

LATVIJAS UNIVERSITĀTE
FIZIKAS UN MATEMĀTIKAS FAKULTĀTE
MATEMĀTIKAS NODAĻA

**NEDZĪVĪBAS APDROŠINĀŠANAS IBNR REZERVJU
APRĒĶINĀŠANA AR VISPĀRINĀTĀ LINEĀRĀ MODEĻA
UN BUTSTRAPA METODĒM**

MAĢISTRA DARBS

Autors: **Natalja Vidinejeva**

Stud. apl. MaSt020018

Darba vadītājs: doc. Dr. math. Jānis Valeinis

RĪGA 2009

Anotācija

Darbs tiek veltīts nedzīvības apdrošināšanas IBNR atlīdzību rezervju prognozes un prognozes standartklūdas novērtēšanai, kā arī pilnā prognozes sadalījuma noteikšanai, izmantojot vispārināto lineāro modeli un butstrapa datu pārkārtošanas metodi. Darbā aplūkota klasiskās chain-ladder metodes saistība ar vispārināto lineāro modeli, kas dod iespēju iegūt analītisko rezervju prognozes standartklūdas novērtējumu. Virs-izklaidēta Puasona un Gamma modeļu analītiski aprēķinātā rezervju prognozes standartklūda tiek salīdzināta ar butstrapa standartklūdu un arī tiek iegūts pilns prognozes sadalījums. Šīs metodes tiek pielietotas konkrētiem datiem par apdrošināšanas atlīdzību attīstību kādā laika posmā un tiek salīdzināti iegūtie rezultāti.

Atslēgas vārdi: nedzīvības apdrošināšana, IBNR atlīdzību rezerves, Chain-ladder, butstraps, vispārinātais lineārais modelis

Abstract

This thesis is devoted to the IBNR claims reserves projection and projection error estimation and also estimation of a full predictive distribution, using generalised linear model and bootstrap resampling procedure. Standard chain-ladder model is considered within the generalised linear model framework, which provides analytic standard errors of prediction of reserve estimates. Over-dispersed Poisson model and Gamma model analytic prediction errors are compared with bootstrap standard errors and also full predictive distribution is obtained. These methods are implemented using data about claims development over some period of time.

Keywords: general insurance, IBNR claims reserves, Chain-ladder, bootstrap, generalised linear model

Saturs

Ievads	3
1. Vispārinātais lineārais modelis	5
1.1. Vispārinātā lineārā modeļa pamati	5
1.1.1. Vispārinātā lineārā modeļa struktūra	5
1.1.2. Modeļa parametru novērtējums ar vislielākās ticamības funkciju . . .	9
1.1.3. Vislielākās kvazi-ticamības funkcija	10
1.1.4. Modeļa pielāgošanas kvalitātes mērs un modeļa atlikumi	13
1.2. Vispārinātā lineārā modeļa izmantošana IBNR rezervju noteikšanai	15
1.2.1. Atlīdzību rezervēšanas problēmas nostādne un galvenie jēdzieni . . .	15
1.2.2. Zaudējumu atlīdzību rezervju prognozes kļūda	17
1.2.3. Chain-ladder metode	18
1.2.4. Virs-izkliedēts Puasona modelis	19
1.2.5. Chain-ladder modeļa Normālā aproksimācija	27
1.2.6. Maka modelis	32
1.2.7. Log-Normālais modelis	33
1.2.8. Gamma modelis	35
2. Butstrapa metode	36
2.1. Butstrapa metodes jēdziens	36
2.2. Rezidiju butstraps	36
2.3. Rezidiju butstrapa metodes izmantošana IBNR rezervju noteikšanai	37
3. Empīriskie rezultāti	39
3.1. Chain-ladder metode	39
3.2. Vispārinātais lineārais modelis	40
3.2.1. Virs-izkliedēts Puasona modelis	40
3.2.2. Chain-ladder modeļa Normālā aproksimācija	41

3.2.3.	Maka modelis	45
3.2.4.	Log-Normālais modelis	47
3.2.5.	Gamma modelis	50
3.3.	IBNR atlīdzību rezervju aprēķināšana ar butstrapa metodi	50
3.4.	Modeļu pārbaude	53
Secinājumi		58
Izmantotā literatūra un avoti		59
1.	Atlīdzību pieaugumu attīstības trijstūris	61
2.	Rezervju aprēķins ar chain-ladder metodi	62
3.	Attīstības faktoru λ kovariācijas matrica	63
A	Izveidoto programmu kods	64
A1.	Virs-izkļiedēts Puasona modelis	64
A2.	Butstraps Puasona modelim	65
A3.	Gamma vispārinātais lineārais modelis un butstraps	68
A4.	Chain-ladder modeļa Normālā aproksimācija	71
A5.	Log-Normālais modelis	74
A6.	Maka modelis	77

Ievads

IBNR ir saīsinājums no angļu valodas ”*incurred but not reported*”. Šis termins attiecas uz apdrošināšanas gadījumiem, kuri jau ir iestājušies, bet vēl nav zināmi apdrošinātājam un par kuriem apdrošinātājam pastāv saistības uz rezervju izveidošanas datumu. IBNR rezerves spēlē svarīgu lomu apdrošināšanas kompānijas finanšu uzskaitē, īpaši tad, ja būtisku parakstīto prēmiju daļu sastāda nelaimes gadījumu apdrošināšana. Neprecīzu IBNR rezervju noteikšana noved pie neoptimāliem apdrošināšanas kompānijas vadības lēmumiem. Atbilstošo IBNR rezervju nodrošināšana ir ne tikai piesardzīga finanšu uzskaites politika, bet tas arī tiek prasīts ar likumu.

Pašlaik visizplatītākā metode nedzīvības apdrošināšanas IBNR rezervju aprēķināšanas praksē ir “chain-ladder” metode, kura ir samērā vienkārši pielietojama, pat bez īpašas programmatūras (to var vienkārši realizēt ar programmu Excel). Bet šīs metodes trūkums ir, ka tā dod tikai punktveida rezervju novērtējumu. Taču bieži arvien lielāko interesi izraisa rezervju novērtējuma augšējā vai apakšējā robeža ar noteiktu ticamības līmeni, kas nozīmē nepieciešamību novērtēt atlīdzību rezervju iespējamu variāciju, vai, ideālajā gadījumā, iespējamo iznākumu pilnu sadalījumu, no kura pēc tam var iegūt kvantiles un citus sadalījuma parametrus. Uz šo brīdi jau tiek piedāvāti vairāki modeļi, kuri saista chain-ladder metodi ar kādu stohastisko modeli pie zināmiem pieņēmumiem. Taču šādu metodoloģiju praktizējošu lietotāju skaits pagaidām ir ierobežots, īpaši Latvijā. Tam var būt dažādi iemesli, piemēram, piemērotas programmatūras trūkums, kā arī nepilnīga metodoloģijas izpratne.

Darbā tiek apskatīta chain-ladder metode vispārinātā lineārā modeļa ietvaros, kurš ir plaši lietots statistiskajā modelēšanā. Šādai pieejai ir tā priekšrocība, ka ar to saistīta teorija var tikt izmantota, lai iegūtu parametru novērtējumus, kā arī piemēroto modeļa atbilstības mēru, izmantojot atlikumus. Vispārinātā lineārā modeļa teorija palīdz izvest analītiskās izteiksmes rezervju novērtējumu prognozes standartklūdai. Atlikumi var būt izmantoti butstrapa procedūrā, lai iegūtu butstrapa standartklūdas.

Otrā svarīga metode, kura tika apskatīta darbā ir butstrapa pielietojums atlīdzību rezervju prognozes novērtēšanai. Analītiskās prognozes kļūdu izteiksmes ietver sarežģītās formulas, kuras grūti novērtēt. No otras puses, butstrapa prognozes kļūdas ir ievērojami viegli iegūt, kā arī butstrapa metode dod iespēju iegūt pilnu prognozes sadalījumu.

Darba mērķis ir apskatīt dažus vispārinātā lineārā modeļa veidus un butstrapa metodi,

pielietot apskatītas metodes praktiski uz datiem un salīdzināt rezultātus, kā arī mēģināt noteikt, kurš no modeļiem ir labāk piemērots konkrētiem datiem. Darbā tiek apskatīti sekojoši vispārinātā lineārā modeļa veidi: virs-izkļiedēts Puasona modelis, chain-ladder Normālās aproksimācijas, Gamma, log-Normālais un Maka modelis. Visas metodes tika realizētas programmā R, kurā tika sastādītas programmas katras metodes analīzei.

Šis darbs sastāv no trim pamatdaļām. Pirmajā daļā tiek aplūkoti vispārinātā lineārā modeļa pamati, atlīdzību rezervēšanas problēmas nostādne, galvenie jēdzieni un vispārinātā lineārā modeļa pielietošana atlīdzību rezervēšanā. Otrajā daļā tiek aplūkotas butstrapa datu pārkārtošanas metodes izmantošanas iespējas atlīdzību rezervēšanā. Trešā nodaļa tiek veltīta šo metožu pielietošanas empīriskajiem rezultātiem: rezervju novērtēšanai, prognozes kļūdas un sadalījuma noteikšanai konkrētiem datiem par apdrošināšanas atlīdzību attīstību 15 periodos, no kuriem tiek izdarīti secinājumi. Darbā pielikumā tiek iekļauts izveidotu programmu R kods.

1. Vispārinātais lineārais modelis

1.1. Vispārinātā lineārā modeļa pamati

1.1.1. Vispārinātā lineārā modeļa struktūra

Vispārinātais lineārais modelis (GLM - Generalised linear model) ir klasiskā lineārā modeļa vispārinājums. Pieņemsim, ka n komponentu novērojumu vektors y ir gadījuma lieluma Y , kura komponentes ir neatkarīgi sadalītas ar vidējām vērtībām μ , realizācija. Modeļa sistemātiskā daļa ir vektora μ specififikācija ar mazu skaitu nezināmu parametru β_1, \dots, β_p palīdzību. Klasiskā lineārā modeļa gadījumā šī specififikācija ir formā

$$\mu = \sum_{j=1}^p x_j \beta_j,$$

kur $\beta = (\beta_1, \dots, \beta_p)$ parasti ir nezināmi parametri, kuru vērtības tiek novērtētas no datiem. Ja i ir novērojumu indekss, tad modeļa sistemātisko komponenti var pierakstīt

$$E(Y_i) = \mu_i = \sum_{j=1}^p x_{ij} \beta_j, \quad i = 1, \dots, n,$$

kur x_{ij} ir j -tā prediktora vērtība i -tam novērojumam. Matricu formā (kur μ ir $n \times 1$, \mathbf{X} ir $n \times p$ un β ir $p \times 1$ matricas) to var pierakstīt

$$\mu = \mathbf{X}\beta,$$

kur \mathbf{X} ir dizaina matrica un β ir parametru vektors. Tādējādi modeļa sistemātiskā komponente ir pilnīgi uzdots.

Vispārinātajā lineārajā modelī parasti tiek pieņemts, ka kļūdas ir neatkarīgas un ar konstantu dispersiju. Turpmāka modeļa specififikācija ietver prasību, lai kļūdas būtu Nor-

māli sadalītas ar konstantu dispersiju σ^2 .

Tātad varām apkopot klasisko lineāro modeli šādi:

Y komponentes ir neatkarīgi Normāli sadalīti mainīgie ar konstantu dispersiju σ^2 un

$$E(Y) = \mu, \quad \text{kur} \quad \mu = \mathbf{X}\beta. \quad (1.1.1)$$

Lai pārietu pie vispārinātā lineārā modeļa, ir nepieciešams pārveidot (1.1.1), lai iegūtu sekojošo specifiku no trim daļām:

1. Gadījuma komponente: Y komponentes ir neatkarīgi Normāli sadalītas ar $E(Y) = \mu$ un konstantu dispersiju σ^2 ;
2. Sistemātiskā komponente: mainīgie x_1, x_2, \dots, x_p veido lineāru prediktoru η

$$\eta = \sum_{j=1}^p x_j \beta_j;$$

3. Saite starp gadījuma un sistemātisko komponenti:

$$\mu = \eta.$$

Šis vispārinājums ievieš jauno simbolu η priekš lineāra prediktora un trešā specifiku daļa nozīmē, ka μ un η faktiski ir vienādi. Pieņemsim, ka $\eta_i = g(\mu_i)$, tad $g(\cdot)$ tiek saukta par saites funkciju. Pie šāda formulējuma klasiskais lineārais modelis ir modelis ar Normāli sadalītu gadījuma komponenti un identitātes saites funkciju. Vispārinātais lineārais modelis vispārina klasisko lineāro modeli divos virzienos. Pirmkārt, gadījuma komponentes sadalījums var būt jebkurš no eksponenciālās funkciju saimes, ne tikai Normālais, un, otrkārt, par saites funkciju var kalpot jebkura monotoni diferencējama funkcija.

Sākumā apskatīsim paplašināto pieņēmumu par sadalījumu. Tiek pieņemts, ka katrai Y komponentei ir sadalījums no eksponenciālās saimes formā

$$f_Y(y; \theta, \phi) = \exp\{(y\theta - b(\theta)) / a(\phi) + c(y, \phi)\}, \quad (1.1.2)$$

kur $a(\cdot)$, $b(\cdot)$ un $c(\cdot)$ ir kādas specifiskās funkcijas. θ ir kanonisks parametrs, kas raksturo izvietojumu, bet ϕ ir mēroga parametrs.

Tā Normālajam sadalījumam

$$\begin{aligned} f_Y(y; \theta, \phi) &= \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{-(y - \mu)^2/(2\sigma^2)\} \\ &= \exp\{(y\mu - \mu^2/2)/\sigma^2 - 1/2(y^2/\sigma^2 + \ln(2\pi\sigma^2))\}, \end{aligned}$$

tātad $\theta = \mu$, $\phi = \sigma^2$ un $a(\phi) = \phi$, $b(\theta) = \theta^2/2$, $c(y, \phi) = -1/2\{y^2/\sigma^2 + \ln(2\pi\sigma^2)\}$.

Vispārinātā lineārā modeļa parametru novērtējumus parasti iegūst maksimizējot vislielākās ticamības funkciju (vai vislielākās ticamības funkcijas logaritmu (*log likelihood*)) novērotiem datiem. Vislielākās ticamības funkcija ir funkcija no parametra θ datiem y_1, \dots, y_n

$$L(y_1, \dots, y_n; \theta) = \prod_{i=1}^n f_Y(y_i; \theta),$$

vai vislielākās ticamības funkcijas logaritms ir

$$l(y_1, \dots, y_n; \theta) = \ln L(y_1, \dots, y_n; \theta) = \ln f_Y(y_1, \dots, y_n; \theta) = \sum_{i=1}^n \ln f_Y(y_i; \theta).$$

Vislielākās ticamības parametra θ novērtējums tiek definēts

$$\hat{\theta}_{ML} = \operatorname{argmax}_{\theta} \{L(y_1, \dots, y_n; \theta)\}.$$

Log-ticamības funkcija, kā funkcija no parametriem θ un ϕ pie dotajiem y ir $l(\theta, \phi; y) = \ln f_Y(y; \theta, \phi)$. Vislielākās ticamības funkcijas atvasinājumi apmierina sekojošas vienādības [1]

$$E\left(\frac{\partial l}{\partial \theta}\right) = 0 \tag{1.1.3}$$

un

$$D\left(\frac{\partial l}{\partial \theta}\right) = -E\left(\frac{\partial^2 l}{\partial \theta^2}\right), \tag{1.1.4}$$

kuras tiek iegūtas diferencējot $\int f_Y(y; \theta) = 1$ pēc θ . Izmantojot šīs vienādības, var tikt novērtēta Y vidējā vērtība un dispersija. No (1.1.2) iegūstam, ka

$$l(\theta; y) = (y\theta - b(\theta)) / a(\phi) + c(y, \phi),$$

no kurienes

$$\frac{\partial l}{\partial \theta} = (y - b'(\theta)) / a(\phi) \tag{1.1.5}$$

un

$$\frac{\partial^2 l}{\partial \theta^2} = -b''(\theta)/a(\phi). \quad (1.1.6)$$

No (1.1.3) un (1.1.5) izriet, ka

$$0 = E\left(\frac{\partial l}{\partial \theta}\right) = (\mu - b'(\theta))/a(\phi),$$

tātad

$$E(Y) = \mu = b'(\theta). \quad (1.1.7)$$

Bet no (1.1.4), (1.1.5) un (1.1.6) seko, ka

$$D\left(\frac{y - b'(\theta)}{a(\phi)}\right) = -E\left(-\frac{b''(\theta)}{a(\phi)}\right)$$

no kā iegūstam

$$D\left(\frac{y - \mu}{a(\phi)}\right) = \frac{b''(\theta)}{a(\phi)}$$

un

$$D(Y) = b''(\theta)a(\phi). \quad (1.1.8)$$

Tādā veidā Y dispersija ir divu funkciju reizinājums; viena no tām, $b''(\theta)$, ir atkarīga tikai no kanoniskā parametra (un tādējādi no vidējā) un tiek saukta par variācijas funkciju. Variācijas funkcija kā funkcija no μ tiek apzīmēta $V(\mu) = b''(\theta)$.

Funkcija $a(\phi)$ parasti ir formā

$$a(\phi) = \phi/w,$$

kur ϕ , ko apzīmē arī ar σ^2 un sauc par dispersijas parametru, ir konstants visiem novērojumiem, un w ir apriori zināmi svāri, kuri mainās no novērojuma uz novērojumu. Tā Normālajā modelī, kurā katrs novērojums ir m neatkarīgu rādījumu vidējais, $a(\phi) = \sigma^2/m$ un $w = m$.

Saites funkcija saista lineāru prediktoru η ar sagaidāmo μ vērtību datiem y . Klasiskajā lineārajā modelī vidējais un lineārs prediktors ir identiski un identitātes saites funkcija ir pieņemama, jo abi η un μ pieņem vērtības reālajā asī. Taču, darbojoties ar skaitu un Puasona sadalījumu, jābūt $\mu > 0$ un tādēļ identitātes saites funkcija ir mazāk pievilcīga, daļēji tāpēc, ka η var būt negatīva kamēr μ nevar būt. Modeļi priekš skaita, balstīti uz neatkarību klasificētajos datos, parasti noved pie multiplikatīviem efektiem un tas tiek

izteikts ar logaritmisko saites funkciju, $\eta = \ln \mu$, ar inverso $\mu = e^\eta$. Tagad aditīvi efekti, kuri ietekmē η pārvēršas par multiplikatīviem efektiem, kuri ietekmē μ un μ pieņem pozitīvas vērtības [1].

1.1.2. Modeļa parametru novērtējums ar vislielākās ticamības funkciju

Parametru β un ϕ vislielākās ticamības novērtējumu (*MLE*) iegūst maksimizējot log-ticamības funkciju [2]

$$l(\beta, \phi; y) = \sum_{i=1}^n \ln f(y_i; \beta, \phi) = \sum_{i=1}^n (y_i \theta_i - b(\theta_i)) / a(\phi) - \sum_{i=1}^n c(y_i, \phi),$$

pie pieņēmuma, ka y_i ir neatkarīgie rezultējošie mainīgie no eksponenciālās saimes.

Apskatīsim parametra β MLE novērtējumu. Lai atrastu maksimumu, $l(\beta, \phi; y)$ atvasināsim pēc β

$$\begin{aligned} \frac{\partial l}{\partial \beta} &= \frac{1}{a(\phi)} \sum_{i=1}^n \left[y_i \frac{\partial \theta_i}{\partial \beta} - \frac{\partial b(\theta_i)}{\partial \theta_i} \frac{\partial \theta_i}{\partial \beta} \right] = \frac{1}{a(\phi)} \sum_{i=1}^n (y_i - \mu_i) \frac{\partial \theta_i}{\partial \beta} \\ &= \frac{1}{a(\phi)} \sum_{i=1}^n (y_i - \mu_i) \frac{\partial \theta_i}{\partial \mu_i} \frac{\partial \mu_i}{\partial \beta}. \end{aligned}$$

Tā ka

$$\frac{\partial \theta_i}{\partial \mu_i} = \left(\frac{\partial \mu_i}{\partial \theta_i} \right)^{-1} = \left(\frac{\partial^2 b(\theta_i)}{\partial \theta_i^2} \right)^{-1} = \frac{1}{V(\mu_i)} \quad (1.1.9)$$

un

$$\frac{\partial \mu_i}{\partial \beta} = \frac{\partial \mu_i}{\partial g(\mu_i)} \frac{\partial g(\mu_i)}{\partial \beta} = \left(\frac{\partial g(\mu_i)}{\partial \mu_i} \right)^{-1} \frac{\partial x_i^T \beta}{\partial \beta} = \left(\frac{\partial g(\mu_i)}{\partial \mu_i} \right)^{-1} x_i^T. \quad (1.1.10)$$

No šīm vienādībām ((1.1.9) un (1.1.10)) seko, ka

$$\frac{\partial l}{\partial \beta} = \frac{1}{a(\phi)} \sum_{i=1}^n \frac{y_i - \mu_i}{V(\mu_i) g(\mu_i)} x_i^T = \frac{1}{a(\phi)} \sum_{i=1}^n (y_i - \mu_i) w_i g(\mu_i) x_i^T, \quad (1.1.11)$$

kur ar w_i tiek apzīmēts $[V(\mu_i) g^2(\mu_i)]^{-1}$. Pierakstīsim (1.1.11) matricu formā

$$\frac{\partial l}{\partial \beta} = \frac{1}{a(\phi)} X^T W G (y - \mu), \quad (1.1.12)$$

kur W ir diagonāla matrica ar elementiem w_i un G ir diagonāla matrica ar elementiem

$g(\mu_i)$. Tādējādi iegūstam, tā sauktos, vislielākās ticamības vienādojumus parametra β novērtēšanai

$$X^T W G y = X^T W G \mu, \quad (1.1.13)$$

kur W , G un μ iekļauj sevī nezināmo parametru β . Parasti tās ir nelineāras funkcijas no β un tāpēc (1.1.13) nevar atrisināt analītiski.

Parametra β lielākās ticamības vienādojumu atrisinājums parasti tiek iegūts ar iteratīvo svērto mazāko kvadrātu metodi. Tā izriet no Fišera iteratīvās metodes (*Fisher scoring*) ticamības funkcijas maksimizēšanai formā

$$\Theta^{(m+1)} = \Theta^{(m)} + I(\Theta^{(m)})^{-1} \left. \frac{\partial l}{\partial \Theta} \right|_{\Theta = \Theta^{(m)}}, \quad (1.1.14)$$

kur (m) apzīmē m -to iterācijas soli, $I(\Theta)$ ir informācijas matrica un Θ ir parametru vektors.

Lai izvestu šo vienādojumu parametram β , jāatrod parametra β informācijas matrica $I(\beta) = -E \left[\frac{\partial^2 l}{\partial \beta \partial \beta^T} \right]$. Lai iegūtu šo matricu, aprēķināsim log-ticamības funkcijas otro atvasinājumu pēc β

$$\frac{\partial^2 l}{\partial \beta \partial \beta^T} = -\frac{1}{a(\phi)} X^T W G \frac{\partial \mu}{\partial \beta^T} + \frac{1}{a(\phi)} X^T \frac{\partial W G}{\partial \beta^T} (y - \mu)$$

un tad

$$-E \left[\frac{\partial^2 l}{\partial \beta \partial \beta^T} \right] = \frac{1}{a(\phi)} X^T W G \frac{\partial \mu}{\partial \beta^T} + 0 = \frac{1}{a(\phi)} X^T W G G^{-1} X = \frac{1}{a(\phi)} X^T W X,$$

kur $W = \text{diag}\{w_i\} = \text{diag}\{[V(\mu_i)g^2(\mu_i)]^{-1}\}$. Tādējādi (1.1.14) vienādojumu priekš parametra β iegūstam formā

$$\beta^{(m+1)} = \beta^{(m)} + (X^T W X)^{-1} X^T W G (y - \mu), \quad (1.1.15)$$

kur W , G un μ ir novērtēti pie $\beta^{(m)}$.

1.1.3. Vislielākās kvazi-ticamības funkcija

Iepriekšējā apakšnodaļā apskatītā vislielākās ticamības funkcija var būt konstruēta, kad datu sadalījums ir zināms. Bet dažreiz mēs precīzi nezinām apriori pareizo sadalījuma

formu. Un šī nezināšana padara ticamības funkcijas konstruēšanu neiespējamu un tādējādi nav iespējams izmantot lielākās ticamības metodi parametru novērtēšanai. Tādēļ būtu nepieciešama metode, kura strādātu tik pat vai gandrīz tik pat labi kā lielākās ticamības funkcija, tikai bez specifiskiem pieņēmumiem par sadalījumu. Vislielākās kvazi-ticamības funkcijas pamatideja ir iegūt ticamības funkcijai līdzīgu objektu, kura konstruēšana prasa mazāk pieņēmumu.

Definēsim ticamības funkcijas analogu izmantojot (1.1.3) un (1.1.4), izņemot to, ka diferencēšana notiek pēc μ , nevis θ . Pirmkārt, no (1.1.3) gribam iegūt

$$E \left[\frac{\partial \ln f_Y(y_i)}{\partial \mu_i} \right] = 0. \quad (1.1.16)$$

Apzīmējot ar v^* dispersiju $D \left(\frac{\partial \ln f_Y(y_i)}{\partial \mu_i} \right)$ veiksīm sekojošus pārveidojumus

$$\begin{aligned} v^* &= D \left(\frac{\partial \ln f_Y(y_i)}{\partial \mu_i} \right) = D \left(\frac{\partial \ln f_Y(y_i)}{\partial \theta_i} \frac{\partial \theta_i}{\partial \mu_i} \right) \\ &= \left[D \left(\frac{\partial \ln f_Y(y_i)}{\partial \theta_i} \right) \right] \left(\frac{\partial \theta_i}{\partial \mu_i} \right)^2 = \left[-E \left(\frac{\partial^2 \ln f_Y(y_i)}{\partial \theta_i^2} \right) \right] \left(\frac{\partial \theta_i}{\partial \mu_i} \right)^2 \\ &= \frac{1}{a(\phi)} \left[\frac{\partial^2 b(\theta_i)}{\partial \theta_i^2} \right] \left(\frac{\partial \theta_i}{\partial \mu_i} \right)^2 = \frac{V(\mu_i)}{a(\phi)} \left(\frac{\partial \theta_i}{\partial \mu_i} \right)^2 = \frac{V(\mu_i)}{a(\phi)} \frac{1}{V^2(\mu_i)}, \end{aligned} \quad (1.1.17)$$

kur tiek izmantota sakarība (1.1.4), $f_Y(y_i)$ veids (1.1.2), $V(\mu)$ definīcija pēc (1.1.8) un sakarība (1.1.9). Tādā veidā esam ieguvuši

$$D \left(\frac{\partial \ln f_Y(y_i)}{\partial \mu_i} \right) = -E \left(\frac{\partial^2 \ln f_Y(y_i)}{\partial \theta_i^2} \right) = \frac{1}{a(\phi)V(\mu_i)}. \quad (1.1.18)$$

Ievērosim, ka sakarības (1.1.16) un (1.1.18) ir analogas ar (1.1.3) un (1.1.4). Tātad $\partial \ln f_Y(y_i)/\partial \mu_i$ vietā ir jāatrod tādu objektu, kuram izpildās īpašības (1.1.16) un (1.1.18). Acīmredzami, ka

$$q_i = \frac{y_i - \mu_i}{a(\phi)V(\mu_i)} \quad (1.1.19)$$

apmierina šos nosacījumus, proti

$$\begin{aligned} E(q_i) &= E \left(\frac{y_i - \mu_i}{a(\phi)V(\mu_i)} \right) = \frac{E(y_i) - \mu_i}{a(\phi)V(\mu_i)} = 0, \\ D(q_i) &= D \left(\frac{y_i - \mu_i}{a(\phi)V(\mu_i)} \right) = \frac{D(y_i)}{a^2(\phi)V^2(\mu_i)} = \frac{V(\mu_i)a(\phi)}{a^2(\phi)V^2(\mu_i)} = \frac{1}{a(\phi)V(\mu_i)}. \end{aligned}$$

$a(\phi)$, kura parādās (1.1.19) ir tikai proporcionalitātes konstante starp $D(y_i)$ un $V(\mu_i)$, kura nav precīzi tā pati $a(\phi)$, kas parādās blīvuma funkcijā (1.1.2). Taču, lietosim to pašu apzīmējumu, jo kā redzēsīm tālāk, tās spēlē vienādu lomu.

