	0,8486	0,3315	0,8478	0,3032
Szeparációs 1 Cl 1	0,7315	0,2938	0,6878	0,2732	0,8952	0,2900	0,8948	0,2652	0,8912	0,3450	0,8904	0,3207
Szeparációs 1 2	0,6692	0,2819	0,6262	0,2581	0,8200	0,2699	0,8195	0,2414	0,8127	0,3253	0,8116	0,2969
Szeparációs 1 Cl 2	0,7119	0,2915	0,6687	0,2710	0,8740	0,2846	0,8735	0,2598	0,8686	0,3395	0,8677	0,3153

4. Összefoglalás

A való életben nem dönthető el egyértelműen, hogy melyik tartalékképzési módszer a legpontosabb, hiszen ez függ az adott káradatok jellegétől. A mi káadataink alapján azonban elmondható, hogy a hagyományos módszerek jól illeszkednek, továbbá az egyszerű szeparációs módszer még ha a bekövetkezési évekre lebontva nem is ad olyan jó eredményt, az egész kifutási háromszögre nézve azonban ezek a hibák valamelyest kiegyenlítik egymást, ezért összességében nem tekinthető rossznak. A bővített szeparációs módszer azonban rosszul illeszkedik adatainkra, ami alapján elvethetjük azt a feltevést, miszerint a bekövetkezési évektől függő inflációs hatás megfigyelhető az adatainkon. A stagnáló infláció módszer túl gyakran tartalékol alul, a dupla lánclétra módszer viszont kielégítő eredményeket ad. Fontos azonban megjegyezni, hogy bármennyire is jók, vagy rosszak ezek a módszerek, a valóságban a biztosító társaságok nem ennyire egyszerűen számolnak, hiszen évközben is folyamatos a tartalékképzés, és az év végére becsült kifizetendő összeg nem csak az eddigi évek tapasztalataitól, hanem az ideig eddigi tapasztalatoktól és az évből hátralévő időtől egyaránt függenek. Ezzel együtt azonban fontos a megfelelő tartalékképzési módszert használata, és ennek érdekében ezeknek különböző már meglévő adatokon való tesztelése.

Köszönetnyilvánítás

Ezúton szeretném megköszönni Arató Miklós tanár úrnak a dolgozat elkészítéséhez nyújtott értékes segítségét, és ötleteit, amelyek nagy mértékben segítettek munkámat.

Irodalomjegyzék

- [1] Arató Miklós: Nem-életbiztosítási matematika, Eötvös kiadó, 2001

- [2] G.C. Taylor: Separation of inflation and effects from the distribution of non-life insurance claim delays, ASTIN Bulletin / Volume 9 / Issue 1-2 / January 1977, pp 219-230

- [3] Susanna Björkwall, Ola Hössjer and Esbjörn Ohlsson: Bootstrapping the separation method in claims reserving 2010, ASTIN Bulletin / Volume 40 / Issue 02 / November 2010, pp 845-869

- [4] Bártfai Barnabás: Makróhasználat Excelben, BBS-INFO Kiadó, 2010.

- [5] P. D. England and R. J. Verrall: Stochastic claims reserving in general insurance [Presented to the Institute of Actuaries, 28 January 2002], British Actuarial Journal / Volume 8 / Issue 03 / August 2002, pp 443-518

- [6] Maria Dolores Martinez Miranda, Jens Perch Nielsen and Richard Verrall: Double Chain Ladder, ASTIN Bulletin / Volume 42 / Issue 01 / May 2012, pp 59-76