Tā kā y_i ieguldījums log-ticamības funkcijā ir integrālis pēc μ_i no $\partial \ln f_Y(y_i)/\partial \mu_i$, log kvazi-ticamības funkciju definēsīm caur y_i ieguldījumu tajā

$$Q_i = \int_{y_i}^{\mu_i} \frac{y_i - t}{a(\phi)V(t)} dt, \quad (1.1.20)$$

kura atvasinājums pēc μ_i vienāds ar q_i pēc definīcijas. Beidzot, lai atrastu parametra β vislielākās kvazi-ticamības novērtējumu ir jāatrisina lielākās kvazi-ticamības vienādojumi

$$\frac{\partial}{\partial \beta} \sum Q_i = 0. \quad (1.1.21)$$

Novērtējot atvasinājumu vienādojumā (1.1.21) iegūstam

$$\sum \frac{y_i - \mu_i}{a(\phi)V(\mu_i)} \frac{\partial \mu_i}{\partial \beta} = 0,$$

kurš izmantojot (1.1.10) ir formā

$$\sum \frac{y_i - \mu_i}{a(\phi)V(\mu_i)g(\mu_i)} x_i^T = 0,$$

vai matricu formā

$$\frac{1}{a(\phi)} X^T W G (y - \mu),$$

ir tāds pats kā (1.1.12). Ir jāievēro, ka definējot vislielākās kvazi-ticamības novērtējumu kā vislielākās kvazi-ticamības vienādojumu (1.1.21) atrisinājumu, mēs izvairāmies no īstās maksimizēšanas problēmas risināšanas vai pat no pašas vislielākās kvazi-ticamības vai log kvazi-ticamības funkcijas definēšanas.

Tas ir ievērojams rezultāts, jo Q_i ir konstruēta izmantojot tikai informāciju par to, kā mainās dispersija atkarībā no μ un neko citu. Bieži gadās tā, ka ja mēs uzdodam sakarību starp vidējo un dispersiju, mēs iegūstam vislielākās kvazi-ticamības vienādojumus, kuri pilnīgi sakrīt ar tiem, kuri atbilst īstai vislielākās ticamības funkcijai.

Vislielākās kvazi-ticamības funkcijai ir svarīga priekšrocība salīdzinot ar parasto vislielākās ticamības funkciju. Pieņemsim, ka ir regresija, kuras rezultējošais mainīgais ir

sadalīts pēc Puasona likuma. Izmantojot vislielākās ticamības funkciju pieņēmums būs tāds, ka $D(y_i) = V(\mu_i)$. Tomēr praksē bieži izrādās tā, ka dati ir no sadalījuma, kur dispersija ir lielāka nekā vidējais. Ja dispersija ir proporcionāla vidējam, tad modeļa specifikācija izmantojot vislielākās kvazi-ticamības funkciju ir joprojām korekta, jo pieņēmums ir tikai par to, ka $D(y_i) = a(\phi)V(\mu_i)$, t.i., $D(y_i)$ ir proporcionāla $V(\mu_i)$ un nav obligāti vienāda.

Tātad vislielākās kvazi-ticamības funkcijas izmantošana ļauj mums, pirmkārt, netaisīt pieņēmumus par sadalījumu un, otrkārt, mums ir nepieciešams uzdot sakarību starp vidējo un dispersiju ar precizitāti tikai līdz proporcionalitātes konstantei, kura var būt novērtēta no datiem.

Vispārpieņemts proporcionalitātes konstantes $a(\phi)$ novērtējums ir ar momentu metodi iegūstamais novērtējums [1]

$$a(\phi) = \frac{1}{n-p} \sum_i \frac{(y_i - \mu_i)^2}{V(\mu_i)} = \frac{X^2}{n-p},$$

kur n ir novērojumu skaits, p ir vektora β dimensija un X^2 ir vispārinātā Pīrsona statistika.

1.1.4. Modeļa pielāgošanas kvalitātes mērs un modeļa atlikumi

Modeļa pielāgošanas kvalitātes mēri var būt formulēti dažādos veidos, apskatīsim tā saukto novirzes funkciju (*deviance*), kuru veido vislielākās ticamības funkciju attiecības logaritms.

Esot n novērojumiem tiem ir iespējams pielāgot modeļus ar līdz n parametriem. Nulles modelī ir viens parametrs, kurš reprezentē visiem y kopējo parametru μ ; tātad nulles modelis visu starp y vērtībām esošu variāciju izskaidro ar gadījuma komponenti. Pilnajā modelī ir n parametri, viens katram novērojumam, un μ , kuri iegūti novērtējot šo modeli pilnīgi atbilst datiem.

Izteiksim log-ticamības funkciju kā funkciju no parametra μ , nevis θ . Pieņemsim, ka $l(\hat{\mu}, \phi; y)$ ir log-ticamības funkcija maksimizēta pēc β ar fiksētu mēroga parametru ϕ . Maksimālā ticamības funkcijas vērtība, kura ir sasniedzama pilnajā modelī ar n parametriem ir $l(y, \phi; y)$ un parasti ir galīga. Pielāgošanas neatbilstības mērs ir proporcionāls divkārsībai starpībai starp maksimālo sasniedzamo log-ticamības funkcijas vērtību un vērtību, kura ir sasniegta pētāmajā modelī. Apzīmējot ar $\hat{\theta} = \theta(\hat{\mu})$ un $\tilde{\theta} = \theta(y)$ abu modeļu

kanonisku parametru novērtējumus un pieņemot, ka $a_i(\phi) = \phi/w_i$, modeļa pielāgošanas neatbilstības mērs var būt formulēts šādi

$$\sum 2w_i\{y_i(\tilde{\theta}_i - \hat{\theta}_i) - b(\tilde{\theta}_i) + b(\hat{\theta}_i)\}/\phi = D(y; \hat{\mu})/\phi,$$

kur modeļa novirzes funkcija $D(y; \hat{\mu})$ ir funkcija tikai no datiem.

Normālā sadalījuma gadījumā novirzes funkcija ir vienkārši atlikumu kvadrātu summa. Novirzes funkcija dalīta ar mēroga parametru ir svērtā novirzes funkcija (*scaled deviance*)

$$D(y; \hat{\mu})/\phi = D^*(y; \hat{\mu}) = 2l(y; y) - 2l(\mu; y),$$

kur eksponenciālās saimes modeļu gadījumā $l(y; y)$ ir vislielākās ticamības funkcijas vērtība, kura tiek sasniegta, kad pielāgotās vērtības sakrīt ar novērotiem datiem.

Cits svarīgs modeļa neatbilstības mērs ir vispārinātā Pīrsona X^2 statistika

$$X^2 = \sum (y - \hat{\mu})^2/V(\hat{\mu}),$$

kur $V(\hat{\mu})$ ir attiecīga sadalījuma variācijas funkcijas novērtējums. Normālā sadalījuma gadījumā X^2 , tāpat ka novirzes funkcija, ir vienāds ar atlikumu kvadrātu summu.

Abām, novirzes funkcijai un vispārinātajam Pīrsona X^2 , ir precīzi χ^2 sadalījums Normālo lineāro modeļu gadījumā (protams, ja modelis ir spēkā) un ir pieejami asimptotiski rezultāti priekš citiem sadalījumiem.

Normālajā modelī atkarīgais mainīgais var būt izteikts formā

$$y = \hat{\mu} + (y - \hat{\mu}),$$

t.i., dati = pielāgoti dati + atlikums. Atlikumi var būt izmantoti, lai izpētītu modeļa pielāgošanas kvalitāti attiecībā uz variācijas funkcijas, saites funkcijas un lineāra prediktora komponentu izvēli. Atlikumi arī var norādīt uz anomālu vērtību klātbūtni, kas prasa papildus izpēti. Vispārinātajā lineārajā modelī tiek prasīta vispārīga atlikumu definīcija, kura būtu derīga visiem sadalījumiem, kuri var aizvietot Normālo. Ērtā ir, ja šos atlikumus var izmantot tiem pašiem mērķiem, kā standarta Normālā atlikumus.

Tālāk definēsim divas visbiežāk lietojamas vispārināto atlikumu formas, proti Pīrsona un novirzes atlikumus. Turpmākajā teorētiskajā aprakstā $\hat{\mu}$ vietā izmantosim μ .

Definīcija 1. Pīrsona atlikumi tiek definēti kā parastie atlikumi dalīti ar Y standartnovirzes novērtējumu

$$r_P = \frac{y - \mu}{\sqrt{V(\mu)}}.$$

Šāds nosaukums ir izvēlēts tāpēc, ka Puasona sadalījuma gadījumā Pīrsona atlikumi ir vienkārši Pīrsona X^2 statistikas komponentu kvadrātsaknes ar attiecīgu zīmi, tādi, ka $\sum r_P^2 = X^2$. Taču šeit Pīrsona statistika tiks lietota vairāk kā atlikumu variācijas mērs, nevis kā modeļa pielāgošanas kvalitātes mērs.

Ja novirzes funkcija tiek izmantota kā vispārinātā lineārā modeļa neatbilstības mērs, tad katra vienība tajā iegulda lielumu d_i , tādu, ka $\sum d_i = D$.

Definīcija 2. Novirzes atlikumi tiek definēti kā

$$r_D = \text{sign}(y - \mu)\sqrt{d_i},$$

kur $\sum d_i = D$ un D ir novirzes funkcija.

Pie šādas novirzes atlikumu definīcijas iegūstam lielumu, kurš pieaug ar $y_i - \mu_i$ un kuram $\sum r_D^2 = D$. Tādā veidā Puasona sadalījumam

$$r_D = \text{sign}(y - \mu)\{2(y \ln(y/\mu) - y + \mu)\}^{1/2}.$$

1.2. Vispārinātā lineārā modeļa izmantošana IBNR rezervju noteikšanai

1.2.1. Atlīdzību rezervēšanas problēmas nostādne un galvenie jēdzieni

Apdrošināšanas IBNR atlīdzību rezervēšanas problēmu var apkopot sekojošā veidā: novērtēt turpmāko atlīdzību summu (jeb atlīdzību skaitu), par pagātnes periodos notikušajiem apdrošināšanas gadījumiem, kuri vēl būs pieteikti nākotnē, ja ir pieejama informācija par pagātni. Pie tam ir nepieciešams noteikt piesardzīgu rezervju augšējo robežu ar pietiekamo nozīmības līmeni.

Datus par apdrošināšanas kompānijas izmaksātajām atlīdzībām (vai atlīdzību skaitu)

uzskatāmi var apkopot tā sauktajā *attīstības trijstūrī*. Turpmāk, nezaudējot vispārīgumu, uzskatīsim, ka C_{ij} ir atlīdzību summas pieaugumi. Pieņemsim, ka mēs atrodamies n -jā periodā un zinām visu pagātnes informāciju. Katra notikušā apdrošināšanas gadījuma prasība (turpmāk: notikums) tiek atlīdzināta vai nu notikuma periodā vai arī kādā no sekojošajiem n attīstības periodiem, bet izmaksāto atlīdzību summa izvēlētajos periodos raksturo apdrošināšanas kompānijas zaudējumu sadalījumu pa šiem periodiem. Attīstības trijstūrī izmaksātās atlīdzības iespējams apkopot kumulētā vai pieaugumu formā.

Veidojot attīstības trijstūri atlīdzību pieaugumu formā, tiek aplūkota gadījumu lielumu saime $\{C_{ij} : i = 1, \dots, n; j = 1, \dots, n\}$, kur gadījuma lielums C_{ij} tiek interpretēts kā notikuma perioda i zaudējumi, kas atlīdzināti j -jā attīstības periodā pēc gadījuma iestāšanās jeb kalendārajā periodā $i + j - 1$. C_{ij} tiek saukts par notikuma perioda i zaudējumu atlīdzību pieaugumu attīstības periodā j . Atlīdzību pieaugumi ir novērojami kalendārajiem periodiem $i + j + 1 \leq n$ un nenovērojami kalendārajiem periodiem $i + j + 1 > n$. Novērojami atlīdzību pieaugumi $\{C_{ij} : i = 1, \dots, n; j = 1, \dots, n - i + 1\}$ veido attīstības trijstūri, kas parādīts 1.1. tabulā.

1.1. tabula Attīstības trijstūris

Notikuma periods	Attīstības periods						
	1	2	...	j	...	$n - 1$	n
1	$C_{1,1}$	$C_{1,2}$...	$C_{1,j}$...	$C_{1,n-1}$	$C_{1,n}$
2	$C_{2,1}$	$C_{2,2}$...	$C_{2,j}$...	$C_{2,n-1}$	
⋮	⋮	⋮	...	⋮			
i	$C_{i,1}$	$C_{i,2}$...	$C_{i,n-i+1}$			
⋮	⋮	⋮					
$n - 1$	$C_{n-1,1}$	$C_{n-1,2}$					
n	$C_{n,1}$						

Uz attīstības trijstūra diagonāles atrodas pēdējie novērotie atlīdzību pieaugumi, tādēļ šo attīstības trijstūri sauksim par *pagātnes trijstūri*. Pa labi no attīstības diagonāles atrodas nenovērojami (nākotnes) atlīdzību pieaugumi, tāpēc šo datu trijstūri sauksim par *nākotnes trijstūri*.

Attīstības trijstūru var izveidot arī kumulēto atlīdzību formā. Šajā gadījumā tiek apskatīta gadījuma lielumu saime $\{D_{ij} : i = 1, \dots, n; j = 1, \dots, n\}$, kur gadījuma lielums D_{ij} tiek definēts kā i -tā notikuma perioda atlīdzību pieaugumu summa līdz attīstības

periodam j

$$D_{ij} = \sum_{k=1}^j C_{ik}.$$

D_{ij} tiek interpretēts kā i -tā notikuma perioda summārie zaudējumi, kas atlīdzināti ar ne vairāk kā $j - 1$ periodu nobīdi jeb ne vēlāk kā j -jā attīstības periodā. D_{ij} sauksim par i -tā notikuma perioda un j -tā attīstības perioda kumulētajam zaudējumam atlīdzībām. $D_{i,n-i}$ apzīmē līdz šim brīdim kumulētu zaudējumu atlīdzību summu par i -tā notikuma perioda apdrošināšanas gadījumiem, savukārt D_{in} apzīmē galīgu kumulētu zaudējumu atlīdzību summu par i -tā notikuma perioda apdrošināšanas gadījumiem.

Vairāk nekā individuālo vērtību C_{ij} vai D_{ij} ($i = 1, 2, \dots, n$ un $j = n - i + 2, n - i + 3, \dots, n$) prognozēšana, interesē rindu kopsummu prognozēšana, t.i., rezervju summas nepieciešamas, lai segtu zaudējumus, kuri notikuši i -jā periodā, un īpaši summāro prognozi, kurš apzīmē sagaidāmas summāras nepieciešamas rezerves. Citiem vārdiem, i -tajam notikuma periodam nepieciešamas rezerves var būt aprēķinātas kā starpība starp attiecīgā notikuma perioda galīgo atlīdzību summu un līdz šim brīdim izmaksāto atlīdzību summu, t.i., $R_i = D_{in} - D_{i,n-i+1}$; un tad summārās rezerves $R = \sum_i R_i$.

Iegūstot rezervju summas novērtējumu, dabisks nākamais solis ir attīstīt novērtējumu iespējamai iegūtas prognozes variācijai, lai varētu izdarīt secinājumus par to vai ir nepieciešama papildus rezervju uzlikšana virs prognozētās vērtības piesardzības nolūkos. Vispār sakot, lai varētu noteikt rezervju augšējo robežu ar pietiekamo nozīmības līmeni. Variācijas mērs, kurš parasti tiek lietots šajā sakarā ir prognozes kļūda, definēta kā iespējamo rezervju iznākumu sadalījuma standartnovirze.

1.2.2. Zaudējumu atlīdzību rezervju prognozes kļūda

Prognozes kļūda ir kvadrātsakne no prognozes vidējās kvadrātiskās kļūdas (RMSE). Vidējā kvadrātiskā kļūda (MSE) D_{in} novērtējumam \hat{D}_{in} tiek definēta kā

$$MSE(\hat{D}_{in}) = E((\hat{D}_{in} - D_{in})^2 | T_p),$$

kur $T_p = \{D_{ij} | i + j \leq n + 1\}$ ir visu līdz šim brīdim novērotu datu kopa. Jāatzīmē, ka šeit netiek izmantota nenosacīta vidējā kvadrātiskā kļūda $E((\hat{D}_{in} - D_{in})^2) = E(E((\hat{D}_{in} - D_{in})^2 | T_p))$, jo tā vidējo pa visiem iespējamiem datiem T_p no datu ģenerējoša sadalījuma. Tajā vietā praksē mūs vairāk interesē nosacīta vidējā kvadrātiskā kļūda

konkrētam novērtējumam \hat{D}_{in} , kurš balstās uz konkrētu novērotu datu kopu T_p un tāpēc ir jāizmanto $E((\hat{D}_{in} - D_{in})^2|T_p)$, kura dos tikai vidējo novirzi starp \hat{D}_{in} un D_{in} nākotnes nejaušības dēļ.

Vispirms, ievērosim, ka

$$MSE(\hat{R}_i) = E((\hat{R}_i - R_i)^2|T_p) = E((\hat{D}_{in} - D_{in})^2|T_p) = MSE(\hat{D}_{in}).$$

Tad, izmantojot vispārīgo likumu $E(X - a)^2 = D[X] + (EX - a)^2$, iegūstam

$$MSE(\hat{D}_{in}) = D[D_{in}|T_p] + (E(D_{in}|T_p) - \hat{D}_{in})^2 = D[D_{in}|T_p] + D[\hat{D}_{in}|T_p], \quad (1.2.1)$$

kas parāda, ka vidējā kvadrātiskā kļūda ir stohastiskās kļūdas (procesa dispersijas) un novērtējuma kļūdas summa.

1.2.3. Chain-ladder metode

Pieņemsim, ka ir dots kumulēto atlīdzību attīstības trijstūris $\{D_{ij} : i = 1, \dots, n; j = 1, \dots, n - i + 1\}$. Attīstības faktoru matrica kumulēto atlīdzību kopai $\{D_{ij}\}$ sastāv no faktoriem λ_{ij} , kuri tiek aprēķināti pēc formulas

$$\lambda_{ij} = \frac{D_{ij}}{D_{i,j-1}}, \quad i = 1, \dots, n; \quad j = 2, \dots, n - i + 1.$$

Pirmais pamatpieņēmums ir sekojošs [3].

Pieņēmums 1. Katram notikuma periodam ir viens un tas pats attīstības faktors, t.i., katram $j = 2, \dots, n$ izpildās

$$\lambda_{ij} = \lambda_j \quad \forall i = 1, \dots, n.$$

Saskaņā ar 1. pieņēmumu, vispopulārākais attīstības faktoru λ_j novērtējums ir svērtais vidējais

$$\hat{\lambda}_j = \frac{\sum_{i=1}^{n-j+1} D_{ij}}{\sum_{i=1}^{n-j+1} D_{i,j-1}} = \frac{\sum_{i=1}^{n-j+1} D_{i,j-1} \lambda_{ij}}{\sum_{i=1}^{n-j+1} D_{i,j-1}}. \quad (1.2.2)$$

Attīstības faktors λ_{ij} ir nosvērts ar attiecīgo “apjoma” mēru $D_{i,j-1}$.

Nākotnes trijstūra prognozētas vērtības \hat{D}_{ij} tiek aprēķinātas pēc formulas

$$\hat{D}_{ij} = D_{i,n-i+1} \prod_{k=n-i+2}^j \hat{\lambda}_k, \quad i = 2, \dots, n, \quad j = n - i + 2, \dots, n.$$

Šis prognozēšanas paņēmieni balstās uz to, ka otrais modeļa pieņēmums ir spēkā. Tiek pieņemts, ka katram notikuma periodam obligāti ir atšķirīgs līmenis, kurš tiek novērtēts ar attiecīga notikuma perioda individuālo pieredzi. Lielums $D_{i,n-i+1}$ reprezentē notikuma perioda līmeņa novērtējumu.

Pieņēmums 2. Katram notikuma periodam ir parametrs, kurš raksturo to līmeni. i -tā notikuma perioda līmeņa parametrs ir novērtēts ar $D_{i,n-i+1}$.

Pēdējo notikuma periodu pārstāv viens vienīgs novērojums D_{i1} . Ja būtu pieņēmuši, ka notikuma periodi ir pilnīgi homogēni, tad notikuma periodu līmenis būtu jānovērtē ar $\sum_{i=1}^n D_{i1}/n$ (vai labāku novērtējumu 1. attīstības perioda vidējam līmenim). Pilnīga homogenitāte nozīmē, ka novērojumi $D_{11}, D_{21}, \dots, D_{n1}$ ir uzģenerēti ar vienu un to pašu procesu. Chain-ladder metode pieņem, ka procesi, kuri uzģenerēja atlīdzību pieaugumus $D_{11}, D_{21}, \dots, D_{n1}$ ir tik nesaistīti, ka informācijas apvienošana nesniedz nekādu efektivitātes uzlabojumu. Vispār ir ļoti grūti noticēt, ka šāds pieņēmums kādreiz varētu būt spēkā praksē.

Chain-ladder metode dod iespēju vienkārši atrast rezervju prognozi, bet šī metode nedod iespēju novērtēt šīs prognozes kļūdu. Tāpēc tālāk šajā nodaļā tiks apskatīti daži statistiski modeļi, kuri dod tos pašus vai līdzīgus novērtējumus kā chain-ladder metode, bet dod arī iespēju novērtēt prognozes kļūdu.

1.2.4. Virs-izklaidēts Puasona modelis

Šajā apakšnodaļā aprakstīsim saistību starp *Chain-ladder* metodi un sekojošo statistisko modeli priekš atlīdzību pieaugumiem C_{ij} [4].

$$C_{ij} \quad \forall i, j \text{ neatkarīgi sadalīts pēc Puasona sadalījuma ar vidējo vērtību } \mu_{ij}, \quad (1.2.3)$$

$$\ln \mu_{ij} = c + \alpha_i + \beta_j \quad \text{un } \alpha_1 = \beta_1 = 0. \quad (1.2.4)$$

Turpmāk pieņemsim, ka

$$\sum_{i=1}^{n-j+1} C_{ij} \geq 0 \quad \text{visiem } j. \quad (1.2.5)$$

Šī pieņēmuma iemesls kļūs skaidrs pēc vienādojuma (1.2.7). Jāievēro, ka mēs neprasām, lai visi atlīdzību pieaugumi būtu nenegatīvi, bet tikai kolonu summas. Puasona

modelēšanas sadalījuma izmantošana nenozīmē, ka datiem, jābūt pozitīviem, veseliem skaitļiem; ir iespējams izmantot vislielākās kvazi-ticamības funkciju, kurai ir visas Puasona vislielākās ticamības funkcijas īpašības, bez faktiskās atsauces uz Puasona sadalījuma blīvuma funkciju. Tas nozīmē, ka šis modelis var būt pielietots negatīviem atlīdzību pieaugumiem.

Šobrīd ir vienkāršāk saglabāt pieņēmumu, ka datiem ir Puasona sadalījums, kaut gan visur tālāk tā ir tikai vislielākās ticamības funkcijas forma, kura ir svarīga. Acīmredzot, šis modelis ir saprātīgs, kad trijstūris sastāv no atlīdzību skaita, nevis atlīdzību summām. Tādēļ, lai parādītu ekvivalenci ar chain-ladder metodi, vispirms pieņemsim, ka attīstības trijstūri veido atlīdzību skaits sagrupēts pēc notikuma perioda i un attīstības perioda (kurš šajā gadījumā varētu būt atlīdzības pieteikšanas periods) j .

Modeļa (1.2.3), (1.2.4) parametru novērtējumi tiek iegūti ar vislielākās ticamības metodi. Novērtējums summāram atlīdzību skaitam tiek iegūts no vienādības

$$\hat{D}_{in} = \hat{D}_{i,n-i+1} + \sum_{j=n-i+2}^n \exp(\hat{c} + \hat{\alpha}_i + \hat{\beta}_j), \quad i = 2, 3, \dots, n, \quad (1.2.6)$$

kur \hat{c} , $\hat{\alpha}_i$, $\hat{\beta}_j$ ir vislielākās ticamības parametru novērtējumi.

Tie paši novērtējumi $\{\hat{D}_{in} : i = 2, \dots, n\}$ tiek iegūti ar sekojošo nosacīto vislielākās ticamības funkciju. Apzīmēsim nosacītu varbūtību, ka i -tā notikuma perioda atlīdzība, kura mēs zinām ir pieteikta, ir pieteikta attīstības periodā j , ar $p_{j|i}$. Notikuma periodam i , tā ir varbūtība, ka attīstības periods ir j , pie nosacījuma, ka tas ir mazāks vai vienāds ar $n - i + 1$. Parasts pieņēmums par to, ka atlīdzību pieteikšanas process ir stacionārs nozīmē, ka varbūtība tam, ka atlīdzība tiek pieteikta katrā attīstības periodā, nav atkarīga no notikuma perioda. Tātad

$$p_{j|i} = \frac{p_j}{\sum_{k=1}^{n-i+1} p_k}, \quad (1.2.7)$$

kur p_j ir varbūtība, ka atlīdzība ir pieteikta attīstības periodā j un $\sum_{k=1}^n p_k = 1$.

Pēdējais pieņēmums nozīmē, ka visas atlīdzības ir pieteiktas līdz n -tā attīstības perioda beigām. Tas arī ir saskaņots ar chain-ladder metodi vispār, kura nenodarbojas ar prognozēšanu pēc D_{in} , t.i., pēc pēdēja attīstības perioda, kurš jau ir novērots. Tas arī izskaidro ierobežojuma (1.2.5) nepieciešamību, jo bez tā mēs nevarētu izmantot varbūtības.

Nosacītā vislielākās ticamības funkcija l_c šiem datiem var būt iegūta izmantojot Mul-

tinomiālā sadalījuma blīvuma funkciju [4]

$$f(x_1, \dots, x_k; N; p_1, \dots, p_k) = \frac{N!}{x_1! \dots x_k!} p_1^{x_1} \dots p_k^{x_k}, \quad \sum_{i=1}^k x_i = N.$$

Mūsu gadījumā x_i ir vērtības C_{ij} , bet $k = n - i + 1$ ir pēdēja novērota attīstības perioda indekss katram notikuma periodam i . Vislielākās ticamības funkcija ir nosacītā, jo tā ir pie nosacījuma, ka visvēlākā novērota rindas summa ir $D_{i,n-i+1}$, un ir sekojošā formā

$$L_c = \prod_{i=1}^n \left(\frac{D_{i,n-i+1}!}{\prod_{j=1}^{n-i+1} C_{ij}!} \prod_{j=1}^{n-i+1} p_{j|i}^{C_{ij}} \right). \quad (1.2.8)$$

Maksimizējot šo ticamības funkciju iegūstam sekojošus summāra atlīdzību skaita novērtējumus

$$\hat{D}_{in} = \frac{D_{i,n-i+1}}{1 - \sum_{j=n-i+2}^n \hat{p}_j}, \quad i = 2, \dots, n, \quad (1.2.9)$$

kur \hat{p}_j ir p_j novērtējums, kurš iegūts maksimizējot L_c . Var parādīt, ka Puasona modelis dod tos pašus novērtējumus kā nosacītā vislielākās ticamības funkcija, t.i., ka vienādojumi (1.2.6) un 1.2.9 dod vienādus rezultātus.

Dažreiz ir ērtāk apskatīt prognozi notikuma perioda atlīdzību skaitam, kurš bija pieņemts līdz attīstības periodam j vienādības (1.2.9) vietā. Galīgs notikuma perioda $n-j+1$ kumulēts atlīdzību skaits ir

$$\hat{D}_{n-j+1,n} = \frac{D_{n-j+1,j}}{1 - \sum_{k=j+1}^n \hat{p}_k}. \quad (1.2.10)$$

Tas var būt salīdzināts ar chain-ladder metodes prognozēm

$$\hat{D}_{n-j+1,n} = D_{n-j+1,j} \hat{\lambda}_{j+1} \hat{\lambda}_{j+2} \dots \hat{\lambda}_n, \quad (1.2.11)$$

kur

$$\hat{\lambda}_j = \frac{\sum_{i=1}^{n-j+1} D_{ij}}{\sum_{i=1}^{n-j+1} D_{i,j-1}}. \quad (1.2.12)$$

Rozenbergs 1990. gadā piedāvāja rekursīvo formulu novērtējumu \hat{p}_j , $j = 1, \dots, n$ iegūšanai. Pieņemsim, ka ir doti $p_n, p_{n-1}, \dots, p_{j+1}$ novērtējumi, tad p_j novērtējums \hat{p}_j var būt

iegūts ar formulu

$$\hat{p}_j = \frac{C_{1j} + C_{2j} + \dots + C_{n-j+1,j}}{D_{1n} + \frac{D_{2,n-1}}{1-\hat{p}_n} + \dots + \frac{D_{n-j+1,j}}{1-\hat{p}_{j+1}-\dots-\hat{p}_n}} \quad (1.2.13)$$

$$= \frac{C_{1j} + C_{2j} + \dots + C_{n-j+1,j}}{D_{1n} + \hat{D}_{2,n} + \dots + \hat{D}_{n-j+1,n}}. \quad (1.2.14)$$

Rekursija sākās ar $\hat{p}_n = C_{1n}/D_{1n}$. Var pierādīt, ka iegūtie novērtējumi \hat{p}_j , $j = 1, \dots, n$ ir vislielākās ticamības novērtējumi funkcijai L_c .

Teorēma 1. *Novērtējumi, kuri tiek iegūti maksimizējot nosacīto vislielākās ticamības funkciju L_c var būt iegūti rekursīvi no vienādojuma (1.2.13).*

Pierādījums. Ir jāatrod novērtējumi, kuri maksimizē

$$L_c = \prod_{i=1}^n \left(\frac{D_{i,n-i+1}!}{\prod_{j=1}^{n-i+1} C_{ij}!} \prod_{j=1}^{n-i+1} p_{j|i}^{C_{ij}} \right),$$

kas ir ekvivalents $\ln L_c$ maksimizēšanai. Lai atrisinātu šo problēmu, apskatīsim tikai tos $\ln L_c$ locekļus, kuri iekļauj $p_{j|i}$, tātad apskatīsim

$$l_c = \sum_{i=1}^n \sum_{j=1}^{n-i+1} C_{ij} \ln p_{j|i} = \sum_{i=1}^n \sum_{j=1}^{n-i+1} C_{ij} \left(\ln p_j - \ln \sum_{k=1}^{n-i+1} p_k \right) \quad \text{izmantojot (1.2.7).}$$

Ievērojot, ka p_n parādās tikai locekļos, kuros $i = 1$, iegūstam

$$\frac{\partial l_c}{\partial p_n} = \frac{C_{1n}}{p_n} - \sum_{j=1}^n \frac{C_{1j}}{\sum_{k=1}^n p_k}$$

un no $\partial l_c / \partial p_n = 0$ izriet, ka $\hat{p}_n = C_{1n} / \sum_{j=1}^n C_{1j}$, jo $\sum_{k=1}^n p_k = 1$. Tādējādi $\hat{p}_n = C_{1n} / D_{1n}$.

Tagad pieņemsim, ka teorēma ir spēkā priekš $j + 1, j + 2, \dots, n$. Atvasinot l_c pēc p_j iegūstam

$$\frac{\partial l_c}{\partial p_j} = \sum_{i=1}^{n-j+1} \left(\frac{C_{ij}}{p_j} - \sum_{j=1}^{n-i+1} \frac{C_{ij}}{\sum_{k=1}^{n-i+1} p_k} \right) = \frac{\sum_{k=1}^{n-i+1} C_{ij}}{p_j} - \sum_{i=1}^{n-j+1} \left(\frac{D_{i,n-i+1}}{\sum_{k=1}^{n-i+1} p_k} \right)$$

un no $\partial l_c / \partial p_j = 0$ izriet, ka

$$\hat{p}_j = \frac{\sum_{i=1}^{n-j+1} C_{ij}}{\sum_{i=1}^{n-j+1} \left(\frac{D_{i,n-i+1}}{\sum_{k=1}^{n-i+1} p_k} \right)}.$$

Tā kā $\sum_{k=1}^{n-i+1} p_k = 1 - \sum_{k=n-i+2}^n p_k$, ja $i > 1$ un $\sum_{k=1}^n p_k = 1$, tas ir ekvivalents (1.2.13).

Tādā veidā teorēma ir pierādīta pēc indukcijas. \square

Tagad parādīsim, ka Rozenberga metode ir ekvivalenta chain-ladder metodei. Tas parādīs to, ka chain-ladder metode ir tikai vienkāršs veids, kā iegūt vislielākās ticamības novērtējumus nosacītai ticamības funkcijai L_c un tādējādi arī priekš Puasona modeļa (1.2.3), (1.2.4).

No vienādībām (1.2.10) un (1.2.11) izriet, ka

$$\hat{\lambda}_{j+1}\hat{\lambda}_{j+2}\dots\hat{\lambda}_n = \frac{\hat{D}_{n-j+1,n}}{\hat{D}_{n-j+1,j}} = \frac{1}{1 - \hat{p}_{j+1} - \hat{p}_{j+2} - \dots - \hat{p}_n}$$

un analogiski

$$\hat{\lambda}_j\hat{\lambda}_{j+1}\dots\hat{\lambda}_n = \frac{1}{1 - \hat{p}_j - \hat{p}_{j+1} - \dots - \hat{p}_n}.$$

Tātad

$$\hat{\lambda}_j\hat{\lambda}_{j+1}\dots\hat{\lambda}_n = \frac{1}{\frac{1}{\hat{\lambda}_{j+1}\hat{\lambda}_{j+2}\dots\hat{\lambda}_n} - \hat{p}_j}$$

un tad

$$\hat{\lambda}_j = \frac{1}{1 - \hat{p}_j\hat{\lambda}_{j+1}\hat{\lambda}_{j+2}\dots\hat{\lambda}_n}. \quad (1.2.15)$$

Ar atpakaļgaitas indukciju parādīsim, ka pēc formulas (1.2.15), izmantojot \hat{p}_j formā (1.2.14), iegūti novērtējumi $\hat{\lambda}_j$, ir tādi paši kā tie, kurus iegūst ar formulu (1.2.12).

Tas izpildās pie $j = n$, jo

$$\begin{aligned} \hat{\lambda}_n &= \frac{1}{1 - \hat{p}_n} && \text{no (1.2.15)} \\ &= \frac{1}{1 - \frac{C_{1n}}{D_{1n}}} && \text{no (1.2.14)} \\ &= \frac{D_{1n}}{D_{1n} - C_{1n}} = \frac{D_{1n}}{D_{1,n-1}}, \end{aligned}$$

kas sakrīt ar vienādojumu (1.2.12), kad $j = n$.

Tagad pieņemsim, ka šis rezultāts ir spēkā pie $j + 1, j + 2, \dots, n$. Tad vienādojumu (1.2.13) varam pārrakstīt formā

$$\hat{p}_j = \frac{C_{1j} + C_{2j} + \dots + C_{n-j+1,j}}{D_{1n} + D_{2,n-1}\hat{\lambda}_n + \dots + D_{n-j+1,j}\hat{\lambda}_{j+1}\dots\hat{\lambda}_n}$$

un ievietojot to vienādojumā (1.2.15) iegūstam

$$\hat{\lambda}_j = \frac{1}{1 - \frac{C_{1j} + C_{2j} + \dots + C_{n-j+1,j}}{D_{1n} + D_{2,n-1}\hat{\lambda}_n + \dots + D_{n-j+1,j}\hat{\lambda}_{j+1}\dots\hat{\lambda}_n}} \hat{\lambda}_{j+1}\hat{\lambda}_{j+2}\dots\hat{\lambda}_n. \quad (1.2.16)$$

Teorēma 2. $D_{1n} + D_{2,n-1}\hat{\lambda}_n + \dots + D_{n-j+1,j}\hat{\lambda}_{j+1}\dots\hat{\lambda}_n = \hat{\lambda}_{j+1}\dots\hat{\lambda}_n \sum_{i=1}^{n-j+1} D_{ij}$.

Pierādījums.

$$\begin{aligned} D_{1n} + D_{2,n-1}\hat{\lambda}_n &= D_{1n} + D_{2,n-1} \frac{D_{1n}}{D_{1,n-1}} \\ &= \frac{D_{1n}}{D_{1,n-1}} (D_{1,n-1} + D_{2,n-1}) \\ &= \hat{\lambda}_n (D_{1,n-1} + D_{2,n-1}). \end{aligned}$$

Tādēļ

$$\begin{aligned} D_{1n} + D_{2,n-1}\hat{\lambda}_n + D_{3,n-2}\hat{\lambda}_n\hat{\lambda}_{n-1} &= \hat{\lambda}_n (D_{1,n-1} + D_{2,n-1} + D_{3,n-2}\hat{\lambda}_{n-1}) \\ &= \hat{\lambda}_n \left((D_{1,n-2} + D_{2,n-2})\hat{\lambda}_{n-1} + D_{3,n-2}\hat{\lambda}_{n-1} \right), \end{aligned}$$

jo $\hat{\lambda}_{n-1} = (D_{1,n-1} + D_{2,n-1}) / (D_{1,n-2} + D_{2,n-2})$.

Tādējādi $D_{1n} + D_{2,n-1}\hat{\lambda}_n + D_{3,n-2}\hat{\lambda}_n\hat{\lambda}_{n-1} = \hat{\lambda}_{n-1}\hat{\lambda}_n (D_{1,n-2} + D_{2,n-2} + D_{3,n-2})$ utt.

Analogiski turpinot visiem locekļiem, teorēma tiks pierādīta. \square

Izmantojot 2. teorēmas rezultātu, (1.2.16) var pārveidot sekojoši

$$\begin{aligned} \hat{\lambda}_j &= \frac{1}{1 - \frac{C_{1j} + C_{2j} + \dots + C_{n-j+1,j}}{\hat{\lambda}_{j+1}\dots\hat{\lambda}_n \sum_{i=1}^{n-j+1} D_{ij}}} \hat{\lambda}_{j+1}\hat{\lambda}_{j+2}\dots\hat{\lambda}_n \\ &= \frac{\sum_{i=1}^{n-j+1} D_{ij}}{\sum_{i=1}^{n-j+1} D_{ij} - (C_{1j} + C_{2j} + \dots + C_{n-j+1,j})} \\ &= \frac{\sum_{i=1}^{n-j+1} D_{ij}}{\sum_{i=1}^{n-j+1} D_{i,j-1}}, \end{aligned}$$

kas sakrīt ar chain-ladder novērtējumu. Tādējādi ar indukcijas palīdzību tika pierādīts, ka Rozenberga metode ir ekvivalenta chain-ladder metodei.

Tātad mēs parādījām, ka chain-ladder metode dod tieši tos pašus novērtējumus kā modelis (1.2.3) un (1.2.4) pie pieņēmuma (1.2.5). Tātad mums ir dažādi veidi iegūt tos pašus novērtējumus $\{D_{in} : i = 2, 3, \dots, n\}$: chain-ladder metode, Rozenberga metode un vislielākās ticamības novērtējums vispārinātajā lineārajā modelī. Mēs varām pamatoti

saukt šo vispārināto lineāro modeli par chain-ladder lineāro modeli pie pieņēmuma (1.2.5). Taču, izmantosim iespēju atvieglot pieņēmumu par to, ka datiem ir Puasona sadalījums, kaut arī atstāsim to pašu modeļa formu un lielākās ticamības funkciju. Tādā veidā mēs iegūsim tos pašus rezultātus, bet izvairīsimies no datu sadalījuma specifiskācijas. Tas nozīmē, ka chain-ladder lineārais modelis var būt pielietots arī atlīdzību summām (ne tikai atlīdzību skaitam). Vēl vairāk, pie nosacījuma (1.2.5) modeli var pielietot negatīviem atlīdzību pieaugumiem.

Jāatzīmē, ka dažādu modeļu kolonu parametri arī ir saistīti. Chain-ladder metodes parametri λ_j , $j = 2, \dots, n$ ir rekursīvi saistīti ar nosacītās vislielākās ticamības parametriem p_j , $j = 1, \dots, n$ šādā veidā

$$\lambda_j \lambda_{j+1} \dots \lambda_n = 1 / (1 - p_j - p_{j+1} - \dots - p_n).$$

Vispārinātā lineārā modeļa kolonu parametri ir saistīti ar chain-ladder metodes parametriem un nosacītās vislielākās ticamības parametriem caur sakarībām [5]

$$\lambda_j = 1 + \frac{e^{\beta_j}}{\sum_{k=1}^{j-1} e^{\beta_k}} \quad \text{un} \quad p_j = \frac{e^{\beta_j}}{\sum_{k=1}^n e^{\beta_k}}.$$

Tā kā izmantojam vislielākās ticamības novērtējumu, jebkura modeļa parametru novērtējumi tiek iegūti tieši no lineārā modeļa novērtējumiem ar substitūciju. \hat{D}_{in} var iegūt gan no (1.2.6), gan no (1.2.10), jeb (1.2.11) – visos gadījumos tiks iegūti vienādi rezultāti.

Visa augstāk apskatītā galvenā ideja ir tā, ka tika atrasts vispārinātais lineārais modelis, kurš dod tieši tos pašus rezultātus kā chain-ladder metode, pie pieņēmuma (1.2.5).

Definīcija 3. Stohastisks chain-ladder modelis ir vispārinātais lineārais modelis, kurā atlīdzību pieaugumu summas tiek modelētas kā neatkarīgie mainīgie C_{ij} ar vidējo vērtību $\mu_{ij} = E(C_{ij})$, variācijas funkciju $V(\mu_{ij}) = \mu_{ij}$ un mēroga parametru $\phi > 0$, izmantojot logaritmisko saites funkciju, šādā formā

$$\eta_{ij} = \ln \mu_{ij} = c + \alpha_i + \beta_j.$$

Parametru novērtējumi $\hat{c}, \hat{\alpha}_i, \hat{\beta}_j$ tiek noteikti maksimizējot kvazi-ticamības funkciju un

prognozētās atlīdzību summas nākotnes trijstūrī tiek novērtētas ar

$$\hat{\mu}_{ij} = \exp(\hat{c} + \hat{\alpha}_i + \hat{\beta}_j).$$

Pie šīs modeļa definīcijas ir jāatzīmē, pirmkārt, ka šāda tipa modelī, tehniskajā nozīmē, ir pārāk daudz parametru un tāpēc ir nepieciešams uzdot divus papildus nosacījumus, parasti, $\alpha_1 = \beta_1 = 0$ un, otrkārt, nav nepieciešams iepriekš uzdot mēroga parametra ϕ vērtību, jo parametru novērtējumi un tādēļ arī prognozētās atlīdzību vērtības ir invariantas pret ϕ vērtību.

Gadījumā, kad ir doti daži negatīvi atlīdzību pieaugumi tiek rekomendēts [4] izmantot vispārinātos Pīrsona atlikumus, kuri atšķirībā no novirzes atlikumiem, ir vienādi definēti priekš negatīvām un pozitīvām C_{ij} vērtībām. Tas ir tāpēc, ka formāla integrālās izteiksmes (1.1.20) novērtēšana, uz kuras balstās novirzes atlikumi, izraisa $\ln C_{ij}$ aprēķināšanu, kuri nav definēti priekš negatīvām C_{ij} vērtībām. Tas viens pats netraucē izteiksmes (1.1.20) optimizācijai vislielākās kvazi-ticamības parametru novērtēšanas procedūrā, ar noteikumu, ka locekļi $\ln \mu_{ij}$ ir labi definēti, t.i., $\mu_{ij} > 0 \forall i, j$. Pateicoties lineāra prediktora $\eta_{ij} = c + \alpha_i + \beta_j$ specifiskai parametriskai struktūrai, šis nosacījums izpildās tiklīdz visas attīstības trijstūra atlīdzību pieaugumos kolonu (un rindu) summas ir pozitīvas. Tas saskaņojas ar pieņēmumu (1.2.5).

Ja modelis tiek definēts saskaņā ar 3. definīciju, tad nākotnes individuālās atlīdzību prognozes vidējā kvadrātiskā kļūda, kā parādīja Renshaw [6] ir

$$MSE(\hat{C}_{ij}) = E[(C_{ij} - \hat{C}_{ij})^2] \approx D[C_{ij}] + D[\hat{C}_{ij}].$$

No $\hat{C}_{ij} = \hat{\mu}_{ij} = \exp(\hat{\eta}_{ij})$ ar delta metodi iegūstam, ka

$$D[\hat{C}_{ij}] \approx \left| \frac{\partial \mu_{ij}}{\partial \eta_{ij}} \right|^2 D[\hat{\eta}_{ij}],$$

no 3. definīcijas arī zināms, ka

$$D[C_{ij}] = \phi \mu_{ij},$$

tātad

$$E[(C_{ij} - \hat{C}_{ij})^2] \approx \phi \hat{\mu}_{ij} + \hat{\mu}_{ij}^2 D[\hat{\eta}_{ij}]. \quad (1.2.17)$$

Renshaw arī [6] ir parādīja, ka notikuma perioda rezervju prognozes vidējā kvadrātiskā kļūda ir

$$MSE(\hat{R}_i) \approx \sum_j \phi \hat{\mu}_{ij} + \sum_j \hat{\mu}_{ij}^2 D[\hat{\eta}_{ij}] + 2 \sum_{\substack{j_1, j_2 \\ j_2 > j_1}} \hat{\mu}_{ij_1} \hat{\mu}_{ij_2} \text{Cov}[\hat{\eta}_{ij_1}, \hat{\eta}_{ij_2}] \quad (1.2.18)$$

un summāro rezervju prognozes vidējā kvadrātiskā kļūda ir

$$MSE(\hat{R}) \approx \sum_{i,j} \phi \hat{\mu}_{ij} + \sum_{i,j} \hat{\mu}_{ij}^2 D[\hat{\eta}_{ij}] + 2 \sum_{\substack{i_1, j_1 \\ i_2, j_2 \\ i_1, j_1 \neq i_2, j_2}} \hat{\mu}_{i_1 j_1} \hat{\mu}_{i_2 j_2} \text{Cov}[\hat{\eta}_{i_1 j_1}, \hat{\eta}_{i_2 j_2}]. \quad (1.2.19)$$

Summēšana visur notiek pa nākotnes trijstūra indeksiem j .

1.2.5. Chain-ladder modeļa Normālā aproksimācija

Gadījumā, kad negatīvo atlīdzību pieaugumu skaits ir pietiekoši liels, Puasona modeļa pieņēmums (1.2.5) neizpildās un šis modelis vairs nevar būt pielietots. Šajā gadījumā chain-ladder metodes novērtējumi var joprojām būt reproducēti izmantojot Normālo aproksimāciju. Izmantojot atlīdzību pieaugumu sadalījuma Normālo aproksimāciju, C_{ij} ir aptuveni Normāli sadalīti ar parametriem

$$E[D_{ij}] = \lambda_j D_{i,j-1} \quad \text{un} \quad D[D_{ij}] = \phi_j D_{i,j-1}, \quad (1.2.20)$$

kur $j \geq 2$ un pieņemts, ka $D_{i,j-1}$ ir zināmi. Apzīmējot $w_{ij} := D_{i,j-1}$ un izdalot ar to izteiksmes (1.2.20), iegūstam

$$E[D_{ij}/w_{ij}] = \lambda_j \quad \text{un} \quad D[D_{ij}/w_{ij}] = \phi_j/w_{ij}.$$

Modelis tiek analizēts koncentrējoties uz lielumiem $f_{ij} = D_{ij}/w_{ij}$, kuri saskaņā ar modeli, ir Normāli sadalīti ar mēroga parametriem ϕ_j un svariem w_{ij} . Atzīmēsim, ka f_{ij} veido individuālo attīstības faktoru trijstūri un ϕ_j ir atkarīgi no attīstības perioda j . Tas ir sarežģītums, jo parasti vispārinātajos lineārajos modeļos mēroga parametrs ir konstants visiem novērojumiem.

Šis modelis tiek novērtēts ar iteratīvo novērtēšanas metodi, tā saukto “kopējo modelēšanu” (joint modelling) [1, 357–365 lpp.], jo tas ietver parametru novērtēšanu gan vidējai

vērtībai, gan dispersijai, kas atšķiras no parastas tikai vidējās vērtības modelēšanas.

Kopējās modelēšanas process sākās ar pozitīvu patvaļīgu sākuma vērtību piešķiršanu lielumiem ϕ_j un vispārinātā lineārā modeļa pielāgošanu parametriem f_{ij} ar svariem $W_{ij} = w_{ij}/\phi_j$ (t.i., svari ir apgriezti proporcionāli dispersijai). Arī konstante parasti tiek iekļauta lineārajā prediktorā, tādējādi iegūstot

$$E[f_{ij}] = c + \alpha_{j-1} \quad \text{ar} \quad \alpha_1 = 0, \quad j \geq 2. \quad (1.2.21)$$

Jāatzīmē, ka lineārs prediktors ir atkarīgs tikai no attīstības perioda j un pielāgotās vērtības \hat{f}_{ij} ir attīstības faktoru λ_j novērtējumi. Iegūstot pielāgotās vērtības, tiek aprēķināti atlikumu kvadrāti

$$r_{ij}^2 = w_{ij}(f_{ij} - \hat{f}_{ij})^2.$$

Šie atlikumu kvadrāti tiek izmantoti, kā rezultējošais mainīgais otrajā vispārinātajā lineārajā modelī, kurā prediktors ir atkarīgs tikai no attīstības perioda j , t.i.,

$$E[r_{ij}^2] = c + \alpha_{j-1} \quad \text{ar} \quad \alpha_1 = 0, \quad j \geq 2. \quad (1.2.22)$$

Pielāgotās vērtības \hat{r}_{ij}^2 no otrā modeļa tiek izmantoti, lai precizētu vērtības ϕ_j un pirmais modelis tiek atkārtoti pielāgots pēc W_{ij} precizēšanas, lai atspoguļotu revidētus ϕ_j novērtējumus. Tas arī pabeidz kopējās modelēšanas procesu, kura rezultātā λ_j un ϕ_j novērtējumi tiek iegūti.

Piezīme 3. *Tā kā atlikumu kvadrātu modelis ir atkarīgs tikai no attīstības perioda j , pielāgotās vērtības \hat{r}_{ij}^2 būs novērtētas vienkārši kā attīstības perioda j atlikumu kvadrātu vidējais pa i . Atlikumu formu nosaka kļūdu sadalījums, kurš tiek lietots modelī priekš vidējās vērtības, kurš šajā gadījumā ir Normālais sadalījums.*

Prognozes kļūda šim modelim tiek aprēķināta pēc formulas (1.2.1), savukārt procesa un novērtējuma dispersijas, $D[D_{in}|T_p]$ un $D[\hat{D}_{in}|T_p]$, tiek aprēķinātas rekursīvi. Sākumā izvedīsim rekursīvo formulu procesa kļūdas $D[D_{in}|T_p]$ aprēķināšanai. Lai atvieglotu apzīmējumus, apskatīsim atsevišķu rindu, atmetot indeksu i .

No modeļa specifikācijas $E[D_j|C_{j-1}] = \lambda_j D_{j-1}$. Izmantojot šo faktu divu soļu uz

priekšu nosacītā matemātiskā cerība ir

$$\begin{aligned} E[D_{j+1}|C_1, \dots, C_{j-1}] &= E[E[D_{j+1}|C_1, \dots, C_{j-1}, C_j]|C_1, \dots, C_{j-1}] \\ &= E[\lambda_j D_j|C_1, \dots, C_{j-1}] = \lambda_{j+1} \lambda_j D_{j-1}. \end{aligned}$$

Analoģiski k soļu uz priekšu nosacītā matemātiskā cerība ir

$$E[D_{j+k}|C_1, \dots, C_{j-1}] = \lambda_{j+k} \lambda_{j+k-1} \dots \lambda_j D_{j-1}.$$

Tagad apskatīsim dispersiju. No modeļa specififikācijas $D[D_j|C_1, \dots, C_{j-1}] = \phi_j D_{j-1}$. Tad divu soļu uz priekšu nosacītā dispersija ir

$$\begin{aligned} D[D_{j+1}|C_1, \dots, C_{j-1}] &= E[D[D_{j+1}|C_1, \dots, C_j]|C_1, \dots, C_{j-1}] + \lambda_{j+1}^2 D[D_j|C_1, \dots, C_{j-1}] \\ &= E[\phi_{j+1} D_j|C_1, \dots, C_{j-1}] + \lambda_{j+1}^2 \phi_j D_{j-1} \\ &= (\phi_{j+1} \lambda_j + \lambda_{j+1}^2 \phi_j) D_{j-1}. \end{aligned}$$

Analoģiski triju soļu uz priekšu nosacītā dispersija ir

$$\begin{aligned} D[D_{j+2}|C_1, \dots, C_{j-1}] &= E[D[D_{j+2}|C_1, \dots, C_j]|C_1, \dots, C_{j-1}] + \lambda_{j+2}^2 \lambda_{j+1}^2 D[D_j|C_1, \dots, C_{j-1}] \\ &= E[(\phi_{j+2} \lambda_{j+1} + \phi_{j+1} \lambda_{j+2}^2) D_j|C_1, \dots, C_{j-1}] + \lambda_{j+2}^2 \lambda_{j+1}^2 \phi_j D_{j-1} \\ &= (\phi_{j+2} \lambda_{j+1} \lambda_j + \lambda_{j+2}^2 \phi_{j+1} \lambda_j + \lambda_{j+2}^2 \lambda_{j+1}^2 \phi_j) D_{j-1}. \end{aligned}$$

Tā turpinot, iegūstam $k + 1$ soļu uz priekšu nosacītu dispersiju

$$\begin{aligned} D[D_{j+k}|C_1, \dots, C_{j-1}] &= E[D[D_{j+k}|C_1, \dots, C_j]|C_1, \dots, C_{j-1}] + \lambda_{j+k}^2 \dots \lambda_{j+1}^2 D[D_j|C_1, \dots, C_{j-1}] \\ &= (\phi_{j+k} \lambda_{j+k-1} \dots \lambda_j + \lambda_{j+k}^2 \phi_{j+k-1} \lambda_{j+k-2} \dots \lambda_j \\ &\quad + \lambda_{j+k}^2 \lambda_{j+k-1}^2 \phi_{j+k-2} \dots \lambda_j + \dots + \lambda_{j+k}^2 \dots \lambda_{j+2}^2 \phi_{j+1} \lambda_j \\ &\quad + \lambda_{j+k}^2 \dots \lambda_{j+1}^2 \phi_j) D_{j-1}. \end{aligned}$$

Tagad izvedīsim formulu novērtējuma kļūdas aprēķināšanai. Apskatot atsevišķu rindu un atmetot indeksu i , mums interesē lielums $D[\hat{D}_n|D_{n-i+1}]$, kas ir notikuma perioda prognozētā summāras atlīdzības novērtējuma dispersija, zinot vēlākās kumulētas atlīdzības vērtību D_{n-i+1} . Atzīmēsim, ka to pašu rezultātu var iegūt apskatot dispersiju no

prognozētu atlīdzību pieaugumu summas.

$$D[\hat{D}_n|D_{n-i+1}] = D[\hat{\lambda}_{n-i+2}\dots\hat{\lambda}_n D_{n-i+1}|D_{n-i+1}] = D_{n-i+1}^2 D[\hat{\lambda}_{n-i+2}\dots\hat{\lambda}_n|D_{n-i+1}].$$

No šīs formulas redzams, ka mums ir nepieciešams atrast attīstības faktoru novērtējumu reizinājuma dispersiju, kura var būt iegūta rekursīvi, reizinātu ar vēlāko kumulēto atlīdzību kvadrātā. Turpmāka izveduma pārskatāmībai, formulu pierakstā izlaidīsim nosacījumu.

Izvedumā izmantosim sekojošu faktu. Ja divi gadījuma lielumi X un Y ir neatkarīgi, tad ir spēkā sakarība

$$\begin{aligned} D[XY] &= EX^2EY^2 - (EX)^2(EY)^2 \\ &= (EX)^2EY^2 - (EX)^2(EY)^2 + (EY)^2EX^2 - (EY)^2(EX)^2 \\ &\quad + EY^2EX^2 - EY^2(EX)^2 - (EY)^2EX^2 + (EY)^2(EX)^2 \\ &= (EX)^2(EY^2 - (EY)^2) + (EY)^2(EX^2 - (EX)^2) \\ &\quad + (EY^2 - (EY)^2)(EX^2 - (EX)^2) \\ &= (EX)^2DY + (EY)^2DX + DYDX. \end{aligned}$$

Notikuma periodam $i = 2$, novērtējuma dispersija ir vienkārši $D_{n-i+1}^2 D[\hat{\lambda}_n]$. Notikuma periodam $i = 3$, mums interesē $D_{n-i+1}^2 D[\hat{\lambda}_{n-1}\hat{\lambda}_n]$, kur pie nosacījuma, ka attīstības faktori ir neatkarīgi vai, kā parādīts [7], vismaz nekorelēti,

$$D[\hat{\lambda}_{n-1}\hat{\lambda}_n] = \left(E[\hat{\lambda}_{n-1}]\right)^2 D[\hat{\lambda}_n] + \left(E[\hat{\lambda}_n]\right)^2 D[\hat{\lambda}_{n-1}] + D[\hat{\lambda}_n]D[\hat{\lambda}_{n-1}],$$

kur attīstības faktoru dispersijas tiek iegūtas modeļa pielāgošanas rezultātā, bet matemātiskās cerības tiek aizvietotas ar novērotām vērtībām, tādā veidā

$$D[\hat{\lambda}_{n-1}\hat{\lambda}_n] \approx \left(\hat{\lambda}_{n-1}\right)^2 D[\hat{\lambda}_n] + \left(\hat{\lambda}_n\right)^2 D[\hat{\lambda}_{n-1}] + D[\hat{\lambda}_n]D[\hat{\lambda}_{n-1}].$$

Nākamajam notikuma periodam $i = 4$ ir vajadzīga triju faktoru reizinājuma dispersija

$$D[\hat{\lambda}_{n-2}(\hat{\lambda}_{n-1}\hat{\lambda}_n)] = \left(E[\hat{\lambda}_{n-2}]\right)^2 D[\hat{\lambda}_{n-1}\hat{\lambda}_n] + \left(E[\hat{\lambda}_{n-1}\hat{\lambda}_n]\right)^2 D[\hat{\lambda}_{n-2}] + D[\hat{\lambda}_{n-1}\hat{\lambda}_n]D[\hat{\lambda}_{n-2}].$$

Ievietojot šajā izteiksmē iepriekšējā soli aprēķinātu vērtību $D[\hat{\lambda}_{n-1}\hat{\lambda}_n]$, iegūstam nepie-

ciešamo dispersiju. Tādā veidā šī rekursīva procedūra tiek turpināta līdz $i = n$, kur

$$D[\hat{\lambda}_{n-i+2}\dots\hat{\lambda}_n] = \left(E[\hat{\lambda}_{n-i+2}]\right)^2 D[\hat{\lambda}_{n-i+3}\dots\hat{\lambda}_n] + \left(E[\hat{\lambda}_{n-i+3}\dots\hat{\lambda}_n]\right)^2 D[\hat{\lambda}_{n-i+2}] \\ + D[\hat{\lambda}_{n-i+3}\dots\hat{\lambda}_n]D[\hat{\lambda}_{n-i+2}]$$

tiek iegūta ievietojot iepriekšējā solī $i = n - 1$ aprēķinātu vērtību.

Summāro rezervju prognozes kļūda var būt aprēķināta kā summa

$$MSE(\hat{R}) = D[R] + D[\hat{R}],$$

kur summāro rezervju procesa dispersija ir individuālo notikuma periodu procesu dispersiju summa, t.i.,

$$D[R] = \sum_{i=2}^n D[R_i] = \sum_{i=2}^n D[D_{in}],$$

bet summāro rezervju novērtējuma kļūda ir dota ar

$$D[\hat{R}] \approx \sum_{i=2}^n D[\hat{D}_{in}] + 2 \sum_{\substack{i=2 \\ j>i}}^n \text{Cov}[\hat{D}_{in}, \hat{D}_{jn}].$$

Summāro rezervju novērtēšanas kļūdas aprēķins prasa kovariācijas $\text{Cov}[\hat{D}_{in}, \hat{D}_{jn}]$ aprēķinu. Savukārt šīs kovariācijas var būt iegūtas izmantojot iepriekš aprēķinātās attīstības faktoru reizinājuma dispersijas. Apskatīsim notikuma periodu $i = 2$, tad

$$\text{Cov}[\hat{D}_{2,n}, \hat{D}_{3,n}] = \text{Cov}[D_{2,n-1}\hat{\lambda}_n, D_{3,n-2}\hat{\lambda}_{n-1}\hat{\lambda}_n] \\ = D_{2,n-1}D_{3,n-2}\hat{\lambda}_{n-1}D[\hat{\lambda}_n],$$

jo attīstības faktori ir neatkarīgi. Līdzīgi

$$\text{Cov}[\hat{D}_{2,n}, \hat{D}_{4,n}] = \text{Cov}[D_{2,n-1}\hat{\lambda}_n, D_{4,n-3}\hat{\lambda}_{n-2}\hat{\lambda}_{n-1}\hat{\lambda}_n] \\ = D_{2,n-1}D_{4,n-3}\hat{\lambda}_{n-2}\hat{\lambda}_{n-1}D[\hat{\lambda}_n],$$

$$\text{Cov}[\hat{D}_{2,n}, \hat{D}_{5,n}] = D_{2,n-1}D_{5,n-4}\hat{\lambda}_{n-3}\hat{\lambda}_{n-2}\hat{\lambda}_{n-1}D[\hat{\lambda}_n]$$

un tā tālāk, apskatot visu notikuma periodu lielāku par 2 kovariācijas ar otro notikuma

periodu. Apskatīsim $i = 3$

$$\begin{aligned}\text{Cov}[\hat{D}_{3,n}, \hat{D}_{4,n}] &= \text{Cov}[D_{3,n-2}\hat{\lambda}_{n-1}\hat{\lambda}_n, D_{4,n-3}\hat{\lambda}_{n-2}\hat{\lambda}_{n-1}\hat{\lambda}_n] \\ &= D_{3,n-2}D_{4,n-3}\hat{\lambda}_{n-2}D[\hat{\lambda}_{n-1}\hat{\lambda}_n].\end{aligned}$$

Šeit $D[\hat{\lambda}_{n-1}\hat{\lambda}_n]$ vietā ievietojām vērtību, kura bija izrēķinātā agrāk un analogiski turpinot iegūstam kovariācijas visiem $j > 3$ ar $i = 3$. Šī procedūra tiek turpināta līdz notikuma periodam $i = n - 1$, kur

$$\begin{aligned}\text{Cov}[\hat{D}_{n-1,n}, \hat{D}_{n,n}] &= \text{Cov}[D_{n-1,2}\hat{\lambda}_3\hat{\lambda}_4\dots\hat{\lambda}_n, D_{n,1}\hat{\lambda}_2\hat{\lambda}_3\dots\hat{\lambda}_n] \\ &= D_{n-1,2}D_{n,1}\hat{\lambda}_2D[\hat{\lambda}_3\hat{\lambda}_4\dots\hat{\lambda}_n].\end{aligned}$$

1.2.6. Maka modelis

Viens no pirmajiem stohastiskajiem modeļiem, kurš dod tos pašus novērtējumus kā chain-ladder metode ir Maka modelis (“Mack’s model”). Šim modelim ir ierobežoti pieņēmumi attiecībā uz datu sadalījumu, norādot tikai pirmos divus momentus. Tāpēc šā modeļa priekšrocība ir vienkāršība, bet tai pat laikā tas ierobežo atlīdzību rezervju sadalījuma analīzi tikai ar pirmo divu momentu noteikšanu.

Saskaņā ar Maka modeli [8] D_{ij} vidējā vērtība un dispersija ir

$$E[D_{ij}] = \lambda_j D_{i,j-1} \quad \text{un} \quad D[D_{ij}] = \sigma_j^2 D_{i,j-1}. \quad (1.2.23)$$

Maks piedāvāja sekojošus novērtējumus parametriem λ_j un σ_j^2

$$\hat{\lambda}_j = \frac{\sum_{i=1}^{n-j+1} w_{ij} f_{ij}}{\sum_{i=1}^{n-j+1} w_{ij}}, \quad \text{kur} \quad w_{ij} = D_{i,j-1} \quad \text{un} \quad f_{ij} = \frac{D_{ij}}{D_{i,j-1}} \quad (1.2.24)$$

un

$$\hat{\sigma}_j^2 = \frac{1}{n-j} \sum_{i=1}^{n-j+1} w_{ij} (f_{ij} - \hat{\lambda}_j)^2. \quad (1.2.25)$$

Attīstības faktoru λ_j novērtējums ir individuālo attīstības faktoru svērtais vidējais [7]. Maks pierāda, ka tas dod nenovirzītus un efektīvus attīstības faktoru novērtējumus. Dispersijas komponente σ_j^2 tiek novērtēta kā svērto atlikumu vidējais, kur dalītājs ir atlikumu skaits (ko izmantoja novērtējuma aprēķinā) mīnus viens. Viens tiek atņemts, lai nodrošinātu σ_j^2 nenovirzītu novērtējumu. Šis dispersijas komponentes neizmanto novērtējot at-

tīstības faktorus, bet tās ir nepieciešamas apskatot nākotnes atlīdzību prognozes kļūdas. Līdzīgi, ka Normālajā modelī, nepietiek informācijas pēdējās dispersijas komponentes σ_n^2 novērtēšanai.

Atlīdzību rezervju prognozes kļūdu Maka modeļa gadījumā var aprēķināt izmantojot sekojošu teorēmu, ko pierādīja Maks [8] 1993.gadā un tās sekas.

Teorēma 4. *Pieņemsim, ka kumulēto atlīdzību vidējā vērtība un dispersija ir dotas ar (1.2.23) un dažādu notikuma perioda kumulētas atlīdzības ir neatkarīgas, t.i., $\{D_{i1}, \dots, D_{in}\}$, $\{D_{k1}, \dots, D_{kn}\}$, ja $i \neq k$ ir neatkarīgi, tad notikuma perioda rezervju prognozes vidējās kvadrātiskās kļūdas novērtējums ir*

$$\widehat{MSE}(\hat{R}_i) = \hat{D}_{in}^2 \sum_{k=n-i+1}^{n-1} \frac{\hat{\sigma}_{k+1}^2}{\hat{\lambda}_{k+1}^2} \left(\frac{1}{\hat{D}_{ik}} + \frac{1}{\sum_{q=1}^{n-k} D_{qk}} \right).$$

Sekas 5. *Pie 4. teorēmas pieņēmumiem summāro rezervju prognozes $\hat{R} = \hat{R}_2 + \dots + \hat{R}_n$ vidējās kvadrātiskās kļūdas novērtējums ir*

$$\widehat{MSE}(\hat{R}) = \sum_{i=2}^n \left[MSE(\hat{R}_i) + \hat{D}_{in} \left(\sum_{q=i+1}^n \hat{D}_{qn} \right) \sum_{k=n-i+1}^{n-1} \frac{2\hat{\sigma}_{k+1}^2}{\hat{\lambda}_{k+1}^2 \sum_{q=1}^{n-k} D_{qk}} \right]. \quad (1.2.26)$$

1.2.7. Log-Normālais modelis

Apskatot log-Normālo modeli ir jāsāk ar atlīdzību pieaugumu transformēšanu, proti logaritmēšanu. Tad transformētajām vērtībām modelis tiek pielāgots izmantojot parasto regresijas analīzes mazāko kvadrātu metodi. Tad, pārņemot chain-ladder tipa prediktora struktūru,

$$Y_{ij} = \ln C_{ij} = \eta_{ij} + \epsilon_{ij},$$

$$\text{kur } \eta_{ij} = c + \alpha_i + \beta_j \quad \text{un} \quad \epsilon_{ij} \sim N(0, \sigma^2). \quad (1.2.27)$$

Novērtējot lineāra prediktora parametrus un procesa dispersiju σ^2 , pielāgotās vērtības uz logaritmiskās skalas tiek iegūtas veidojot atbilstošas parametru novērtējumu summas. Taču, lai iegūtu vidējās vērtības novērtējumu uz oriģinālās skalas, mēs nevaram vienkārši

kāpināt lineāru prediktoru, jo tas dotu mediānas novērtējumu. Pielāgotās vērtības uz oriģinālās skalas tiek aprēķinātas sekojoši

$$\hat{C}_{ij} = \exp(\hat{\eta}_{ij} + 1/2\hat{\sigma}_{ij}^2), \quad \text{kur} \quad \hat{\sigma}_{ij}^2 = D[\hat{\eta}_{ij}] + \hat{\sigma}^2. \quad (1.2.28)$$

Komponentes σ_{ij}^2 faktiski ir lineāra prediktora prognozes kļūdas un tās tiek aprēķinātas kā lineāra prediktora dispersijas un procesa dispersijas summa.

Piezīme 6. *Iemesls tam, ka dispersijas komponente vienādojumā (1.2.28) log-Normālajā modelī ir vajadzīga ir tas, ka Puasona modelī paši atlīdzību pieaugumi tiek izmantoti kā rezultējošs mainīgais, bet log-Normālajā modelī - atlīdzību pieaugumu logaritmi.*

Vienādība (1.2.28) ir log-Normālā sadalījuma matemātiskās cerības standarta veids. Jāatzīmē, ka dispersijas komponente iekļauj sevī gan procesa, gan novērtējuma kļūdu. Nākotnes atlīdzību pieaugumu \hat{C}_{ij} prognozes dispersija ir

$$MSE(\hat{C}_{ij}) = \hat{C}_{ij}^2 (\exp(\hat{\sigma}_{ij}^2) - 1),$$

kas ir log-Normālā sadalījuma dispersijas standarta veids.

Lai aprēķinātu notikuma perioda un summāro rezervju prognozes kļūdu, nepieciešams apskatīt prognozēto vērtību summas dispersiju, ņemot vērā visas kovariācijas starp prognozētajām vērtībām. Notikuma perioda rezervju $\hat{R}_i = \sum_{j=n-i+2}^n \hat{C}_{ij}$ prognozes kļūda tiek aprēķināta sekojoši

$$MSE(\hat{R}_i) \approx \sum_{j=n-i+2}^n MSE(\hat{C}_{ij}) + 2 \sum_{\substack{j_1, j_2=n-i+2 \\ j_2 > j_1}}^n \hat{C}_{ij_1} \hat{C}_{ij_2} (\exp(\text{Cov}[\hat{\eta}_{ij_1}, \hat{\eta}_{ij_2}]) - 1) \quad (1.2.29)$$

un summāro rezervju prognozes kļūda tiek aprēķināta pēc formulas

$$MSE(\hat{R}) \approx \sum_{i,j} MSE(\hat{C}_{ij}) + 2 \sum_{\substack{i_1, j_1 \\ i_2, j_2 \\ i_1, j_1 \neq i_2, j_2}} \hat{C}_{i_1 j_1} \hat{C}_{i_2 j_2} (\exp(\text{Cov}[\hat{\eta}_{i_1 j_1}, \hat{\eta}_{i_2 j_2}]) - 1), \quad (1.2.30)$$

kur summēšana notiek pa prognozēto nākotnes atlīdzību indeksiem.

Negatīvi atlīdzību pieaugumi uzreiz izraisa problēmas šajā modelī, tā kā logaritms nav definēts negatīvām vērtībām.

1.2.8. Gamma modelis

Maks [9] ierosināja izmantot tādu pašu lineāro prediktoru kā 3. definīcijā, bet piedāvāja izmantot Gamma sadalījumu atlīdzību summām. Tādējādi tiek analizēts tāds pats modelis kā aprakstīts 1.2.4.. apakšnodaļā, tikai dispersija formā $D[C_{ij}] = \phi\mu_{ij}$ tiek aizvietota ar $D[C_{ij}] = \phi\mu_{ij}^2$, t.i. Gamma modeļa dispersija ir proporcionāla vidējās vērtības kvadrātam nevis pašai vidējai vērtībai. Ar šo modeli prognozētās vērtības parasti, bet ne vienmēr, ir tuvas chain-ladder novērtējumiem.

Pārņemot chain-ladder tipa prediktora struktūru $\ln \mu_{ij} = \eta_{ij} = c + \alpha_i + \beta_j$, parametru novērtējumi un prognozētās vērtības tiek iegūtas tiešā veidā izmantojot vispārināto lineāro modeli. Arī acīmredzami, ka nākotnes atlīdzību rezervju prognozes kļūdas tiek novērtētas analogiski Puasona modeļa gadījumam, izdarot attiecīgas izmaiņas procesa kļūdā izteiksmēs (1.2.17), (1.2.18) un (1.2.19). Tātad

$$MSE(\hat{C}_{ij}) \approx \phi\hat{\mu}_{ij}^2 + \hat{\mu}_{ij}^2 D[\hat{\eta}_{ij}], \quad (1.2.31)$$

$$MSE(\hat{R}_i) \approx \sum_j \phi\hat{\mu}_{ij}^2 + \sum_j \hat{\mu}_{ij}^2 D[\hat{\eta}_{ij}] + 2 \sum_{\substack{j_1, j_2 \\ j_2 > j_1}} \hat{\mu}_{ij_1} \hat{\mu}_{ij_2} \text{Cov}[\hat{\eta}_{ij_1}, \hat{\eta}_{ij_2}], \quad (1.2.32)$$

$$MSE(\hat{R}) \approx \sum_{i,j} \phi\hat{\mu}_{ij}^2 + \sum_{i,j} \hat{\mu}_{ij}^2 D[\hat{\eta}_{ij}] + 2 \sum_{\substack{i_1, j_1 \\ i_2, j_2 \\ i_1, j_1 \neq i_2, j_2}} \hat{\mu}_{i_1 j_1} \hat{\mu}_{i_2 j_2} \text{Cov}[\hat{\eta}_{i_1 j_1}, \hat{\eta}_{i_2 j_2}], \quad (1.2.33)$$

kur summēšana notiek pa nākotnes trijstūra indeksiem j .

2. Butstrapa metode

2.1. Butstrapa metodes jēdziens

Butstraps ir datu pārkārtošanas metode statistisko funkcionāļu $T(Y_1, Y_2, \dots, Y_n, F)$ sadalījuma aproksimācijai, kur Y_1, \dots, Y_n ir izlases novērojumi un F ir sadalījuma funkcija.

Pieņemsim, ka Y_1, Y_2, \dots, Y_n ir neatkarīgi un vienādi sadalīti ar $Y_1 \sim F$ un $T = T(Y_1, \dots, Y_n, F)$ ir kāds funkcionālis. Viena no galvenajām matemātiskās statistikas problēmām ir atrast T sadalījumu, tas ir atrast $P_F(T(Y_1, \dots, Y_n, F) \leq t)$. Pieņemsim, ka Y_1, \dots, Y_n pietiekami labi reprezentē īsto populācijas sadalījumu. Butstrapa ideja ir ģenerēt daudz izlašu no sākotnējās un aproksimēt statistikas T sadalījumu. Tas ir, izvēlēsimies no dotās izlases Y_1, \dots, Y_n jaunas izlases ar atkārtojumiem no empīriskās sadalījuma funkcijas \hat{F}_n ,

$$\hat{F}_n(y) = 1/n \sum_{i=1}^n I_{\{Y_i \leq y\}}, \quad \text{kur} \quad I_{\{Y_i \leq y\}} = \begin{cases} 1, & \text{ja } Y_i \leq y \\ 0, & \text{ja } Y_i > y \end{cases}.$$

Līdz ar to visiem novērojumiem ir vienāda $1/n$ varbūtība tikt izvēlētiem (tāpēc, ka izvēlamies faktiski no novērtētā populācijas sadalījuma \hat{F}_n). To arī sauc par empīrisko, neparametrisko vai dažkārt par parasto butstrapu.

Apzīmēsim iegūtās butstrapa izlases $\{Y_{11}^*, \dots, Y_{1n}^*\}, \{Y_{21}^*, \dots, Y_{2n}^*\}, \dots, \{Y_{B1}^*, \dots, Y_{Bn}^*\}$, kur B apzīmē butstrapoto izlašu skaitu. Statistiskā funkcionāļa T sadalījumu punktā t , tas ir, $P_F(T \leq t)$ aproksimēsim ar $\{j \text{ skaits} : T_j^* \leq t\}/B$, kur $T_1^*, T_2^*, \dots, T_B^*$ apzīmē statistikas T vērtības B dažādajām butstrapa izlasēm.

2.2. Rezidiju butstraps

Standartā butstrapa pielietojumā [10], [11], kad pieņemts, ka dati ir neatkarīgi un vienādi sadalīti, datu pārkārtošanu var veikt tieši no pašiem datiem. Regresijas tipa

problēmu gadījumā, parasti tiek pieņemts ka dati ir neatkarīgi, bet nav vienādi sadalīti, jo vidējie (un iespējams dispersijas) ir atkarīgas no prediktoriem. Tādēļ regresijas tipa problēmām parasti tiek izmantots rezidiju butstraps [12], nevis pāru butstraps, jo atlikumi parasti ir neatkarīgi un vienādi sadalīti, vai var tikt uztaisīti par tādiem.

Pieņemsim, ka ir regresija Y uz X . X ir $n \times p$ dizaina matrica ar i -to rindu vienādu ar x_i^T , β ir parametru vektors un $\hat{\beta}$ ir parametru novērtējumi, kuri iegūti apskatāma regresijas modeļa pielāgošanas rezultātā.

Definīcija 4. Pieņemsim, ka r_1, r_2, \dots, r_n ir atlikumi iegūti modeļa pielāgošanas rezultātā; $\bar{r} = 0$, ja $x_i = (1, x_{i1}, \dots, x_{i,p-1})^T$, bet ne otrādi. Definēsim $\tilde{r}_i = r_i - \bar{r}$ un, lai r_1^*, \dots, r_n^* ir izlase ar atkārtojumiem ar apjomu n no $\{\tilde{r}_1, \dots, \tilde{r}_n\}$. Pie tam, lai β^* ir parametra β novērtējums, kas aprēķināts no (x_i, y_i^*) , $i = 1, \dots, n$. Tā ir $\hat{\beta}$ butstrapotā versija un šī shēma tiek saukta par rezidiju butstrapu.

2.3. Rezidiju butstrapa metodes izmantošana IBNR rezervju noteikšanai

Lai realizētu rezidiju butstrapa metodi ir nepieciešams izvēlēties modeli, definēt atbilstošus atlikumus un veikt butstrapa prognozēšanas procedūru.

Lai definētu piemērotākus atlikumus ir jāņem vērā divas lietas. Pirmkārt, datu pārkārtošanas metode balstās uz tā, ka atlikumi ir neatkarīgi un vienādi sadalīti, un, otrkārt, ir vienalga pārkārtot atlikumus vai atlikumus reizinātus ar konstanti, kamēr tas tiek ņemts vērā ģenerējot pseido-datus.

Vispārinātā lineārā modeļa ietvaros varētu izmantot dažādus atlikumu tipus (Pīrsona, novirzes, utt.). Puasona modeļa gadījumā butstrapam tiek izmantoti Pīrsona atlikumi, kuri saskaņā ar 1. definīciju ir šādi

$$r_{P(ij)} = \frac{C_{ij} - \hat{\mu}_{ij}}{\sqrt{\hat{\mu}_{ij}}},$$

kur $\hat{\mu}_{ij}$ ir pielāgotās atlīdzību pieaugumu vērtības dotas ar vienādību (1.2.4).

Apskatīsim butstrapa procedūru, kura ļauj iegūt rezervju vidējās vērtības novērtējumu, novērtējuma dispersiju un rezervju novērtējumu sadalījumu.

Pirmais variants [10] izmanto Centrālās robežteorēmas priekšrocības un pēc būtības

aprosimē rezervju sadalījumu ar Normālo sadalījumu, kur kā matemātiskā cerība ir sākotnējā prognoze (uz sākotnējiem datiem) un standartnovirze tiek dota ar prognozes standartklūdu. Galvenā šis pieejas atšķirība no teorētiskās standartklūdu aproksimācijas (ar vispārināto lineāro modeli) ir tas, ka novērtējuma dispersija tiek novērtēta ar butstrapa palīdzību nevis izmantojot (aptuvenas) teorētiskās izteiksmes. Šajā gadījumā butstrapa prognozes klūda ir

$$RMSE_b(R_i) = \sqrt{\hat{\phi}\hat{R}_i + \frac{n}{0.5n(n+1) - 2n + 1} D_b[\hat{R}_i]}, \quad (2.3.1)$$

kur $\hat{\phi}$ un \hat{R}_i ir vislielākās kvazi-ticamības parametru novērtējumi, bet $D_b[\hat{R}_i]$ ir parametra novērtējuma \hat{D}_{in} dispersijas butstrapa novērtējums, t.i.,

$$D_b[\hat{R}_i] = 1/B \sum_{k=1}^B (R_k^* - \hat{R}_i)^2, \quad (2.3.2)$$

kur B ir butstrapa cilpas atkārtojumu skaits un R_k^* ir R_i butstrapa novērtējums k -jā atkārtojumā. Summāro rezervju prognozes klūda tiek iegūta pēc tas pašas formulas, tikai R_i tiek aizvietota ar R .

3. Empīriskie rezultāti

Šajā darba daļā pielietosim teorētiski apskatītas metodes uz konkrētiem datiem par apdrošināšanas atlīdzību summas attīstību 15 periodos (1. pielikums, dati tiek doti naudas vienību tūkstošos), ar mērķi prognozēt zaudējumu atlīdzību rezervju summu un šīs prognozes vidējo kvadrātisko kļūdu, kā arī iegūt prognozes sadalījumu. Visi nepieciešamie aprēķini un parametru novērtējumi tika veikti ar programmas R palīdzību, A. pielikumā ievietoti visi R programmu kodi.

3.1. Chain-ladder metode

Ar chain-ladder metodi iegūtais rezervju summas novērtējums ir dots 3.1. tabulā.

2. pielikumā ir parādīts rezervju aprēķina izvērsums. Tabulā tiek dotas kumulētas atlīdzības D_{ij} , pie $i + j \leq 14$, kā arī nākotnes prognozētas kumulētas atlīdzības \hat{D}_{ij} , pie

3.1. tabula Rezervju summas novērtējums ar chain-ladder metodi

Notikuma periods	Rezerves
1	0
2	227
3	1012
4	1731
5	2049
6	3167
7	5757
8	8820
9	12323
10	16405
11	17508
12	27131
13	39937
14	58621
15	92199
Summa	286887

$i + j > 14$ un attīstības faktori $\lambda_2, \dots, \lambda_{15}$, ko iegūstam pielietojot formulu (1.2.2).

3.2. Vispārinātais lineārais modelis

Lai iegūtu nākotnes atlīdzību summas, jeb rezervju prognozi ar vispārināto lineāro modeli ir jāveic sekojoša procedūra:

- definēt modeli;
- novērtēt parametrus c, α_i, β_j visiem $i, j = 1, 2, \dots, n$ un ϕ ;
- iegūt pielāgotās vērtības $\hat{\mu}_{ij}$ ($i = 1, \dots, n$ un $j = 1, \dots, n - i + 1$);
- pārbaudīt modeli;
- iegūt individuālās prognozes $\hat{C}_{ij} = \hat{\mu}_{ij} = \hat{c} + \hat{\alpha}_i + \hat{\beta}_j$ ($i = 2, \dots, n$ un $j = n - i + 2, \dots, n$);
- iegūt katra notikuma perioda summāro rezervju prognozi $\hat{R}_i = \sum_{j=n-i+2}^n \hat{\mu}_{ij}$ ($i = 2, \dots, n$);
- iegūt summāro rezervju prognozi $\hat{R} = \sum_{i=2}^n \hat{R}_i$.

3.2.1. Virs-izklidēts Puasona modelis

Pielāgosim datiem virs-izklidēto Puasona modeli, definēto kā 3. definīcijā. 3.2. tabulā tiek parādīti modeļa parametru novērtējumi un to standartklūdas, kas iegūtas ar vislielākās kvazi-ticamības metodi programmā R. Visi parametru novērtējumi ir būtiski 95% ticamības līmenī, izņemot α_2 . Bet tas neliecina par to, ka šim parametram jābūt nulle, jo tā izdarot, mēs varām neievērot trendu, kas pats par sevi var būt statistiski nozīmīgs.

Ar šo metodi iegūtais rezervju summas novērtējums, kā arī prognozes klūdas novērtējums un prognozes klūdas kā procents no vidējās vērtības, tiek parādītas 3.3. tabulā. Prognozes klūdas tiek aprēķinātās kā kvadrātsakne no MSE, kuri bija aprēķināti, izmantojot formulas (1.2.18) un (1.2.19) priekš notikuma perioda un summāro rezervju prognozes klūdām respektīvi.

Pēc iegūtajiem rezultātiem redzams, ka rezervju novērtējumi tiešām pilnīgi sakrīt ar chain-ladder novērtējumiem. Pirmajiem notikuma periodiem, kur rezervju prognozes ir

3.2. tabula Virs-izklidēta Puasona modeļa parametru novērtējums

	Parametra novērtējums	Standart- kļūda	t-statistika	p -vērtība
c	9.39292	0.13791	68.111	0.0000
α_2	0.28182	0.18005	1.565	0.1210
α_3	0.74644	0.16569	4.505	0.0000
α_4	1.01363	0.15950	6.355	0.0000
α_5	1.08994	0.15799	6.899	0.0000
α_6	1.38169	0.15292	9.035	0.0000
α_7	1.26401	0.15564	8.121	0.0000
α_8	1.50191	0.15209	9.875	0.0000
α_9	1.69699	0.14968	11.337	0.0000
α_{10}	1.79643	0.1488	12.073	0.0000
α_{11}	1.69480	0.15037	11.271	0.0000
α_{12}	1.88874	0.14863	12.708	0.0000
α_{13}	1.99028	0.14838	13.413	0.0000
α_{14}	2.08441	0.14862	14.025	0.0000
α_{15}	2.00537	0.15505	12.933	0.0000
β_2	-0.85208	0.04521	-18.848	0.0000
β_3	-1.87885	0.07268	-25.851	0.0000
β_4	-2.18229	0.08936	-24.421	0.0000
β_5	-2.60309	0.11712	-22.226	0.0000
β_6	-3.19098	0.16695	-19.114	0.0000
β_7	-3.25413	0.18980	-17.145	0.0000
β_8	-3.70992	0.26453	-14.024	0.0000
β_9	-3.57051	0.27649	-12.914	0.0000
β_{10}	-2.67025	0.1996	-13.378	0.0000
β_{11}	-4.72527	0.6452	-7.324	0.0000
β_{12}	-5.28576	1.01396	-5.213	0.0000
β_{13}	-4.39324	0.82792	-5.306	0.0000
β_{14}	-3.66040	0.79523	-4.603	0.0000
β_{15}	-4.25125	1.62441	-2.617	0.0104
ϕ	447.96			

mazas, prognozes nedrošums ir liels, bet vēlākiem notikuma periodiem prognozes kļūda, ka procents no vidējās vērtības samazinās. Summāro rezervju prognozes kļūda sastāda 12% no summāro rezervju vidējās vērtības.

3.2.2. Chain-ladder modeļa Normālā aproksimācija

Izmantojot rezervju novērtējuma iegūšanai chain-ladder modeļa Normālo aproksimāciju, tiek izmantota kopējā modelēšanas procedūra. Šīs procedūras 1. solī tiek iegūti (1.2.21) modeļa parametru novērtējumi, kuri parādīti 3.4. tabulā. Dispersijas komponentes sākuma vērtības tika izvēlētas $\phi_j = 1$, $j = 1, \dots, 15$.

3.3. tabula Rezervju summas novērtējums ar virs-izklaidēto Puasona modeli

Notikuma periods	Rezerves	Prognozes kļūda	Prognozes kļūda, %
1	0	-	-
2	227	488	215%
3	1012	1038	103%
4	1731	1402	81%
5	2049	1530	75%
6	3167	2012	64%
7	5757	2274	39%
8	8820	2893	33%
9	12323	3509	28%
10	16405	4026	25%
11	17508	3970	23%
12	27131	5015	18%
13	39937	6087	15%
14	58621	7496	13%
15	92199	10297	11%
Summa	286887	33129	12%

3.4. tabula: Chain-ladder modeļa Normālā aproksimācija; parametru novērtējums kopējās modelēšanas 1. solī

	Parametra novērtējums	Standart- kļūda	t-statistika	<i>p</i> -vērtība
<i>c</i>	1.4265	0.0073	194.69	0.0000
α_2	-0.3194	0.0099	-32.43	0.0000
α_3	-0.3551	0.0100	-35.67	0.0000
α_4	-0.3828	0.0102	-37.67	0.0000
α_5	-0.4032	0.0104	-38.65	0.0000
α_6	-0.4052	0.0109	-37.09	0.0000
α_7	-0.4133	0.0116	-35.73	0.0000
α_8	-0.4115	0.0124	-33.29	0.0000
α_9	-0.3901	0.0133	-29.42	0.0000
α_{10}	-0.4220	0.0148	-28.61	0.0000
α_{11}	-0.4240	0.0169	-25.14	0.0000
α_{12}	-0.4203	0.0207	-20.34	0.0000
α_{13}	-0.4136	0.0276	-15	0.0000
α_{14}	-0.4195	0.0410	-10.24	0.0000

3.5. tabulā tiek parādīti otrā modeļa (1.2.22) parametru novērtējumi, no kuriem tiek iegūtas precizētas ϕ_j vērtības (3.6. tabula). Izmantojot jaunās ϕ_j vērtības, tiek pārrēķināti svāri W_{ij} .

Pēdējā parametra ϕ_{15} novērtēšanai ir nepietiekama informācija, tā kā šajā punktā ir tikai viens atlikums. ϕ_{15} novērtējums tika paņemts vienāds ar iepriekšējo vērtību ϕ_{14} . Pēdējā procedūras solī pirmais modelis tiek pielāgots atkārtoti, izmantojot precizētas svāru W_{ij} vērtības. Modeļa parametru novērtējumi tiek parādīti 3.7. tabulā.

3.5. tabula: Chain-ladder modeļa Normālā aproksimācija; parametru novērtējums kopējās modelēšanas 2. solī

	Parametra novērtējums	Standart- kļūda	t-statistika	<i>p</i> -vērtība
<i>c</i>	127.1	24.58	5.17	0.0000
α_2	-95.2	35.42	-2.69	0.0086
α_3	-79.5	36.18	-2.20	0.0305
α_4	-125.0	37.05	-3.37	0.0011
α_5	-123.7	38.07	-3.25	0.0016
α_6	-119.6	39.29	-3.05	0.0031
α_7	-124.7	40.76	-3.06	0.0029
α_8	-122.5	42.57	-2.88	0.0050
α_9	-23.7	44.87	-0.53	0.5984
α_{10}	-126.4	47.91	-2.64	0.0098
α_{11}	-127.0	52.14	-2.44	0.0168
α_{12}	-125.3	58.51	-2.14	0.0349
α_{13}	-122.4	69.51	-1.76	0.0817
α_{14}	-127.1	95.19	-1.34	0.1851

3.6. tabula Chain-ladder modeļa Normālā aproksimācija; dispersijas komponentes ϕ_j

	Parametra novērtējums
ϕ_2	127.10
ϕ_3	31.94
ϕ_4	47.59
ϕ_5	2.15
ϕ_6	3.45
ϕ_7	7.49
ϕ_8	2.38
ϕ_9	4.63
ϕ_{10}	103.38
ϕ_{11}	0.70
ϕ_{12}	0.12
ϕ_{13}	1.80
ϕ_{14}	4.74
ϕ_{15}	4.74

Tā kā šajā modelī identitātes saites funkcija tiek izmantota, šajā gadījumā parametru novērtējumi ir uz tās pašas skalas kā attīstības faktori. 3.8. tabulā tiek parādīti attīstības faktori un to standartkļūdas, kuras tiek aprēķinātas izmantojot parametru α_j kovariācijas matricu, ko ieguvām procedūras 3. solī kopā ar parametru α_j novērtējumiem.

3.9. tabulā tiek parādītas ar Normālo modeli iegūtas rezervju prognozes, kā arī procesa un novērtējuma dispersijas un prognozes kļūda, kuras bija aprēķinātas izmantojot rekursīvas formulas iegūtas 1.2.5. apakšnodaļā. Formulas izvedumā tika izmantots, ka attīstības faktori nav korelēti, lai to pārbaudītu praksē tiek aprēķināta parametru λ kova-

3.7. tabula: Chain-ladder modeļa Normālā aproksimācija; parametru novērtējums kopējās modelēšanas 3. solī

	Parametra novērtējums	Standart- kļūda	t-statistika	<i>p</i> -vērtība
<i>c</i>	1.4265	0.0141	101.3	0.0000
α_2	-0.3194	0.0154	-20.7	0.0000
α_3	-0.3551	0.0162	-22.0	0.0000
α_4	-0.3828	0.0142	-27.0	0.0000
α_5	-0.4032	0.0143	-28.3	0.0000
α_6	-0.4052	0.0146	-27.8	0.0000
α_7	-0.4133	0.0143	-29.0	0.0000
α_8	-0.4115	0.0145	-28.3	0.0000
α_9	-0.3901	0.0238	-16.4	0.0000
α_{10}	-0.4220	0.0142	-29.7	0.0000
α_{11}	-0.4240	0.0141	-30.1	0.0000
α_{12}	-0.4203	0.0148	-28.5	0.0000
α_{13}	-0.4136	0.0172	-24.1	0.0000
α_{14}	-0.4195	0.0205	-20.4	0.0000
ϕ	1.1429			

3.8. tabula: Chain-ladder modeļa Normālā aproksimācija; attīstības faktori un to standartkļūdas

	Parametra novērtējums	Standart- kļūda
λ_2	1.4265	0.0141
λ_3	1.1071	0.0063
λ_4	1.0714	0.0079
λ_5	1.0438	0.0017
λ_6	1.0233	0.0024
λ_7	1.0214	0.0037
λ_8	1.0133	0.0024
λ_9	1.0150	0.0036
λ_{10}	1.0365	0.0192
λ_{11}	1.0045	0.0017
λ_{12}	1.0026	0.0010
λ_{13}	1.0062	0.0044
λ_{14}	1.0129	0.0098
λ_{15}	1.0071	0.0149

riācijas matrica (3. pielikums). Apskatot šo matricu varām secināt, ka šie faktori tiešām ir nekorelēti, jo visas matricas vērtības izņemot galveno diagonāli ir nulles.

Šī modeļa attīstības faktoru λ novērtējumi ir identiski chain-ladder attīstības faktoriem. Rezervju novērtējums arī pilnīgi sakrīt ar chain-ladder novērtējumiem un tātad arī ar Puasona modeļa novērtējumiem, bet gan summārā, gan notikuma perioda prognozes kļūda, ir Normālajā modelī ir lielāka nekā Puasona modelī.

3.9. tabula Rezervju summas novērtējums ar chain-ladder Normālo aproksimāciju

Notikuma periods	Rezerves	Procesa dispersija	Novērtējuma dispersija	Prognozes kļūda	Prognozes kļūda, %
1	0	-	-	-	-
2	227	152222	230283	618	273%
3	1012	484798	834041	1148	113%
4	1731	756028	1508139	1505	87%
5	2049	825123	1762066	1608	79%
6	3167	1174472	3186255	2088	66%
7	5757	9953914	5070073	3876	67%
8	8820	13162981	8311035	4634	53%
9	12323	16338598	12380845	5359	43%
10	16405	19235620	15395925	5885	36%
11	17508	17880945	12665669	5527	32%
12	27131	22090034	18739058	6390	24%
13	39937	33989567	24711450	7662	19%
14	58621	44637186	31088669	8702	15%
15	92199	64286143	29746142	9697	11%
Summa	286887	244967631	1570886912	42613	15%

3.2.3. Maka modelis

Maka modeļa parametru λ_j un σ_j^2 novērtējumi aprēķināti saskaņā ar formulām (1.2.24) un (1.2.25) ir parādīti 3.10. tabulā

3.10. tabula Maka modelis; attīstības faktori un dispersijas komponentes

j	λ_j	σ_j^2
2	1.427	136.88
3	1.107	34.60
4	1.071	51.92
5	1.044	2.36
6	1.023	3.83
7	1.021	8.42
8	1.013	2.72
9	1.015	5.40
10	1.036	124.06
11	1.005	0.88
12	1.003	0.16
13	1.006	2.70
14	1.013	9.47
15	1.007	

Līdzīgi, ka Normālajā modelī, nepietiek informācijas, lai novērtētu pēdējo dispersijas komponenti σ_{15}^2 . Apskatīsim divus variantus: pirmais, kad pēdējā dispersijas komponente tiek ņemta $\sigma_{15}^2 = \sigma_{13}^2 = 2.7$, kā piedāvāts [8] un otrs, kad $\sigma_{15}^2 = \sigma_{14}^2 = 9.47$. Rezervju

3.11. tabula Rezervju summas novērtējums ar Maka modeli, $\sigma_{15}^2 = \sigma_{13}^2$

Notikuma periods	Rezerves	Procesa dispersija	Novērtējuma dispersija	Prognozes kļūda	Prognozes kļūda, %
1	0	-	-	-	-
2	227	86757	115000	449	198%
3	1012	623166	731282	1164	115%
4	1731	998148	1362235	1536	89%
5	2049	1089491	1591898	1637	80%
6	3167	1545855	2887348	2106	66%
7	5757	12066033	4957048	4126	72%
8	8820	15931648	8136054	4906	56%
9	12323	19752251	12114260	5645	46%
10	16405	23154776	15071554	6183	38%
11	17508	21477403	12390202	5820	33%
12	27131	26494430	18332343	6695	25%
13	39937	39731833	24125803	7991	20%
14	58621	51553955	30321006	9048	15%
15	92199	72449666	28905733	10068	11%
Summa	286887	286955411	1502943859	42307	15%

3.12. tabula Rezervju summas novērtējums ar Maka modeli, $\sigma_{15}^2 = \sigma_{14}^2$

Notikuma periods	Rezerves	Procesa dispersija	Novērtējuma dispersija	Prognozes kļūda	Prognozes kļūda, %
1	-	-	-	-	-
2	227	304443	403551	841	371%
3	1012	969595	1462071	1559	154%
4	1731	1450683	2609232	2015	116%
5	2049	1577912	3044515	2150	105%
6	3167	2199738	5490875	2773	88%
7	5757	12647321	7014573	4434	77%
8	8820	16669061	11447242	5302	60%
9	12323	20648503	17005534	6136	50%
10	16405	24144732	21039075	6722	41%
11	17508	22371701	17260180	6295	36%
12	27131	27580128	25509952	7286	27%
13	39937	40933565	32919604	8594	22%
14	58621	52874299	40936399	9686	17%
15	92199	73669669	37968978	10566	11%
Summa	286887	298041353	2251297488	50491	18%

prognozes, kā arī to prognozes kļūdas, ko iegūst pie viena vai otra pieņēmuma par σ_{15} vērtību tiek parādītas 3.11. un 3.12. tabulās. Salīdzinot šīs divas tabulas, varām secināt, ka prognozes kļūdas novērtējumi, īpaši pirmo gadu rezultāti, ir ļoti jutīgi pret vienu vienīgu parametru.

Maka modeļa rezervju novērtējumi kā arī attīstības faktoru λ novērtējumi ir identiski chain-ladder novērtējumiem. Normālajā modelī izmantotie ϕ_j novērtējumi ir tuvi Maka

modeļa σ_j^2 novērtējumiem, arī prognozes kļūdas ir tuvas (īpaši kad tiek izvēlēta $\sigma_{15}^2 = \sigma_{13}^2$). Faktiski atšķirības rodas tāpēc, ka Maka modelī σ_j^2 novērtējumi ir nenovirzīti, bet Normālajā modelī biasa korekcija nav veikta. Novērtējot σ_j^2 bez biasa korekcijas, iegūst ϕ_j identiskus novērtējumus [13].

3.2.4. Log-Normālais modelis

Pielāgojot datiem log-Normālo modeli ar chain-ladder tipa prediktoru, kurš dots ar (1.2.7.), tiek iegūti parametru novērtējumi, kas parādīti 3.13. tabulā. No tiem iegūtas rezervju summas prognozes un prognozes kļūdas ir parādītas 3.14. tabulā.

3.13. tabula Log-Normālā modeļa parametru novērtējums

	Parametra novērtējums	Standart- kļūda	t-statistika	<i>p</i> -vērtība
<i>c</i>	9.5319	0.3390	28.11	0.0000
α_2	-0.3389	0.3286	-1.03	0.3050
α_3	0.2315	0.3379	0.69	0.4951
α_4	1.0760	0.3474	3.10	0.0026
α_5	0.8276	0.3575	2.32	0.0229
α_6	1.4320	0.3689	3.88	0.0002
α_7	1.3898	0.3819	3.64	0.0005
α_8	1.5498	0.3973	3.90	0.0002
α_9	1.6566	0.4160	3.98	0.0001
α_{10}	1.6576	0.4393	3.77	0.0003
α_{11}	1.5270	0.4698	3.25	0.0016
α_{12}	1.6581	0.5118	3.24	0.0017
α_{13}	1.7976	0.5746	3.13	0.0024
α_{14}	1.9325	0.6825	2.83	0.0057
α_{15}	1.8664	0.9331	2.00	0.0485
β_2	-0.8353	0.3286	-2.54	0.0127
β_3	-1.8560	0.3379	-5.49	0.0000
β_4	-2.1674	0.3474	-6.24	0.0000
β_5	-2.6596	0.3575	-7.44	0.0000
β_6	-3.2811	0.3689	-8.90	0.0000
β_7	-3.5261	0.3819	-9.23	0.0000
β_8	-4.5033	0.3973	-11.34	0.0000
β_9	-3.7769	0.4160	-9.08	0.0000
β_{10}	-4.1163	0.4393	-9.37	0.0000
β_{11}	-5.0490	0.4698	-10.75	0.0000
β_{12}	-5.7015	0.5118	-11.14	0.0000
β_{13}	-4.4824	0.5746	-7.80	0.0000
β_{14}	-4.1103	0.6825	-6.02	0.0000
β_{15}	-4.3902	0.9331	-4.71	0.0000
<i>sigma</i> ²	0.7558			

3.14. tabula: Rezervju summas novērtējums ar log-Normālo modeli (vidējās vērtības novērtējums)

Notikuma periods	Rezerves	Prognozes kļūda	Prognozes kļūda, %
1	0	-	-
2	199	400	201%
3	819	1029	126%
4	2663	2682	101%
5	2259	2143	95%
6	4764	4112	86%
7	6090	4606	76%
8	9677	6692	69%
9	12199	8108	66%
10	15957	10339	65%
11	18320	11954	65%
12	30248	21252	70%
13	53353	42014	79%
14	94393	87041	92%
15	200560	283299	141%
Summa	451501	305346	68%

3.15. tabula: Rezervju summas novērtējums ar log-Normālo modeli (mediānas novērtējums)

Notikuma periods	Rezerves	Prognozes kļūda	Prognozes kļūda, %
1	0	-	-
2	122	245	201%
3	501	629	126%
4	1622	1634	101%
5	1371	1301	95%
6	2880	2485	86%
7	3663	2771	76%
8	5786	4002	69%
9	7239	4811	66%
10	9375	6074	65%
11	10615	6927	65%
12	17169	12062	70%
13	29268	23047	79%
14	48385	44617	92%
15	83965	118604	141%
Summa	221960	132706	60%

Lai aprēķinātu notikuma perioda un summāro rezervju prognozes kļūdas saskaņā ar formulām (1.2.29) un (1.2.30) ir jāaprēķinā variācijas-kovariācijas matricu $\sigma^2 \mathbf{X}_f (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}_f^T$, kur σ^2 ir modeļa dispersija un ir atkarīga no datiem, \mathbf{X}_f ir nākotnes vērtību dizaina matrica, $(\mathbf{X}^T \mathbf{X})^{-1}$ ir modeļa informācijas matrica un \mathbf{X} ir dizaina matrica [14].

No 3.14. tabulas ir skaidrs, ka sagaidāmo atlīdzību rezervju novērtējums, ko dod

3.16. tabula Gamma modeļa parametru novērtējums

	Parametra novērtējums	Standart- kļūda	t-statistika	<i>p</i> -vērtība
<i>c</i>	9.4944	0.2385	39.81	0.0000
α_2	-0.0228	0.2311	-0.10	0.9216
α_3	0.4134	0.2377	1.74	0.0854
α_4	0.9825	0.2443	4.02	0.0001
α_5	0.9377	0.2515	3.73	0.0003
α_6	1.5317	0.2595	5.90	0.0000
α_7	1.3495	0.2686	5.02	0.0000
α_8	1.5004	0.2795	5.37	0.0000
α_9	1.6653	0.2926	5.69	0.0000
α_{10}	1.6858	0.3091	5.46	0.0000
α_{11}	1.5458	0.3305	4.68	0.0000
α_{12}	1.6902	0.3600	4.70	0.0000
α_{13}	1.8334	0.4042	4.54	0.0000
α_{14}	1.9671	0.4801	4.10	0.0001
α_{15}	1.9039	0.6564	2.90	0.0047
β_2	-0.8290	0.2311	-3.59	0.0005
β_3	-1.8440	0.2377	-7.76	0.0000
β_4	-2.0799	0.2443	-8.51	0.0000
β_5	-2.6629	0.2515	-10.59	0.0000
β_6	-3.2008	0.2595	-12.34	0.0000
β_7	-3.3771	0.2686	-12.57	0.0000
β_8	-3.9442	0.2795	-14.11	0.0000
β_9	-3.5104	0.2926	-12.00	0.0000
β_{10}	-3.2608	0.3091	-10.55	0.0000
β_{11}	-4.5918	0.3305	-13.89	0.0000
β_{12}	-5.2492	0.3600	-14.58	0.0000
β_{13}	-4.1218	0.4042	-10.20	0.0000
β_{14}	-3.6093	0.4801	-7.52	0.0000
β_{15}	-4.3528	0.6564	-6.63	0.0000
ϕ	0.3740			

log-Normālais modelis ir tāls no chain-ladder novērtējuma, tas pats ir arī ar prognozes kļūdu, kura pat procentuāli ir vairakkārt lielāka nekā tās ko ieguvām ar iepriekš apskatītajiem modeļiem. Tas ir izraisīts ar to, ka vienādojumā (1.2.28) tiek iekļauta dispersijas komponente, kura var dot nozīmīgu efektu, ja novērojumi ir stipri izkliedēti.

Salīdzinājumam 3.15. tabulā ir parādīti rezervju mediānas novērtējuma rezultāts (kas neiekļauj dispersijas komponenti izteiksmē priekš \hat{C}_{ij} (1.2.28)). Šis rezervju novērtējums ir jau tuvāk chain-ladder novērtējumam, kaut gan prognozes kļūda procentuāli ir joprojām liela.

3.17. tabula Rezervju summas novērtējums ar Gamma modeli

Notikuma periods	Rezerves	Prognozes kļūda	Prognozes kļūda, %
1	0	-	-
2	167	150	90%
3	802	494	62%
4	1993	1008	51%
5	2084	994	48%
6	4397	1923	44%
7	5629	2281	41%
8	8326	3172	38%
9	11179	4148	37%
10	13859	5098	37%
11	14588	5464	37%
12	21875	8704	40%
13	35626	15856	45%
14	55750	28682	51%
15	91254	64184	70%
Summa	267529	76828	29%

3.2.5. Gamma modelis

Pielāgojot datiem Gamma modeli, kurš definēts 1.2.8. apakšnodaļā, tiek iegūti parametru novērtējumi, kas parādīti 3.16. tabulā. No tiem iegūtie rezervju summas prognozes un prognozes kļūdas ir parādītas 3.17. tabulā. Ievērosim, ka sagaidāmas rezerves ir tuvas, bet ne identiskas chain-ladder rezervēm, bet prognozes kļūdas ir daudz lielākas nekā virs-izklidēta Puasona modelī. Būtībā relatīvi lielāki svāri tika uzlikti uz mazākām pieaugumu vērtībām un mazāki uz lielākām pieaugumu vērtībām, izraisot lielāku svaru uz astes, kurš ir vairāk izklidēts.

3.3. IBNR atlīdzību rezervju aprēķināšana ar butstrapa metodi

Lai novērtētu rezervju prognozes kļūdu ar butstrapa metodi ir jāveic sekojoši soļi:

1. novērtēt modeļa parametrus c , α_i , β_j , $i, j = 1, \dots, n$ un ϕ ;
2. aprēķināt pielāgotās vērtības $\hat{\mu}_{ij}$, $i = 1, \dots, n$ un $j = 1, \dots, n - i + 1$;
3. aprēķināt atlikumus $r_{ij} = f(C_{ij}, \hat{\mu}_{ij})$;

4. prognozēt \hat{C}_{ij} , \hat{R}_i , \hat{R} , $i = 2, \dots, n$, $j = n - i + 2, \dots, n$ ar sākotnējiem datiem;
5. atkārtot butstrapa cilpu B reizes (šeit tiek ņemts $B = 10000$):
 - pārkārtot 3. solī iegūtus atlikumus, iegūstot butstrapotus atlikumus r_{ij}^* ;
 - izveidot pseido-datus C_{ij}^* , atrisinot $r_{ij}^* = f(C_{ij}^*, \hat{\mu}_{ij})$;
 - novērtēt modeli ar pseido-datiem, iegūt butstrapa prognozes \hat{C}_{ij}^* , \hat{R}_i^* , \hat{R}^* un saglabāt tos.

Butstrapa prognozes kļūda tiek atrasta pēc formulas (2.3.1). Ticamības intervāla augšējā robeža tiek aprēķināta, aproksimējot rezervju sadalījumu ar Normālo sadalījumu ar matemātisko cerību vienādu ar sākotnējo prognozi (ar oriģināliem datiem) un standartnovirzi vienādu ar prognozes standartkļūdu.

Apskatīsim modeli ar virs-izkļiedēto Puasona datu sadalījumu, lai definētu atlikumus. Šim modelim ir tā priekšrocība, ka attiecīgais vispārinātais lineārais modelis ģenerē tos pašus novērtējumus kā chain-ladder metode, tāpēc veicot butstrapa procedūru, rezervju novērtējums katrai uzģenerētai pseido-datu kopai var tikt iegūts ar chain-ladder metodi. Turpretim otram apskatītam Gamma modelim butstrapa procedūras realizācija ir sarežģītāka, jo katrā butstrapa “iterācijā” ir nepieciešams novērtēt Gamma vispārināto lineāro modeli.

3.18. tabulā ir parādītas virs-izkļiedēta Puasona modeļa rezervju prognozes, rezervju vidējās vērtības butstrapa novērtējums, prognozes kļūda un 95% Normālā ticamības intervāla augšējā robeža. 3.1. attēlā tiek parādīta ar butstrapa procedūru iegūto summāro rezervju histogramma.

Tā kā pieejami dati kombinācijā ar modeli izmantotu lineāro struktūru noved pie dažiem atlikumiem, kuri ir vienādi ar 0 (tipiski $C_{1,n} = \hat{\mu}_{1,n}$ un $C_{n,1} = \hat{\mu}_{n,1}$), šie atlikumi nav jāņem vērā, kā gadījuma lieluma realizāciju un tāpat tos nav jāņem vērā butstrapa procedūrā [15]. 3.19. tabulā tiek parādīti rezervju butstrapa novērtējumi virs-izkļiedēta Puasona modelim, izmantojot koriģētus atlikumus. Salīdzinot rezultātus ar atlikumu korekciju un bez korekcijas, redzam, ka atšķirības rezultātos ir ļoti mazas.

3.20. tabulā ir parādītas Gamma modeļa rezervju prognozes, rezervju vidējās vērtības butstrapa novērtējums, prognozes kļūda un 95% Normālā ticamības intervāla augšējā

3.18. tabula Rezervju butstrapa novērtējums, Puasona modelis

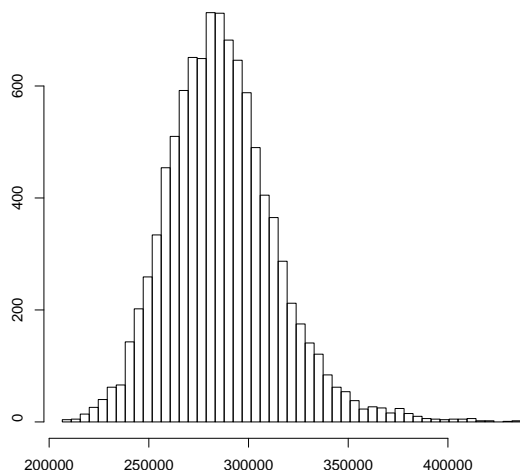
Notikuma periods	Chain-ladder rezerves	Butstrapa rezerves vidējā vērtība	Prognozes kļūda	95% tic. int. augšējā robeža
1	-	-	-	-
2	227	232	492	1192
3	1012	1026	1048	3067
4	1731	1742	1421	4516
5	2049	2060	1555	5097
6	3167	3171	2041	7166
7	5757	5766	2296	10257
8	8820	8831	2920	14542
9	12323	12334	3542	19266
10	16405	16420	4050	24343
11	17508	17520	4000	25347
12	27131	27123	5013	36957
13	39937	39908	6093	51879
14	58621	58562	7525	73371
15	92199	92128	10229	112247
Summa	286887	286821	33381	352311

3.19. tabula Rezervju butstrapa novērtējums, Puasona modelis ar korekciju

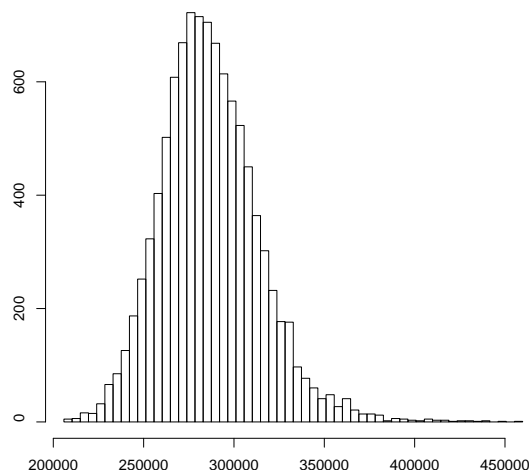
Notikuma periods	Chain-ladder rezerves	Butstrapa rezerves vidējā vērtība	Prognozes kļūda	95% tic. int. augšējā robeža
1	-	-	-	-
2	227	225	497	1201
3	1012	1007	1055	3079
4	1731	1724	1419	4513
5	2049	2036	1547	5081
6	3167	3150	2038	7162
7	5757	5741	2287	10239
8	8820	8796	2909	14521
9	12323	12301	3531	19244
10	16405	16362	4051	24345
11	17508	17459	3985	25319
12	27131	27075	5023	36977
13	39937	39875	6147	51985
14	58621	58538	7521	73362
15	92199	92090	10317	112421
Summa	286887	286380	33392	352335

robeža. 3.3. attēlā tiek parādīta ar butstrapa procedūru iegūto summāro rezervju histogramma.

Gamma modelis parasti dod mazākus rezervju novērtējumus nekā Puasona modelis, bet šī atšķirība nav liela. Tomēr šo divu modeļu prognozes standartkļūdas atšķiras un tāpēc arī intervāla augšējās robežas atšķiras. Kopumā ar butstrapa metodi iegūtas prognozes kļūdas abiem modeļiem ir ļoti tuvas tam, ko ieguvām ar analītiskajām metodēm.



3.1. att.: Summāro rezervju histogramma, Puasona modelis bez korekcijas



3.2. att.: Summāro rezervju histogramma, Puasona modelis ar korekciju

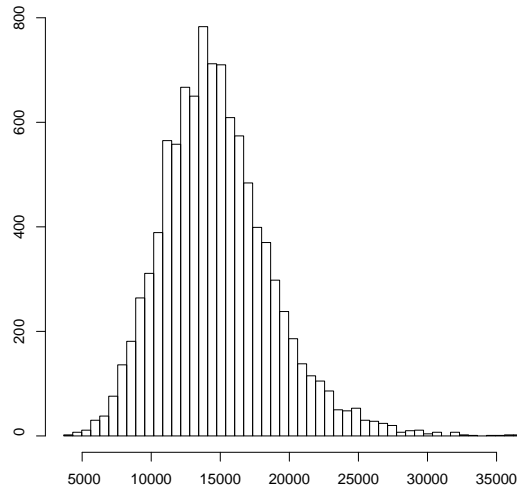
3.20. tabula Rezervju butstrapa novērtējums, Gamma modelis

Notikuma periods	Gamma modeļa rezerves	Butstrapa rezerves vidējā vērtība	Prognozes kļūda	95% tic. int. augšējā robeža
1	-	-	-	-
2	167	170	115	315
3	802	818	344	1244
4	1993	2030	725	2922
5	2084	2117	731	3020
6	4397	4458	1438	6240
7	5629	5695	1624	7710
8	8326	8452	2354	11343
9	11179	11338	3240	15332
10	13859	14057	4159	19190
11	14588	14751	4564	20437
12	21875	22094	7404	31363
13	35626	35802	13663	53136
14	55750	56478	26047	89130
15	91254	94410	66445	176406
Summa	267529	272671	76788	365936

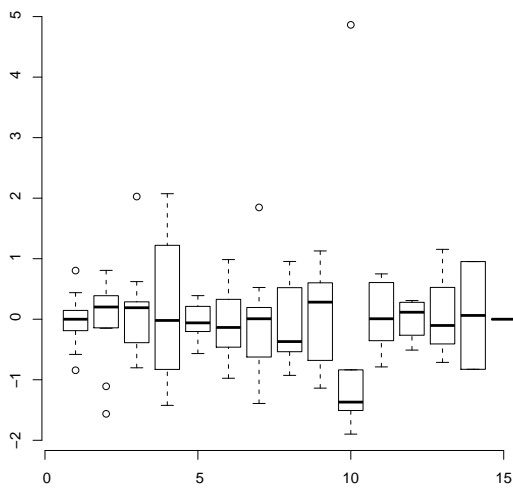
3.4. Modeļu pārbaude

Parasti praksē tiek konstruēti standartizētie atlikumi pret modeļa pielāgotajam vērtībām, pret attīstības periodu un notikuma periodu.

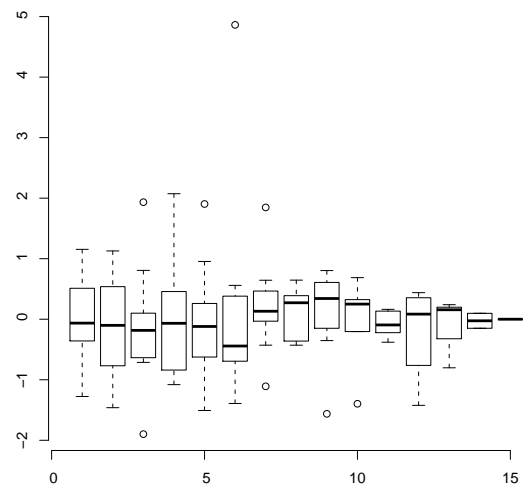
3.4., 3.5. un 3.10. attēlos tiek parādīti šie grafiki Puasona sadalījumam. 3.6., 3.7. un 3.11. attēlos tiek parādīti attiecīgie grafiki Gamma sadalījumam un 3.8., 3.9. un 3.12. attēlos - log-Normālajam sadalījumam. Kā arī 3.13., 3.14. un 3.15. attēlos tiek



3.3. att. Summāro rezervju histogramma, Gamma modelis

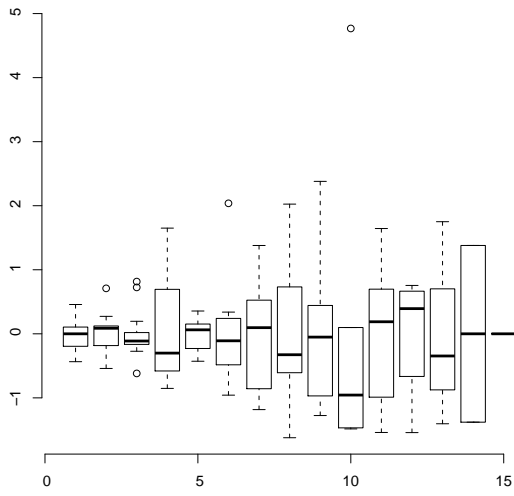


3.4. att.: Pīrsona atlikumi pret attīstības periodu; virs-izkļiedēts Puasona modelis

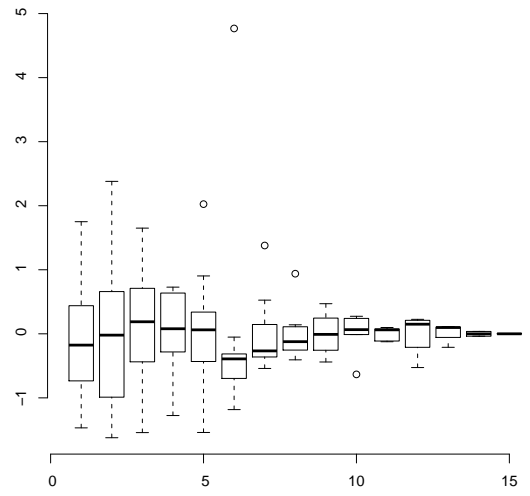


3.5. att.: Pīrsona atlikumi pret notikuma periodu; virs-izkļiedēts Puasona modelis

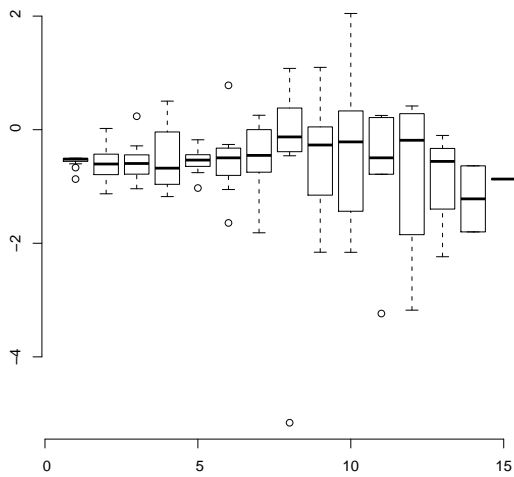
parādītas šo triju sadalījumu autokorelāciju funkcijas, pēc kurām var noteikt, ja atlikumi ir atkarīgi. [4] norādīts, ka ja aptuveni 95% punktu standartizēto atlikumu pret notikuma vai attīstības periodu atrodas robežās 0 ± 2 , tad modelēšana ir veiksmīga. Mūsu gadījumā atlikumi apmierina šo nosacījumu. Bet apskatot standartizētus atlikumus pret pielāgotām vērtībām, redzam, ka tie nav vienādi sadalīti, grafiki rāda heteroskedastisitātes klātbūtni.



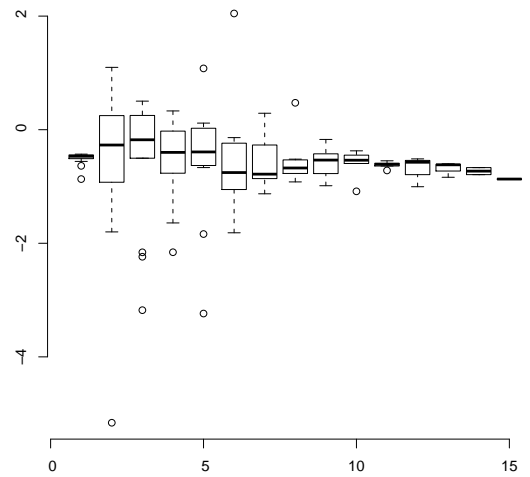
3.6. att.: Pīrsona atlikumi pret attīstības periodu; Gamma modelis



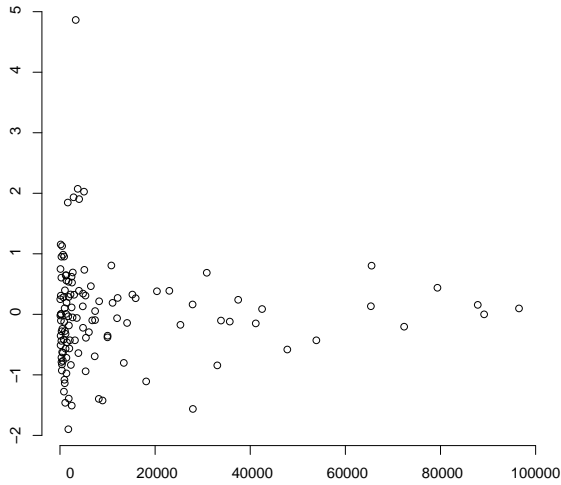
3.7. att.: Pīrsona atlikumi pret notikuma periodu; Gamma modelis



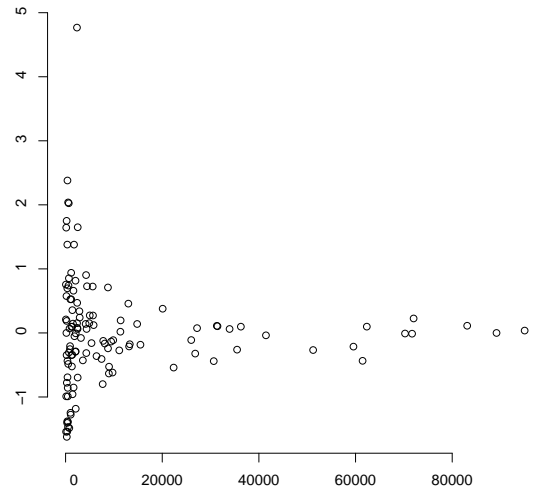
3.8. att.: Pīrsona atlikumi pret attīstības periodu; log-Normālais modelis



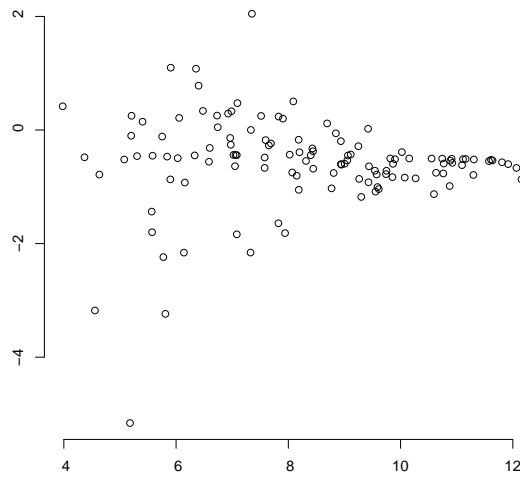
3.9. att.: Pīrsona atlikumi pret notikuma periodu; log-Normālais modelis



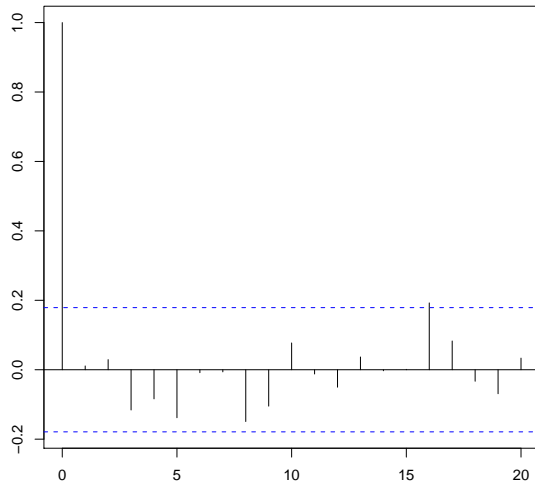
3.10. att.: Pīrsona atlikumi pret pielāgotām vērtībām \hat{C}_{ij} ; virs-izklaidēts Puasona modelis



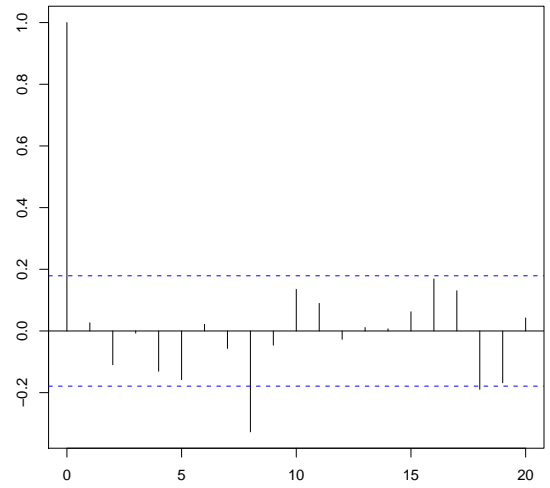
3.11. att.: Pīrsona atlikumi pret pielāgotām vērtībām \hat{C}_{ij} ; Gamma modelis



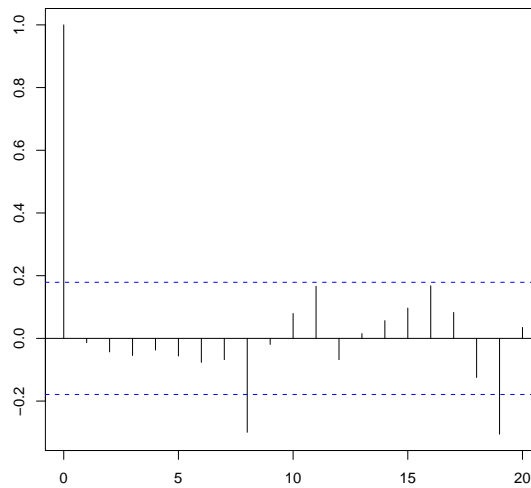
3.12. att. Pīrsona atlikumi pret pielāgotām vērtībām $\ln \hat{C}_{ij}$; log-Normālais modelis



3.13. att.: Pīrsona atlikumu ACF; virs-
izklidēts Puasona modelis



3.14. att.: Pīrsona atlikumu ACF; Gamma
modelis



3.15. att. Pīrsona atlikumu ACF; log-Normālais modelis

Secinājumi

Darbā tika apskatītas IBNR rezervju aprēķināšana un prognozes kļūdas noteikšana ar virs-izkliedēto Puasona, Normālo, Gamma, log-Normālo un Maka modeli. Šie modeļi tika aplūkoti vispārinātā lineārā modeļa teorijas ietvaros, kura dod iespēju iegūt modeļa parametru novērtējumus, novērtējumu kļūdas un definēt atlikumus, modeļa atbilstības pārbaudei. Otrā apskatītā metode ir butstrapa datu pārkārtošanas metode, kura dod iespēju novērtēt pilnu rezervju prognozes sadalījumu. Tika secināts, ka ar butstrapa metodi tiek iegūti prognozes standartkļūdu novērtējumi ļoti tuvu tiem, ko iegūst ar analītiskām tuvinātām izteiksmēm. Kā arī uzreiz tiek iegūts pilns prognozes sadalījums, no kā var novērtēt nepieciešamus sadalījuma raksturlielumus.

Izņemot Maka un Normālās aproksimācijas modeli, pārējiem apskatītiem modeļiem ir vienāda lineāra prediktora struktūra, tāda, ko 1982. gadā piedāvāja Kremers [16]. Šie modeļi atšķirās ar pieņēmumu par kļūdu sadalījumu, izvēle bija starp virs-izkliedēto Puasona, log-Normālo un Gamma sadalījumu.

Puasona modelis, pie dažiem pieņēmumiem, dod ar chain-ladder identiskus rezervju novērtējumus. Arī Maka un Normālais modelis dod tādus pašus kā chain-ladder metode rezervju novērtējumus, bet log-Normālais un Gamma modeļi dod chain-ladder tuvus, bet ne identiskus rezervju prognozes.

Taču lielāku interesi izraisa rezervju novērtējuma prognozes kļūdas, nevis paši novērtējumi. Modeļiem, kuros tiek izdarīti pieņēmumi par sadalījumu, prognozes kļūdas var tikt iegūtas gan analītiski, gan ar rezidiju butstrapa metodi. Rezidiju butstrapa procedūra tika apskatīta tikai Puasona un Gamma modelim. Bet tā kā atlikumi tiek definēti arī log-Normālajam modelim, tad arī šī sadalījuma gadījumā prognozes kļūdu var iegūt ar butstrapa metodi.

Izdarot pietiekoši daudz butstrapa “īterāciju” (10 000) butstrapa rezervju novērtējumi un prognozes kļūdas praktiski sakrīta ar tiem, ko ieguvām analītiski. Taču salīdzinot prognozes kļūdu novērtējumu ar dažādiem modeļiem atšķirības var būt būtiskas, tāpēc ir jābūt ļoti uzmanīgiem izdarot secinājumus no iegūtajiem rezultātiem. Turpmākā analīze ir nepieciešama, lai pamatotu konkrētu atlikumu sadalījuma veidu izmantošanu.

Izmantotā literatūra un avoti

- [1] P. McCullagh and J. Nelder. *Generalised linear models, 2nd edition*. Chapman and Hall, London, 1989.
- [2] C. E. McCulloch and S. R. Searle. *Generalized, Linear, and Mixed Models*. John Wiley and Sons, Inc., New York, 2001.
- [3] B. Zehnwirth. The chain ladder technique - a stochastic model. *Claims reserving manual*, 2, 1997.
- [4] A. E. Renshaw and R. J. Verrall. A stochastic model underlying the chain-ladder technique. *B.A.J.*, 4:903–923, 1998.
- [5] R. J. Verrall. Chain ladder and maximum likelihood. *J.I.A*, 118:489–499, 1991.
- [6] A. E. Renshaw. On the second moment properties and the implementation of certain glim based stochastic claims reserving models. *Actuarial Research Paper No.65, Department of Actuarial Science Statistics, City University, London*, 1994.
- [7] T. Mack. Measuring the variability of chain-ladder reserve estimates. Casualty Actuarial Society, 1994.
- [8] T. Mack. Distribution-free calculation of the standard error of chain ladder reserve estimates. *Astin bulletin*, 23:213–225, 1993.
- [9] T. Mack. A simple parametric model for rating automobile insurance or estimating ibnr claims reserves. *Astin bulletin*, 22:93–109, 1991.
- [10] B. Efron and R. J. Tibshirani. *An introduction to the bootstrap*. Chapman and Hall, 1993.
- [11] A. C. Davison and D. V. Hinkley. *Bootstrap methods and their application*. Cambridge University Press, 1997.

- [12] D. A. Freedman. Bootstrapping regression models. *Annals of Statistics*, 9, 1981.
- [13] P. D. England and R. J. Verrall. *Stochastic claims reserving in general insurance*. Bell and Bain Ltd, Glasgow, 2002.
- [14] S. Christofides. Regression models based on log-incremental payments. *Claims reserving manual*, 2, 1990.
- [15] Paulo J. R. Pinheiro, Joao Manuel Andrade e Silva, and Maria de Lourdes Centeno. Bootstrap methodology in claim reserving, June 2000.
- [16] E. Kremer. Ibrn claims and the two way model of anova. *Scandinavian Actuarial Journal*, pages 47–55, 1982.

1. Atlīdzību pieaugumu attīstības trijstūris

		Attīstības periods														
Notikuma periods (i)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	11854	6233	2094	795	810	381	179	135	229	52	270	102	446	663	171	
2	16621	6613	3079	2277	766	1188	211	2	953	75	52	77	166	55		
3	24736	12575	3025	5043	1706	853	1044	285	872	71	227	6	46			
4	29830	13750	8102	6411	2569	599	1283	559	233	1443	513	252				
5	35220	16080	4848	6583	2588	1094	1531	1471	241	885	20					
6	45103	21537	6047	3928	3463	1437	580	937	1779	9230						
7	42868	14964	7284	4985	2637	1719	3224	1311	1667							
8	51789	24236	8648	5593	4512	2544	1668	1816								
9	69861	22412	9260	7486	5355	3454	3088									
10	71194	33416	11476	5490	5838	3350										
11	66089	28456	9182	7200	4513											
12	81970	33447	12754	6100												
13	88816	38448	11451													
14	97145	40519														
15	89169															

2. Rezervju aprēķins ar chain-ladder metodi

i	Attisīfības periods															Reserves
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	11854	18087	20181	20976	21786	22167	22346	22481	22710	22762	23032	23134	23580	24243	24414	0
2	16621	23234	26313	28590	29356	30544	30755	30757	31710	31785	31837	31914	32080	32135	32362	227
3	24736	37311	40336	45379	47085	47938	48982	49267	50139	50210	50437	50443	50489	51140	51501	1012
4	29830	43580	51682	58093	60662	61261	62544	63103	63336	64779	65292	65544	65953	66804	67275	1731
5	35220	51300	56148	62731	65319	66413	67944	69415	69656	70541	70561	70742	71183	72101	72610	2049
6	45103	66640	72687	76615	80078	81515	82095	83032	84811	94041	94465	94707	95298	96527	97208	3167
7	42868	57832	65116	70101	72738	74457	77681	78992	80659	83600	83977	84192	84718	85810	86416	5757
8	51789	76025	84673	90266	94778	97322	98990	100806	102323	106054	106532	106805	107471	108858	109626	8820
9	69861	92273	101533	109019	114374	117828	120916	122520	124363	128898	129479	129811	130621	132306	133239	12323
10	71194	104610	116086	121576	127414	130764	133558	135329	137365	142375	143016	143383	144277	146138	147169	16405
11	66089	94545	103727	110927	115440	118129	120652	122253	124092	128617	129197	129528	130336	132017	132948	17508
12	81970	115417	128171	134271	140147	143411	146475	148417	150650	156144	156848	157250	158231	160272	161402	27131
13	88816	127264	138715	148621	155125	158738	162129	164279	166751	172832	173611	174056	175141	177401	178652	39937
14	97145	137664	152406	163290	170436	174405	178132	180494	183210	189891	190747	191235	192428	194911	196285	58621
15	89169	127202	140824	150881	157483	161151	164594	166777	169286	175460	176251	176702	177804	180098	181368	92199
Summa																286887
Attisīfības faktori	1.427	1.107	1.107	1.071	1.044	1.023	1.021	1.013	1.015	1.036	1.005	1.003	1.006	1.013	1.007	

3. Attīstības faktoru λ kovariācijas matrica

	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2	0.000198	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	0.000000	0.000040	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
4	0.000000	0.000000	0.000063	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
5	0.000000	0.000000	0.000000	0.000003	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
6	0.000000	0.000000	0.000000	0.000000	0.000006	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
7	0.000000	0.000000	0.000000	0.000000	0.000000	0.000014	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
8	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000006	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
9	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000013	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
10	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000367	0.000000	0.000000	0.000000	0.000000	0.000000
11	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000003	0.000000	0.000000	0.000000	0.000000
12	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000001	0.000000	0.000000	0.000000
13	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000019	0.000000	0.000000
14	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000097	0.000000
15	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000223

A Izveidoto programmu kods

A1. Virs-izklaidēts Puasona modelis

```
data <- read.table("data",header=F)
data <- as.matrix(data)
claims <- as.vector(data)
n.origin <- nrow(data)
n.dev <- ncol(data)
origin <- factor(row <- rep(1:n.origin, n.dev))
dev <- factor(col <- rep(1:n.dev, each=n.origin))
data.cumul<-matrix(NA, nrow(data), ncol(data))
for (i in 1:n.origin) {
  for (j in 1:n.dev) {
    data.cumul[i,j]<-sum(data[i,1:j])
  }
}
claims.cumul <- as.vector(data.cumul)
claim.data <- data.frame(claims=claims, origin=origin, dev=dev)
quasipoisson <- function (link = "log")
{
  linktemp <- substitute(link)
  if (!is.character(linktemp)) {
    linktemp <- deparse(linktemp)
    if (linktemp == "link")
      linktemp <- eval(link)
  }
  if (any(linktemp == c("log", "identity", "sqrt")))
    stats <- make.link(linktemp)
  else stop(paste(linktemp, "link not available for poisson",
    "family; available links are", "\"identity\", \"log\" and
    \"sqrt\""))
  variance <- function(mu) mu
  validmu <- function(mu) all(mu > 0)
  dev.resids <- function(y, mu, wt) wt*(y-mu)^2/mu ###
  aic <- function(y, n, mu, wt, dev) NA
  initialize <- expression({
    n <- rep(1, nobs)
    mustart <- rep(mean(y), length(y)) ###
  })
  structure(list(family = "quasipoisson", link = linktemp,
    linkfun = stats$linkfun, linkinv = stats$linkinv, variance =
variance,
    dev.resids = dev.resids, aic = aic, mu.eta = stats$mu.eta,
    initialize = initialize, validmu = validmu, valideta =
stats$valideta),
    class = "family")
}
model <- glm(claims ~ origin + dev, family = quasipoisson(),subset=!is.na(claims), data=claim.data)
summary(model)
coef <- model$coefficients # Get coefficients
disp <- summary(model)$dispersion
cov.param <- disp * summary(model)$cov.unscaled
n.fut.points <- length(claims[is.na(claims)])
fut.design <- matrix(0, nrow = n.fut.points, ncol=length(coef))
fut.points <- claims
fut.points[!is.na(claims)] <- 0
```

```

fut.points[is.na(claims)] <- 1:n.fut.points
for(p in 1:n.fut.points){
  fut.design[p, 1] <- 1
  fut.design[p, 1 + as.numeric(origin[match(p, fut.points)]) - 1] <- 1
  fut.design[p, 1 + (n.origin-1) + as.numeric(dev[match(p, fut.points)]) - 1] <- 1
}
fitted.values <- diag(as.vector(exp(fut.design %*% coef)))
total.reserve <- sum(fitted.values)
data.fitted <- data
AccY.reserve <- c(rep(0,n.origin))
for (i in 1:n.origin) {
  for (j in 1:n.dev) {
    if(!is.na(data[i,j])) {data.fitted[i,j]<-data[i,j]}
    else {data.fitted[i,j]<- exp(coef[1]+coef[i]+coef[n.origin+j-1])
      AccY.reserve[i]<- AccY.reserve[i]+ data.fitted[i,j]
    }
  }
}

cov.pred <- fut.design %*% cov.param %*% t(fut.design)
cov.fitted <- fitted.values %*% cov.pred %*% fitted.values
total.rmse <- sqrt(disps*total.reserve+sum(cov.fitted))
total.predictionerror <- round(100*total.rmse/total.reserve)
cov.index<-matrix(0, nrow = n.dev, ncol=n.origin)
cov.index[2,1]<-n.fut.points-n.origin+2
for (i in 3:n.origin) {
  cov.index[i,1]<-n.fut.points-n.origin+i
  x<-cov.index[i,1]
  for (k in 3:i-1) {
    x<-x-(n.dev-k+1)
    cov.index[i,k]<-x
    x<-cov.index[i,k]
  }
}
AccY.rmse <- c(rep(0,n.origin))
AccY.rmse.fun <- function(i) { sqrt(disps*AccY.reserve[i]+sum(cov.fitted[cov.index[i,],cov.index[i,]])) }
AccY.rmse <- sapply(2:n.origin,AccY.rmse.fun)
AccY.predictionerror <- c(rep(0,n.origin))
AccY.predictionerror <- round(100*AccY.rmse[1:(n.origin-1)]/AccY.reserve[2:n.origin])

Poisson.model.result<- data.frame(row.names =c(paste("Year", 1:n.origin, sep = ""),"Overall"),
  Reserve=c(round(AccY.reserve),round(total.reserve)), Prediction.error=c("-",round(AccY.rmse),round(total.rmse)),
  Prediction.error.percent=c("-",AccY.predictionerror,total.predictionerror))

```

A2. Butstraps Puasona modelim

```

data <- read.table("data.txt",header=F)
data <- as.matrix(data)
data
claims <- as.vector(data)
n.origin <- nrow(data)
n.dev <- ncol(data)
origin <- factor(row <- rep(1:n.origin, n.dev))
dev <- factor(col <- rep(0:(n.dev-1), each=n.origin))
delay<-factor(col <- rep(1:(n.dev), each=n.origin))
data.cumul.fun <- function(dati) {
  data.cumul<-matrix(NA, nrow(data), ncol(data))
  for (i in 1:n.origin) {
    for (j in 1:n.dev) {
      data.cumul[i,j]<-sum(dati[i,1:j])
    }
  }
  return(data.cumul)
}
cl.lambda.fun <- function(dati.cumuleti) {
  cl.lambda<-c(rep(0,n.origin))
  for (j in 2:n.dev) {
    cl.lambda[j]<-sum(dati.cumuleti[1:(n.origin-j+1),j])/sum(dati.cumuleti[1:(n.origin-j+1),j-1])
  }
  return(cl.lambda)
}

```

```

}
Chain.ladder.reserve <- function(dati) {
  data.cumul <- data.cumul.fun(dati)
  cl.lambda<- cl.lambda.fun(data.cumul)
  Ult.claim<-c(rep(0,n.origin))
  Ult.claim[1]<-data.cumul[1,n.dev]
  for (i in 2:n.origin) {
    Ult.claim[i]<-data.cumul[i,n.dev-i+1]
    for (j in (n.dev-i+2):n.dev) {
      Ult.claim[i]<-Ult.claim[i]*cl.lambda[j]
    }
  }
  Ult.claim
  AccY.reserve<-c(rep(0,n.origin))
  AccY.reserve[1:n.origin]<-Ult.claim[1:n.origin]-diag(data.cumul[1:n.origin,n.dev-(1:n.origin)+1])[1:n.origin]
  return(AccY.reserve)
}
data.cumul<-data.cumul.fun(data)
claims.cumul <- as.vector(data.cumul)
cl.lambda<- cl.lambda.fun(data.cumul)
cl.AccY.reserve<-Chain.ladder.reserve(data)
cl.total.reserve<-sum(cl.AccY.reserve)
cl.AccY.reserve # Reserve for each origin year
cl.total.reserve # Overall reserve
cl.reserve<-c(cl.AccY.reserve[2:n.origin],cl.total.reserve)
#
D.fit <- data.cumul
D.fit[,] <- NA
for (i in 1:n.origin) {
  for (j in n.dev:1) {
    D.fit[i,n.dev-i+1] <- data.cumul[i,n.dev-i+1]
    D.fit[i,j-1] <- D.fit[i,j]/cl.lambda[j]
  }}
D.fit
C.fit <- D.fit
C.fit[,1] <- D.fit[,1]
C.fit[,2:n.dev] <- D.fit[,2:n.dev]-D.fit[, (2:n.dev)-1]
C.fit
D.fut<-matrix(NA, nrow(data), ncol(data))
C.fut<-matrix(NA, nrow(data), ncol(data))
for (i in 2:n.origin) {
  D.fut[i,n.dev-i+1] <- data.cumul[i,n.dev-i+1]
  for (j in (n.dev-i+2):n.dev) {
    D.fut[i,j]<-D.fut[i,j-1]*cl.lambda[j]
  }
}
C.fut<-D.fut
C.fut[,1]<-D.fut[,1]
C.fut[,2:n.dev]<-D.fut[,2:n.dev]-D.fut[, (2:n.dev)-1]
C.fut[i,n.dev-i+1]<-NA
res.P <- C.fit # C.fit := m_i,j
res.P <- (data-C.fit)/sqrt(C.fit)
Phi <- sum(res.P[!is.na(res.P)]^2)/(0.5*n.origin*(n.origin+1)-2*n.origin+1)
res.P.scaled <- res.P/sqrt(Phi)
rezidiju.scaled.data <- data.frame(res.P.scaled.vec=as.vector(res.P.scaled), origin=origin, delay=delay)
rezidiju.scaled.data
acf(rezidiju.scaled.data$res.P.scaled[!is.na(rezidiju.scaled.data$res.P.scaled)],
    main="", xlab="",ylab="")
plot(rezidiju.scaled.data$delay,rezidiju.scaled.data$res.P.scaled.vec,type= "p",axes=F)
axis(1, axTicks(1),format(axTicks(1), scientific = F))
axis(2, axTicks(2),format(axTicks(2), scientific = F))
plot(rezidiju.scaled.data$origin,rezidiju.scaled.data$res.P.scaled.vec,type= "p",axes=F)
axis(1, axTicks(1),format(axTicks(1), scientific = F))
axis(2, axTicks(2),format(axTicks(2), scientific = F))
plot(C.fit,res.P.scaled,type="p",axes=F,xlab="",ylab="")
axis(1, axTicks(1),format(axTicks(1), scientific = F))
axis(2, axTicks(2),format(axTicks(2), scientific = F))

Bootstrap1 <- function (b) {
  res.P.but<-sample(dati,replace=T,size=0.5*n.origin*(n.origin+1))
  C.but <-res.P.but*sqrt(C.fit[!is.na(C.fit)]+C.fit[!is.na(C.fit)])
  pseudo.C<-C.fit
  k<-1
  for (j in 1:n.dev) {

```

```

    for (i in 1:(n.origin-j+1)) {
      pseudo.C[i,j]<-C.but[k]
      k<-k+1
    } }
  AccY.reserve.b <- Chain.ladder.reserve(pseudo.C)
  total.reserve.b<-sum(AccY.reserve.b)
  bootstrap1.sim<-c(AccY.reserve.b[2:n.origin],total.reserve.b)
  return(bootstrap1.sim)
}
dati<-res.P[!is.na(res.P)]
B<-10000
bootstrap1.sim <- lapply (1:B, Bootstrap1)
AccY.reserve.sim<-sapply(1:n.origin, function(i) { sapply(1:B,function(b) {bootstrap1.sim[[b]][i]})})
mean.boot.reserves<-sapply(1:n.origin, function(i) {mean(AccY.reserve.sim[,i])} )
se.boot.reserves<-sapply(1:n.origin, function(i) { sqrt((sum((AccY.reserve.sim[,i]-cl.reserve[i])^2))/B) } )
sep.boot.reserves<-sqrt( Phi*cl.reserve + 0.5*n.origin*(n.origin+1)/
(0.5*n.origin*(n.origin+1)-2*n.origin+1)* se.boot.reserves^2 )
conf.int.fun <- function(a,s,n,p) {
  error <- qnorm((1+p)/2)*s/sqrt(n)
  left <- a-error
  right <- a+error
  return(c(left,right))
}
conf.int <- sapply(1:n.origin, function(i) {conf.int.fun(cl.reserve[i],sep.boot.reserves[i],1,0.95)} )
Bootstrap.result1<- data.frame( row.names =c(paste("Year", 1:n.origin, sep = ""),"Overall"),
  Reserve=c("-",round(cl.reserve)), Bootstrap.mean.reserve=c("-",round(mean.boot.reserves)),
  Prediction.error=c("-",round(sep.boot.reserves)), upper.95=c("-",round(conf.int[2,])) )
Bootstrap.result1
hist.fun <- function(dati,data.title) {
  n<-length(dati)
  d.sd<-sd(dati)
  h<-3.49*d.sd*n^(-1/3)
  m<-round((max(dati)-min(dati))/h)
  xx<-seq(min(dati),max(dati),len=m)
  hist(dati,breaks=xx,prob=F,xlab="",ylab="",main="",axes=F)
  axis(1, axTicks(1),format(axTicks(1), scientific = F))
  axis(2, axTicks(2),format(axTicks(2), scientific = F))
}
hist.fun(AccY.reserve.sim[,n.origin],"Summārā rezerve")
#
res.P[1,n.dev] <- NA
res.P[n.origin,1] <- NA
dati<-res.P[!is.na(res.P)]
B<-10000
bootstrap1.sim <- lapply (1:B, Bootstrap1)
AccY.reserve.sim<-sapply(1:n.origin, function(i) { sapply(1:B,function(b) {bootstrap1.sim[[b]][i]})})
mean.boot.reserves<-sapply(1:n.origin, function(i) {mean(AccY.reserve.sim[,i])} )
se.boot.reserves<-sapply(1:n.origin, function(i) { sqrt((sum((AccY.reserve.sim[,i]-cl.reserve[i])^2))/B) } )
sep.boot.reserves<-sqrt( Phi*cl.reserve + 0.5*n.origin*(n.origin+1)/
(0.5*n.origin*(n.origin+1) - 2*n.origin+1) * se.boot.reserves^2 )
conf.int.fun <- function(a,s,n,p) {
  error <- qnorm((1+p)/2)*s/sqrt(n)
  left <- a-error
  right <- a+error
  return(c(left,right))
}
conf.int <- sapply(1:n.origin, function(i) {conf.int.fun(cl.reserve[i],sep.boot.reserves[i],1,0.95)} )
Bootstrap.result2<- data.frame( row.names =c(paste("Year", 1:n.origin, sep = ""),"Overall"),
  Reserve=c("-",round(cl.reserve)), Bootstrap.mean.reserve=c("-",round(mean.boot.reserves)),
  Prediction.error=c("-",round(sep.boot.reserves)), upper.95=c("-",round(conf.int[2,])) )
Bootstrap.result2
hist.fun(AccY.reserve.sim[,n.origin],"Summārā rezerve")

```

A3. Gamma vispārinātais lineārais modelis un butstraps

```

data <- read.table("data",header=F)
data
data <- as.matrix(data)
claims <- as.vector(data)
n.origin <- nrow(data)
n.dev <- ncol(data)
origin <- factor(row <- rep(1:n.origin, n.dev))
dev <- factor(col <- rep(1:n.dev, each=n.origin))
claim.data <- data.frame(claims=claims, origin=origin, dev=dev)
head(claim.data)
Gamma<-function (link = "inverse")
{
  linktemp <- substitute(link)
  if (!is.character(linktemp)) {
    linktemp <- deparse(linktemp)
    if (linktemp == "link") {
      warning("use of Gamma(link=link) is deprecated\n",
              domain = NA)
      linktemp <- eval(link)
      if (!is.character(linktemp) || length(linktemp) !=
          1)
        stop("'link' is invalid", domain = NA)
    }
  }
  okLinks <- c("inverse", "log", "identity")
  if (linktemp %in% okLinks)
    stats <- make.link(linktemp)
  else if (is.character(link))
    stats <- make.link(link)
  else {
    if (inherits(link, "link-glm")) {
      stats <- link
      if (!is.null(stats$name))
        linktemp <- stats$name
    }
    else {
      stop(gettextf("link \"%s\" not available for gamma family; available links are %s",
                    linktemp, paste(sQuote(okLinks), collapse = ", ")),
           domain = NA)
    }
  }
}
variance <- function(mu) mu^2
validmu <- function(mu) all(mu > 0)
dev.resids <- function(y, mu, wt) wt*(y-mu)^2/(mu^2)
aic <- function(y, n, mu, wt, dev) {
  n <- sum(wt)
  disp <- dev/n
  -2 * sum(dgamma(y, 1/disp, scale = mu * disp, log = TRUE) *
           wt) + 2
}
initialize <- expression({
  n <- rep.int(1, nobs)
  mustart <- rep(mean(y), length(y))
})
simfun <- function(object, nsim) {
  wts <- object$prior.weights
  if (any(wts != 1))
    message("using weights as shape parameters")
  ftd <- fitted(object)
  shape <- MASS::gamma.shape(object)$alpha * wts
  rgamma(nsim * length(ftd), shape = shape, rate = shape/ftd)
}
structure(list(family = "Gamma", link = linktemp, linkfun = stats$linkfun,
              linkinv = stats$linkinv, variance = variance, dev.resids = dev.resids,
              aic = aic, mu.eta = stats$mu.eta, initialize = initialize,

```

```

        validmu = validmu, valideta = stats$valideta, simulate = simfun),
        class = "family")
}
# Fit model
model <- glm(claims ~ origin + dev, family = Gamma(link = "log"),subset=!is.na(claims), data=claim.data)
summary(model)
# Extract useful info from the model
coef <- model$coefficients # Get coefficients
disp <- summary(model)$dispersion # Get dispersion parameter
cov.param <- disp * summary(model)$cov.unscaled # Get covariance matrix of parameters
cc<-coef[1]
alpha<-c(0,coef[2:n.origin])
beta<-c(0,coef[(n.origin+1):(2*n.origin-1)])
n.fut.points <- length(claims[is.na(claims)])
fut.design <- matrix(0, nrow = n.fut.points, ncol=length(coef))
fut.points <- claims
fut.points[is.na(claims)] <- 0
fut.points[is.na(claims)] <- 1:n.fut.points
for(p in 1:n.fut.points){
  fut.design[p, 1] <- 1
  fut.design[p, 1 +
as.numeric(origin[match(p, fut.points)]) - 1] <- 1
  fut.design[p, 1 + (n.origin-1) +
as.numeric(dev[match(p, fut.points)]) - 1] <- 1
}
fitted.values <- diag(as.vector(exp(fut.design %*% coef)))
total.reserve <- sum(fitted.values)
total.reserve2 <- sum(fitted.values^2)
data.fitted <-data
AccY.reserve <- c(rep(0,n.origin))
AccY.reserve2 <- c(rep(0,n.origin))
for (i in 1:n.origin) {
  for (j in 1:n.dev) {
    if(!is.na(data[i,j])) {data.fitted[i,j]<-data[i,j]}
    else {data.fitted[i,j]<- exp(coef[1]+coef[i]+coef[n.origin+j-1])
      AccY.reserve[i]<- AccY.reserve[i]+ data.fitted[i,j]
      AccY.reserve2[i]<- AccY.reserve2[i]+ data.fitted[i,j]^2
    }
  }
}
}

cov.pred <- fut.design %*% cov.param %*% t(fut.design)
cov.fitted <- fitted.values %*% cov.pred %*% fitted.values
total.rmse <- sqrt(disp*total.reserve2+sum(cov.fitted))
total.predictionerror <- round(100*total.rmse/total.reserve) #
cov.index<-matrix(0, nrow = n.dev, ncol=n.origin)
cov.index[2,1]<-n.fut.points-n.origin+2
for (i in 3:n.origin) {
  cov.index[i,1]<-n.fut.points-n.origin+i
  x<-cov.index[i,1]
  for (k in 3:i-1) {
    x<-x-(n.dev-k+1)
    cov.index[i,k]<-x
    x<-cov.index[i,k]
  }
}
}
AccY.rmse <- c(rep(0,n.origin))
AccY.rmse.fun <- function(i) { sqrt(disp*AccY.reserve2[i]+sum(cov.fitted[cov.index[i,1],cov.index[i,1]])) }
AccY.rmse <- sapply(2:n.origin,AccY.rmse.fun)
AccY.predictionerror <- c(rep(0,n.origin))
AccY.predictionerror <- round(100*AccY.rmse[1:(n.origin-1)]/AccY.reserve[2:n.origin])
##### Gamma model; reserve result
Gamma.model.result<- data.frame(row.names =c(paste("Year", 1:n.origin, sep = ""),"Overall"),
  Reserve=c(round(AccY.reserve),round(total.reserve)), Prediction.error=c("-",round(AccY.rmse),
round(total.rmse)),
  Prediction.error.percent=c("-",AccY.predictionerror,total.predictionerror))
Gamma.model.result
gamma.reserve<-c(AccY.reserve[2:n.origin],total.reserve) #10

C.fit <-data
for (i in 1:n.origin) {
  for (j in 1:n.dev) {
    if(is.na(data[i,j])) {C.fit[i,j]<-data[i,j]}
    else {C.fit[i,j]<- exp(cc+alpha[i]+beta[j]) }
  }
}

```



```

}
}

C.fut <-matrix(NA, nrow(data), ncol(data))
for (i in 1:n.origin) {
  for (j in 1:n.dev) {
    if(is.na(data[i,j])) {C.fut[i,j]<-exp(cc+alpha[i]+beta[j])}
    else {C.fut[i,j]<- NA}
  }
}
}
res.P <- C.fit
res.P <- (data-C.fit)/sqrt(C.fit^2)
res.P
Phi <- sum(res.P[!is.na(res.P)]^2)/(0.5*n.origin*(n.origin+1)-2*n.origin+1)
dati<-res.P
Gamma.model.reserve.fun<-function(dati) {
  claims <- as.vector(dati)
  claim.data <- data.frame(claims=claims, origin=origin, dev=dev)
  # Fit model
  model <- glm(claims ~ origin + dev, family = Gamma(link = "log"),subset=!is.na(claims), data=claim.data)
  summary(model)
  coef <- model$coefficients # Get coefficients
  disp <- summary(model)$dispersion # Get dispersion parameter
  cov.param <- disp * summary(model)$cov.unscaled # Get covariance matrix of parameters
  cc<-coef[1]
  alpha<-c(0,coef[2:n.origin])
  beta<-c(0,coef[(n.origin+1):(2*n.origin-1)])
  data.fitted <-dati
  AccY.reserve <- c(rep(0,n.origin))
  AccY.reserve2 <- c(rep(0,n.origin))
  for (i in 1:n.origin) {
    for (j in 1:n.dev) {
      if(!is.na(data[i,j])) {data.fitted[i,j]<-data[i,j]} #not NA (known)
      else {data.fitted[i,j]<- exp(cc+alpha[i]+beta[j])}
      AccY.reserve[i]<- AccY.reserve[i]+ data.fitted[i,j]
      AccY.reserve2[i]<- AccY.reserve2[i]+ data.fitted[i,j]^2
    }
  }
  }
  return(AccY.reserve)
}
}
Bootstrap1 <- function (b) {
  res.P.but<-sample(dati,replace=T,size=0.5*n.origin*(n.origin+1))
  C.but <-res.P.but*sqrt((C.fit[!is.na(C.fit)]^2)+C.fit[!is.na(C.fit)])
  pseudo.C<-C.fit
  k<-1
  for (j in 1:n.dev) {
    for (i in 1:(n.origin-j+1)) {
      pseudo.C[i,j]<-C.but[k]
      k<-k+1
    }
  }

  AccY.reserve.b <- Gamma.model.reserve.fun(pseudo.C) # 10
  total.reserve.b<-sum(AccY.reserve.b)
  bootstrap1.sim<-c(AccY.reserve.b[2:n.origin],total.reserve.b)
  return(bootstrap1.sim)
}
dati<-res.P[!is.na(res.P)]
B<-10000
bootstrap1.sim <- lapply (1:B, Bootstrap1)
AccY.reserve.sim<-sapply(1:n.origin, function(i) { sapply(1:B,function(b) {bootstrap1.sim[[b]][i]})})
mean.boot.reserves<-sapply(1:n.origin, function(i) {mean(AccY.reserve.sim[,i])} )
se.boot.reserves<-sapply(1:n.origin, function(i) { sqrt((sum((AccY.reserve.sim[,i]-gamma.reserve[i])^2))/B) } )
sep.boot.reserves<-sqrt( Phi*gamma.reserve + 0.5*n.origin*(n.origin+1)/(0.5*n.origin*(n.origin+1)-2*n.origin+1)
* se.boot.reserves^2 )
conf.int.fun <- function(a,s,n,p) {
  error <- qnorm(p)*s/sqrt(n)
  left <- a-error
  right <- a+error
  return(c(left,right))
}
}
conf.int <- sapply(1:n.origin, function(i) {conf.int.fun(gamma.reserve[i],sep.boot.reserves[i],1,0.9) } )
Bootstrap.result1<- data.frame( row.names =c(paste("Year", 1:n.origin, sep = ""),"Overall"),
  Reserve=c("-",round(gamma.reserve)), Bootstrap.mean.reserve=c("-",round(mean.boot.reserves)),

```

```

Prediction.error=c("-",round(sep.boot.reserves)), upper.95=c("-",round(conf.int[2,])) )
Bootstrap.result1
hist.fun <- function(dati,data.title) {
  n<-length(dati)
  d.sd<-sd(dati)
  h<-3.49*d.sd*n^(-1/3)
  m<-round((max(dati)-min(dati))/h)
  xx<-seq(min(dati),max(dati),len=m)
  hist(dati,breaks=xx,prob=F,xlab="",ylab="",main="",axes=F)
  axis(1, axTicks(1),format(axTicks(1), scientific = F))
  axis(2, axTicks(2),format(axTicks(2), scientific = F))
}
hist.fun(AccY.reserve.sim[,10],"Summārā rezerve")

```

A4. Chain-ladder modeļa Normālā aproksimācija

```

data <- read.table("data.txt",header=F)
data <- as.matrix(data)
data
claims <- as.vector(data)
n.origin <- nrow(data)
n.dev <- ncol(data)
origin <- factor(row <- rep(1:n.origin, n.dev))
dev <- factor(col <- rep(0:(n.dev-1), each=n.origin))
data.cumul<-matrix(NA, nrow(data), ncol(data))
for (i in 1:n.origin) {
  for (j in 1:n.dev) {
    data.cumul[i,j]<-sum(data[i,1:j])
  }
}
data.cumul
claims.cumul <- as.vector(data.cumul)
ind.factor<-matrix(NA, nrow(data), ncol(data))
for (i in 1:n.origin) {
  for (j in 2:n.dev) {
    ind.factor[i,j]<-data.cumul[i,j]/data.cumul[i,j-1]
  }
}
ind.factor
f<-as.vector(ind.factor)
Weights<-matrix(NA, nrow(data), ncol(data))
disp.param<-c(rep(1,n.dev))
for (i in 1:n.origin) {
  for (j in 2:n.dev) {
    Weights[i,j]<-data.cumul[i,j-1]/disp.param[j]
  }
}
W<-as.vector(Weights)
claim.data <- data.frame(dev=dev, f=f, W=W, subset=!is.na(f))
head(claim.data)
modell <- glm(f ~ dev, family = gaussian(),subset=!is.na(f)&!is.na(W),weights=W, data=claim.data)
summary(modell)
coef1 <- modell$coefficients
disp1 <- summary(modell)$dispersion
cov.param1 <- disp1 * summary(modell)$cov.unscaled
fitted.values.f<-matrix(NA, nrow(data), ncol(data))
for (i in 1:n.origin) {
  fitted.values.f[i,2]<-coef1[1]+0
  for (j in 3:n.dev) {
    fitted.values.f[i,j]<-coef1[1]+coef1[j-1]
  }
}
}
fitted.values.f
lambda<-c(rep(0,n.dev))
lambda<-fitted.values.f[1,]
lambda
res.sq<-matrix(NA, nrow(data), ncol(data))

```

```

for (i in 1:n.origin) {
  for (j in 2:n.dev) {
    res.sq[i,j]<-data.cumul[i,j-1]*(ind.factor[i,j]-fitted.values.f[i,j])^2
  }
}
res.sq
r<-as.vector(res.sq)
r
residuals.data <- data.frame(dev=dev, r=r, subset=!is.na(r))
residuals.data
model2 <- glm(r ~ dev, family = gaussian(),subset=!is.na(r), data=residuals.data)
summary(model2)
coef2 <- model2$coefficients
disp2 <- summary(model2)$dispersion
cov.param2 <- disp2 * summary(model2)$cov.unscaled
fitted.values.r<-matrix(NA, nrow(data), ncol(data))
for (i in 1:n.origin) {
  fitted.values.r[i,2]<- coef2[1]+0
  for (j in 3:n.dev) {
    fitted.values.r[i,j]<-coef2[1]+coef2[j-1]
  }
}
fitted.values.r
Phi<-c(rep(0,n.dev))
Phi<-fitted.values.r[1,]
Phi[n.dev]<-Phi[n.dev-1]
Phi
Weights2<-matrix(NA, nrow(data), ncol(data))
for (i in 1:n.origin) {
  for (j in 2:n.dev) {
    Weights2[i,j]<-data.cumul[i,j-1]/Phi[j]
  }
}
W2<-as.vector(Weights2)
claim.data <- data.frame(dev=dev, f=f, W2=W2, subset=!is.na(f))
head(claim.data)
model3 <- glm(f ~ dev, family = gaussian(),subset=!is.na(f),weights=W2, data=claim.data)
summary(model3)
coef3 <- model3$coefficients
disp3 <- summary(model3)$dispersion
cov.param3 <- disp3 * summary(model3)$cov.unscaled
D<-matrix(NA, nrow(data), ncol(data))
for (i in 1:n.origin) {
  for (j in 1:n.dev) {
    D[i,j]<-sum(data[i,1:j])
  }
}
data.cumul.tr<-D
claims.cumul.tr <- as.vector(data.cumul.tr)
fitted.cumul<-c(rep(0,length(claims.cumul.tr[is.na(claims.cumul.tr)])))
k<-1
for (j in 2:n.dev) {
  for (i in (n.origin-j+2):n.origin) {
    D[i,j]<-D[i,j-1]*lambda[j]
    fitted.cumul[k]<-D[i,j-1]*lambda[j]
    k<-k+1
  }
}
D
fitted.cumul
R<-c(rep(0,n.origin))
for (i in 1:n.origin) {
  R[i]<-D[i,n.dev]-D[i,n.dev-i+1]
}
total.reserve<-sum(R)
total.reserve
Ult.claim<-c(rep(0,n.origin))
Ult.claim[1]<-data.cumul[1,n.dev]
for (i in 2:n.origin) {
  Ult.claim[i]<-data.cumul[i,n.dev-i+1]
  for (j in (n.dev-i+2):n.dev) {
    Ult.claim[i]<-Ult.claim[i]*lambda[j]
  }
}
}

```

```

Ult.claim
AccY.reserve<-c(rep(0,n.origin))
AccY.reserve[1:n.origin]<-Ult.claim[1:n.origin]-diag(data.cumul[1:n.origin,n.dev-(1:n.origin)+1])[1:n.origin]
total.reserve<-sum(AccY.reserve)
AccY.reserve
total.reserve
cov.lambda<-matrix(NA, nrow=length(lambda), ncol=length(lambda))
cov.lambda[2,2]<-round(cov.param3[1,1],6)
for (i in 3:length(lambda)) {
  cov.lambda[i,2]<-round(cov.param3[1,1]+cov.param3[i-1,1],6)
  for (j in 3:length(lambda)) {
    cov.lambda[i,j]<-round(cov.param3[1,1]+cov.param3[1,j-1]+cov.param3[i-1,1]+cov.param3[i-1,j-1],6)
  }
}
cov.lambda
var.lambda<-diag(cov.lambda)
var.lambda
se.lambda<-sqrt(var.lambda)
se.lambda
n<-length(lambda)
m1<-matrix(rev(lambda[!is.na(lambda)]),n-1,n-1)
diag(m1)<-rev(Phi[!is.na(Phi)])
m1
m2<-matrix(NA,n-1,n-1)
m3<-matrix(NA,n-1,n-1)
m2[upper.tri(m1)]<-m1[upper.tri(m1)]^2
m2[lower.tri(m1)]<-m1[lower.tri(m1)]
diag(m2)<-diag(m1)
m2
for (i in 1:(n-1)) {
  for (j in 1:(n-1)) {
    m3[i,j]<-prod(m2[1:i,j])
  }
}
m3
X<-c(rep(0,n-1))
X<-sapply(1:(n-1),function(i) {sum(m3[i,1:i])})
process.var<-c(rep(0,n.origin))
process.var<-diag(data.cumul[1:n.origin,n.dev-(1:n.origin)+1])[2:n.origin]*X
process.var
total.process.var<-sum(process.var)
var.lambda2<-c(rep(0,n))
var.lambda2[2]<-var.lambda[n]
var.prev.step<-var.lambda2[2]
var.Din<-c(rep(NA,n.origin))
var.Din[2]<-data.cumul[2,n-2+1]^2*var.lambda[n]
for (i in 3:n) {
  var.lambda2[i]<-lambda[n-i+2]^2*var.prev.step+prod(lambda[(n-i+3):n])^2*var.lambda[n-i+2]+
var.prev.step*var.lambda[n-i+2]
var.Din[i]<-data.cumul[i,n-i+1]^2*var.lambda2[i]
var.prev.step<-var.lambda2[i]
}
cov.Din<-matrix(NA,n.origin,n.dev)
for (i in 2:n) {
  for (j in 2:n) {
    if(j>i) {
      cov.Din[i,j]<-data.cumul[i,n-i+1]*var.lambda2[i]*data.cumul[j,n-j+1]*prod(lambda[(n-j+2):(n-i+1)])
      cov.Din[j,i]<-cov.Din[i,j]
    }
    if(i==j) {cov.Din[i,j]<-var.Din[i]}
  }
}
cov.Din
var.total.reserve<-sum(cov.Din[!is.na(cov.Din)])
var.total.reserve
var.Din<-var.Din[!is.na(var.Din)]
prediction.variance<-var.Din+process.var
total.prediction.variance<-sum(var.total.reserve+total.process.var)
prediction.error<-sqrt(prediction.variance)
total.prediction.error<-sqrt(total.prediction.variance)
prediction.error.percent<-round(100*prediction.error[1:(n.origin-1)]/AccY.reserve[2:n.origin])
total.prediction.error.percent<-round(100*total.prediction.error/total.reserve)
lambda.result<- data.frame(row.names=paste("lambda", 2:n.dev, sep = ""),Estimate=lambda[2:n.dev],
Standard.error=se.lambda[2:n.dev])

```

```

lambda.result
Phi.result<- data.frame(row.names=paste("Phi", 2:n.dev, sep = ""),Estimate=Phi[2:n.dev])
Phi.result
NormalApprox.model.result<- data.frame(row.names =c(paste("Year", 1:n.origin, sep = ""),"Overall"),
Reserve=c(round(AccY.reserve),round(total.reserve)),Process.variance=
c("-",round(process.var),round(total.process.var)),
Estimation.variance=c("-",round(var.Din),round(var.total.reserve)),
Prediction.variance=c("-",round(prediction.variance),
round(total.prediction.variance)),
Prediction.error=c("-",round(prediction.error),round(total.prediction.error)),
Prediction.error.percent=c("-",round(prediction.error.percent),
round(total.prediction.error.percent)))
NormalApprox.model.result
D<-matrix(NA, nrow(data), ncol(data))
for (i in 1:n.origin) {
  for (j in 1:n.dev) {
    D[i,j]<-sum(data[i,1:j])
  }
}
data.cumul.tr<-D
claims.cumul.tr <- as.vector(data.cumul.tr)
fitted.cumul<-c(rep(0,length(claims.cumul[is.na(claims.cumul.tr)])))
k<-1
for (j in 2:n.dev) {
  for (i in (n.origin-j+2):n.origin) {
    D[i,j]<-D[i,j-1]*lambda[j]
    fitted.cumul[k]<-D[i,j-1]*lambda[j]
    k<-k+1
  }
}
D
fitted.cumul
R<-c(rep(0,n.origin))
for (i in 1:n.origin) {
  R[i]<-D[i,n.dev]-D[i,n.dev-i+1]
}
total.reserve<-sum(R)
R
total.reserve
AccY.predictionerror <- c(rep(0,n.origin))
AccY.predictionerror <- round(100*AccY.rmse[1:(n.origin-1)]/AccY.reserve[2:n.origin])

```

A5. Log-Normālais modelis

```

data <- read.table("data.txt",header=F)
data <- as.matrix(data)
data
log.data<-log(data)
log.data
claims <- as.vector(log.data)
n.origin <- nrow(data)
n.dev <- ncol(data)
origin <- factor(row <- rep(1:n.origin, n.dev))
dev <- factor(col <- rep(1:n.dev, each=n.origin))
claim.data <- data.frame(claims=claims, origin=origin, dev=dev)
head(claim.data)
model <- lm(claims ~ origin + dev,subset=!is.na(claims), data=claim.data)
summary(model)
coef <- model$coefficients
exp(model$fitted.values)
disp <- summary(model)$sigma^2
cov.param <- disp * summary(model)$cov.unscaled
disp
cc<-coef[1]
alpha<-c(0,coef[2:n.origin])

```

```

beta<-c(0,coef[(n.origin+1):(2*n.origin-1)])
data.fitted <-data
median.AccY.reserve <- c(rep(0,n.origin))
for (i in 1:n.origin) {
  for (j in 1:n.dev) {
    if(!is.na(data[i,j])) {data.fitted[i,j]<-data[i,j]}
    else {data.fitted[i,j]<- exp(coef[1]+coef[i]+coef[n.origin+j-1])
          median.AccY.reserve[i]<- median.AccY.reserve[i]+ data.fitted[i,j]
        }
  }
}
median.AccY.reserve
median.total.reserve<-sum(median.AccY.reserve)
n.fut.points <- length(claims[is.na(claims)])
fut.design <- matrix(0, nrow = n.fut.points, ncol=length(coef))
fut.points <- claims
fut.points[is.na(claims)] <- 0
fut.points[is.na(claims)] <- 1:n.fut.points
for(p in 1:n.fut.points){
  fut.design[p, 1] <- 1
  fut.design[p, 1 +
as.numeric(origin[match(p, fut.points)]) - 1] <- 1
  fut.design[p, 1 + (n.origin-1) +
as.numeric(dev[match(p, fut.points)]) - 1] <- 1
}
fitted.values <- diag(as.vector(exp(fut.design %*% coef)))
median.total.reserve <- sum(fitted.values)
cov.pred <- fut.design %*% cov.param %*% t(fut.design)
diag(cov.pred)<-diag(cov.pred)+disp
cov.fitted <- fitted.values %*% (exp(cov.pred)-1) %*% fitted.values
total.rmse <- sqrt(sum(cov.fitted))
total.predictionerror <- round(100*total.rmse/median.total.reserve)
cov.index<-matrix(0, nrow = n.dev, ncol=n.origin)
cov.index[2,1]<-n.fut.points-n.origin+2
for (i in 3:n.origin) {
  cov.index[i,1]<-n.fut.points-n.origin+i
  x<-cov.index[i,1]
  for (k in 3:i-1) {
    x<-x-(n.dev-k+1)
    cov.index[i,k]<-x
    x<-cov.index[i,k]
  }
}
}
AccY.rmse <- c(rep(0,n.origin))
AccY.rmse.fun <- function(i) { sqrt(sum(cov.fitted[cov.index[i,],cov.index[i,]])) }
AccY.rmse <- sapply(2:n.origin,AccY.rmse.fun)
AccY.predictionerror <- c(rep(0,n.origin))
AccY.predictionerror <- round(100*AccY.rmse[1:(n.origin-1)]/median.AccY.reserve[2:n.origin])
LogNormal.model.median.result<- data.frame(row.names =c(paste("Year", 1:n.origin, sep = ""),
"Overall"), Reserve=c(round(median.AccY.reserve),round(median.total.reserve)),
Prediction.error=c("-",round(AccY.rmse),round(total.rmse)),
Prediction.error.percent=c("-",AccY.predictionerror,total.predictionerror))
LogNormal.model.median.result
eta <-log.data
var.eta<-matrix(0,n.origin,n.dev)
mean.AccY.reserve <- c(rep(0,n.origin))
for (i in 1:n.dev) {
  for (j in 1:n.dev) {
    if(!is.na(data[i,j])) {eta[i,j]<-log.data[i,j]}
    else {eta[i,j]<- coef[1]+coef[i]+coef[n.origin+j-1]
          var.eta[i,j]<-cov.param[1,1]+cov.param[i,i]
+cov.param[n.origin+j-1,n.origin+j-1]+2*cov.param[1,i]
+2*cov.param[1,n.origin+j-1]+2*cov.param[i,n.origin+j-1]
        }
  }
}
}
var.eta
data.fitted <-data
for (i in 1:n.origin) {
  for (j in 1:n.dev) {
    if(!is.na(data[i,j])) {data.fitted[i,j]<-data[i,j]}
    else {data.fitted[i,j]<- exp(eta[i,j]+0.5*(var.eta[i,j]+disp))
          mean.AccY.reserve[i]<- mean.AccY.reserve[i]+ data.fitted[i,j]
        }
  }
}
}

```

```

}
mean.AccY.reserve
mean.total.reserve<-sum(mean.AccY.reserve)
n.fut.points <- length(claims[is.na(claims)])
fut.design <- matrix(0, nrow = n.fut.points, ncol=length(coef))
fut.points <- claims
fut.points[!is.na(claims)] <- 0
fut.points[is.na(claims)] <- 1:n.fut.points
for(p in 1:n.fut.points){
fut.design[p, 1] <- 1
fut.design[p, 1 +
as.numeric(origin[match(p, fut.points)]) - 1] <- 1
fut.design[p, 1 + (n.origin-1) +
as.numeric(dev[match(p, fut.points)]) - 1] <- 1
}
fv<-c(rep(0,n.fut.points))
k<-1
for (j in 2:n.dev) {
  for (i in (n.origin-j+2):n.origin) {
    fv[k] <- data.fitted[i,j]
    k<-k+1
  } }
fitted.values <- diag(as.vector(fv))
mean.total.reserve <- sum(fitted.values)
cov.pred <- fut.design %*% cov.param %*% t(fut.design)
diag(cov.pred)<-diag(cov.pred)+disp
cov.fitted <- fitted.values %*% (exp(cov.pred)-1) %*% fitted.values
total.rmse <- sqrt(sum(cov.fitted))
total.predictionerror <- round(100*total.rmse/mean.total.reserve)
cov.index<-matrix(0, nrow = n.dev, ncol=n.origin)
cov.index[2,1]<-n.fut.points-n.origin+2
for (i in 3:n.origin) {
  cov.index[i,1]<-n.fut.points-n.origin+i
  x<-cov.index[i,1]
  for (k in 3:i-1) {
    x<-x-(n.dev-k+1)
    cov.index[i,k]<-x
    x<-cov.index[i,k]
  }
}
AccY.rmse <- c(rep(0,n.origin))
AccY.rmse.fun <- function(i) { sqrt(sum(cov.fitted[cov.index[i,],cov.index[i,]])) }
AccY.rmse <- sapply(2:n.origin,AccY.rmse.fun)
AccY.predictionerror <- c(rep(0,n.origin))
AccY.predictionerror <- round(100*AccY.rmse[1:(n.origin-1)]/mean.AccY.reserve[2:n.origin])
LogNormal.model.mean.result<- data.frame(row.names =
c(paste("Year", 1:n.origin, sep = ""),"Overall"),
  Reserve=c(round(mean.AccY.reserve),round(mean.total.reserve)), Prediction.error=
c("-",round(AccY.rmse),round(total.rmse)),
  Prediction.error.percent=c("-",AccY.predictionerror,total.predictionerror))
LogNormal.model.mean.result
var.eta<-matrix(NA,n.origin,n.dev)
var.eta[1,1]<-cov.param[1,1]
var.eta[1,2:n.dev]<-sapply(2:n.dev, function(k) {var.eta[1,k]<-cov.param[1,1]+
cov.param[n.origin+k-1,n.origin+k-1]+2*cov.param[1,n.origin+j-1]})
var.eta[2:n.origin,1]<-sapply(2:n.origin, function(k) {var.eta[k,1]<-cov.param[1,1]+
cov.param[k,k]+2*cov.param[1,k]})

for (i in 2:n.origin) {
  for (j in 2:n.dev) {
    if(is.na(data[i,j])) {eta[i,j]<-log.data[i,j]}
    else {eta[i,j]<- cc+alpha[i]+beta[j]
      var.eta[i,j]<-cov.param[1,1]+cov.param[i,i]
+cov.param[n.origin+j-1,n.origin+j-1]+2*cov.param[1,i]
+2*cov.param[1,n.origin+j-1]+2*cov.param[i,n.origin+j-1]
    }
  }
}
var.eta
C.fit <-data
for (i in 1:n.origin) {
  for (j in 1:n.dev) {
    if(is.na(data[i,j])) {C.fit[i,j]<-data[i,j]}
    else {C.fit[i,j]<- exp(eta[i,j]+0.5*(var.eta[i,j]+disp))
  }
}

```

```

}
}
res.P <- C.fit
res.P <- (log.data-log(C.fit))
res.P
res.P.scaled <- res.P/sqrt(disps)
rezidiju.scaled.data <- data.frame(res.P.scaled.vec=as.vector(res.P.scaled), origin=origin, dev=dev)
rezidiju.scaled.data
plot(rezidiju.scaled.data$dev, rezidiju.scaled.data$res.P.scaled.vec, type="p", axes=F)
axis(1, axTicks(1), format(axTicks(1), scientific = F))
axis(2, axTicks(2), format(axTicks(2), scientific = F))
plot(rezidiju.scaled.data$origin, rezidiju.scaled.data$res.P.scaled.vec, type="p", axes=F)
axis(1, axTicks(1), format(axTicks(1), scientific = F))
axis(2, axTicks(2), format(axTicks(2), scientific = F))
plot(log(C.fit), res.P.scaled, type="p", axes=F, xlab="", ylab="")
axis(1, axTicks(1), format(axTicks(1), scientific = F))
axis(2, axTicks(2), format(axTicks(2), scientific = F))
plot(res.P.scaled[!is.na(res.P.scaled)], type="p", axes=F, xlab="", ylab="")
axis(1, axTicks(1), format(axTicks(1), scientific = F))
axis(2, axTicks(2), format(axTicks(2), scientific = F))
acf(rezidiju.scaled.data$res.P.scaled[!is.na(rezidiju.scaled.data$res.P.scaled)],
    main="", xlab="", ylab="")

```

A6. Maka modelis

```

data <- read.table("data.txt", header=F)
data <- as.matrix(data)
data
claims <- as.vector(data)
n.origin <- nrow(data)
n.dev <- ncol(data)
origin <- factor(row <- rep(1:n.origin, n.dev))
dev <- factor(col <- rep(0:(n.dev-1), each=n.origin))
data.cumul <- matrix(NA, nrow(data), ncol(data))
for (i in 1:n.origin) {
  for (j in 1:n.dev) {
    data.cumul[i, j] <- sum(data[i, 1:j])
  }
}
data.cumul
claims.cumul <- as.vector(data.cumul)
w <- matrix(NA, nrow(data), ncol(data))
for (i in 1:n.origin) {
  for (j in 2:n.dev) {
    w[i, j] <- data.cumul[i, j-1]
  }
}
f <- matrix(NA, nrow(data), ncol(data))
f <- data.cumul/w
lambda <- c(rep(0, n.dev))
sigma2 <- c(rep(0, n.dev))
for (j in 2:n.dev) {
  lambda[j] <- sum(w[1:(n.origin-j+1), j] * f[1:(n.origin-j+1), j]) / sum(w[1:(n.origin-j+1), j])
  sigma2[j] <- 1/(n.origin-j) * sum(w[1:(n.origin-j+1), j] * (f[1:(n.origin-j+1), j] - lambda[j])^2)
}
sigma2[n.dev] = sigma2[n.dev-2] # as Mack(1994b)
Dev.factors.Var.comp.result1 <- data.frame(row.names = paste("j =", 2:n.dev, sep = ""),
Lambda = round(lambda[2:n.dev], 3),
Sigma.squared = c(round(sigma2[2:(n.dev)], 2)))
Dev.factors.Var.comp.result1
Ult.claim <- c(rep(0, n.origin))
Ult.claim[1] <- data.cumul[1, n.dev]
for (i in 2:n.origin) {
  Ult.claim[i] <- data.cumul[i, n.dev-i+1]
  for (j in (n.dev-i+2):n.dev) {

```



```

      Ult.claim[i]<-Ult.claim[i]*lambda[j]
    }
  }
  AccY.reserve<-c(rep(0,n.origin))
  for (i in 1:n.origin) {
    AccY.reserve[i]<-Ult.claim[i]-data.cumul[i,n.dev-i+1]
  }
  total.reserve<-sum(AccY.reserve)
  AccY.reserve
  total.reserve
  data.cumul.fitted <- data.cumul
  for (i in 2:n.origin) {
    data.cumul.fitted[i,n.dev-i+1] <- data.cumul[i,n.dev-i+1]
    for (j in (n.dev-i+2):n.dev) {
      data.cumul.fitted[i,j]<-data.cumul.fitted[i,j-1]*lambda[j]
    }
  }
  data.cumul.fitted

  process.var <- c(rep(0,n.origin))
  for (i in 2:n.origin) {
    process.var[i] <- Ult.claim[i]^2*sum(sigma2[(n.dev-i+1):(n.dev-1+1)]/
      (lambda[(n.dev-i+1):(n.dev-1+1)]^2* data.cumul.fitted[i,(n.dev-i+1):(n.dev-1)]))
  }
  process.var
  total.process.var<-sum(process.var)
  D <- c(rep(0,n.origin))
  for (i in 2:n.origin) {
    for (k in (n.dev-i+1):(n.dev-1)) {
      D[k] <- sum(data.cumul[1:(n.origin-k),k])
    }
  }
  D
  estimation.var <- c(rep(0,n.origin))
  for (i in 2:n.origin) {
    estimation.var[i]<-Ult.claim[i]^2*sum (sigma2[(n.dev-i+1): (n.dev-1+1)]/
      (lambda[(n.dev-i+1): (n.dev-1+1)]^2* D[(n.dev-i+1):(n.dev-1)]))
  }
  estimation.var

  prediction.variance<-process.var+estimation.var
  prediction.error<-sqrt(prediction.variance)
  prediction.error.percent<-round (100* prediction.error/AccY.reserve)

  total.prediction.variance<-0
  for (i in 2:(n.origin-1)) {
    total.prediction.variance <- total.prediction.variance + (prediction.variance[i]+
      Ult.claim[i]*sum(Ult.claim[(i+1):(n.dev)])*
        sum(2*sigma2[(n.dev-i+1):(n.dev-1+1)]/
          (lambda[(n.dev-i+1): (n.dev-1+1)]^2* D[(n.dev-i+1): (n.dev-1)])))
  }
  total.prediction.variance <- total.prediction.variance + prediction.variance [n.origin]
  total.prediction.error <- sqrt ( total.prediction.variance)

  total.prediction.error.percent <- round( 100 * total.prediction.error /total.reserve)
  total.estimation.var<-total.prediction.variance-total.process.var
  Mack.model.result1<- data.frame(row.names =c(paste("Year", 1:n.origin, sep = ""),"Overall"),
    Reserve=c("-",round (AccY.reserve[2:n.origin]), round(total.reserve)),
    Process.variance=c("-", round (process.var[2:n.origin]), round(total.process.var)),
    Estimation.variance=c("-", round(estimation.var[2:n.origin]), round(total.estimation.var)),
    Prediction.variance=c("-",round(prediction.variance[2:n.origin]), round(total.prediction.variance)),
    Prediction.error=c("-",round(prediction.error[2:n.origin]), round(total.prediction.error)),
    Prediction.error.percent=c("-",round(prediction.error.percent[2:n.origin]),
    round(total.prediction.error.percent))
  )
  Mack.model.result1
  #
  sigma2[n.dev]=sigma2[n.dev-1] # 2. variants
  lambda
  sigma2
  Dev.factors.Var.comp.result2<- data.frame (row.names=paste ("j =", 2:n.dev, sep = ""),
  Lambda=round(lambda[2:n.dev],3),
  Sigma.squared=c(round(sigma2[2:(n.dev)],2)))
  Dev.factors.Var.comp.result2

```

```

Ult.claim<-c(rep(0,n.origin))
Ult.claim[1]<-data.cumul[1,n.dev]
for (i in 2:n.origin) {
  Ult.claim[i]<-data.cumul[i,n.dev-i+1]
  for (j in (n.dev-i+2):n.dev) {
    Ult.claim[i]<-Ult.claim[i]*lambda[j]
  }
}
Ult.claim
AccY.reserve<-c(rep(0,n.origin))
for (i in 1:n.origin) {
  AccY.reserve[i]<-Ult.claim[i]-data.cumul[i,n.dev-i+1]
}
total.reserve<-sum(AccY.reserve)
AccY.reserve # Reserve for each origin year
total.reserve # Overall reserve
data.cumul.fitted <- data.cumul
for (i in 2:n.origin) {
  data.cumul.fitted[i,n.dev-i+1] <- data.cumul[i,n.dev-i+1]
  for (j in (n.dev-i+2):n.dev) {
    data.cumul.fitted[i,j]<-data.cumul.fitted[i,j-1]*lambda[j]
  }
}
data.cumul.fitted
process.var <- c(rep(0,n.origin))
for (i in 2:n.origin) {
  process.var[i] <- Ult.claim[i]^2*sum(sigma2[(n.dev-i+1):(n.dev-1+1)]/
    (lambda[(n.dev-i+1):(n.dev-1+1)]^2*data.cumul.fitted[i,(n.dev-i+1):(n.dev-1)]))
}
process.var
total.process.var<-sum(process.var)
D <- c(rep(0,n.origin))
for (i in 2:n.origin) {
  for (k in (n.dev-i+1):(n.dev-1)) {
    D[k] <- sum(data.cumul[1:(n.origin-k),k])
  }
}
D
estimation.var <- c(rep(0,n.origin))
for (i in 2:n.origin) {
  estimation.var[i]<-Ult.claim[i]^2*sum(sigma2[(n.dev-i+1):(n.dev-1+1)]/
    (lambda[(n.dev-i+1):(n.dev-1+1)]^2*D[(n.dev-i+1):(n.dev-1)]))
}
estimation.var
prediction.variance<-process.var+estimation.var
prediction.error<-sqrt(prediction.variance)
prediction.error.percent<-round(100*prediction.error/AccY.reserve)
total.prediction.variance<-0
for (i in 2:(n.origin-1)) {
  total.prediction.variance<-total.prediction.variance+(prediction.variance[i]+Ult.claim[i]
*sum(Ult.claim[(i+1):(n.dev)]))*
  sum(2*sigma2[(n.dev-i+1):(n.dev-1+1)]/
    (lambda[(n.dev-i+1):(n.dev-1+1)]^2*D[(n.dev-i+1):(n.dev-1)])) )
}
total.prediction.variance<-total.prediction.variance+prediction.variance[n.origin]
total.prediction.error<-sqrt(total.prediction.variance)
total.prediction.error.percent<-round(100*total.prediction.error/total.reserve)
total.estimate.var<-total.prediction.variance-total.process.var
# Mack's model; reserve results using sigma10=sigma8
Mack.model.result2<- data.frame(row.names =c(paste("Year", 1:n.origin, sep = ""),"Overall"),
  Reserve=c("-",round(AccY.reserve[2:n.origin]),round(total.reserve)),
  Process.variance=c("-",round(process.var[2:n.origin]),round(total.process.var)),
  Estimation.variance=c("-",round(estimation.var[2:n.origin]), round(total.estimate.var)),
  Prediction.variance=c("-",round(prediction.variance[2:n.origin]), round(total.prediction.variance)),
  Prediction.error=c("-",round(prediction.error[2:n.origin]), round(total.prediction.error)),
  Prediction.error.percent=c("-",round(prediction.error.percent[2:n.origin]),
round(total.prediction.error.percent))
)
Mack.model.result2

```

Maģistra darbs “Nedzīvības apdrošināšanas IBNR rezervju aprēķināšana ar vispārinātā lineārā modeļa un butstrapa metodēm” izstrādāts LU Fizikas un Matemātikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: Nataļja Vidinejeva

(paraksts)

(datums)

Rekomendēju darbu aizstāvēšanai.

Vadītājs: doc. Dr. math. Jānis Valeinis

(paraksts)

(datums)

Recenzents: prof. Dr. habil. math. Andrejs Reinfelds

(paraksts)

(datums)

Darbs iesniegts Matemātikas nodaļā _____

(datums)

(darbu pieņēma)

Darbs aizstāvēts maģistra gala pārbaudījuma komisijas sēdē

_____ prot. Nr. _____, vērtējums _____

(datums)

Komisijas sekretārs/-e: _____

(Vārds, Uzvārds)

(paraksts)