

Robusthet ved skadereservering

Chain-Ladder og influens funksjoner



Sindre Ones

Masteroppgave i Statistikk

Finansteori og forsikringsmatematikk

Universitetet i Bergen, Norge

1. juni 2011

© Sindre Ones 2011

Denne oppgaven er skrevet i Microsoft Office Word 2003 – Font: Times New Roman, skriftstørrelse 12, linjeavstand 1,3mm. Figurer og tabeller er laget i Microsoft Office Excel 2003. Programmering og simuleringer er gjort i R version 2.12.0.

Takk:

Jeg ønsker å takke min veileder, Trygve Nilsen, for all hjelp han har gitt meg i arbeidet med denne oppgaven. Selv om veiledningstimene ofte har endt med avsporing har Trygve gitt med mange gode innspill til oppgaven. Trygve har sammen med Dag Tjøstheim stått for mye av den læring jeg har tilegnet meg opp gjennom årene ved UiB, noe jeg er svært takknemmelig for.

I tillegg ønsker jeg å nevne et par studenter for alt samarbeid opp gjennom årene ved UiB; Leiv Magne, Miriam, Gunhild og ikke minst Hannu. Sistnevnte spesielt for sine gjentatte forsøk på å lære oss medstudenter finsk i lunsjpausene.

Ellers fortjenter Kari H. Holter (DnB NOR Skadeforsikring) en stor takk for hennes innspill til oppgaven, og også Lars Aga Reisæter (PwC) for gode diskusjoner og generell forsikringsprat.

Min romkamerat, André Wallace, fortjenter også en takk for sitt valg om å pusse opp leiligheten rett før innlevering, noe som gjorde det lettere å tilbringe lange dager på lesesalen i innspurten.

En siste takk til familie (mor, far, bror, søster) for generell støtte gjennom hele studietiden.

Bergen, 1. juni 2011
Sindre Ones

Innholdsfortegnelse:

1. Introduksjon	- 2 -
2. Bakgrunn	- 3 -
2.1 Chain-Ladder metoden.....	- 8 -
2.2 Robust statistikk.....	- 12 -
3. GLM og reservering	- 13 -
3.1 GLM – generaliserte lineære modeller	- 13 -
3.2 Chain-Ladder metoden som GLM og marginale summer	- 14 -
4. Influens funksjoner	- 17 -
4.1 Influens funksjoner	- 17 -
4.2 Influens funksjon for Chain-Ladder	- 23 -
4.3 Influens funksjon for Bornhuetter-Ferguson	- 34 -
4.4 Influens analyse for datasett – Chain-Ladder	- 36 -
5. Robust Chain-Ladder metode	- 41 -
5.1 Robust Chain-Ladder	- 41 -
6. Prediksjonsfeil	- 51 -
6.1 Bootstrap	- 53 -
7. Resultater	- 56 -
8. Hvordan fungerer ting i praksis?	- 65 -
9. Oppsummering	- 75 -
10. Appendix	- 77 -
11. Programkode	- 85 -
12. Kilder	- 91 -

1. Introduksjon

En av oppgavene til aktuarer i forsikringsbransjen er å beregne størrelsen på reservene, eller erstatningsavsetningene, som selskapene må sette av for å dekke fremtidige skadeutbetalinger. Reservene skal dekke skader som vil oppstå innenfor kontraktene som allerede er tegnet, men også skader som allerede har skjedd, både de som er rapportert til selskapene, og de som enda ikke er rapportert. Skader som har skjedd, men som enda ikke er rapportert er noe av det vanskeligste å estimere. Når man videre skal estimere fremtidige krav for skader som har skjedd, men som enda ikke er rapportert, kommer man av og til borti datasett som inneholder verdier som avviker stort fra resten av datasettet. Dette skaper problemer, og hvis man ikke håndterer slike uteliggende verdier riktig, kan dette medføre at man setter av for store/ små reserver.

I denne oppgaven har jeg valgt å ta utgangspunkt i artikkelen "A Robustification of the Chain-Ladder method" av Verdonck, van Wouwe og Dhanene [1], for å se nærmere på hvordan man kan oppdage uteliggere med den klassiske Chain-Ladder metoden. Videre har jeg studert influens funksjoner for sjekk av robusthet, og jeg har også tatt med resultater fra bruk av metodene jeg har studert.

Jeg vil først starte med å gi en introduksjon til reserveringsbransjen, Chain-Ladder metoden, og kort om robust statistikk. Etter dette vil jeg ta for meg influens funksjoner i sin helhet, og videre se på influensfunksjonen for Chain-Ladder metoden. Dette for å undersøke hvorvidt Chain-Ladder metoden er robust på et mer matematisk nivå, når det gjelder estimering av reserven.

I kapittel 5 starter jeg å se på en robust Chain-Ladder metode som er basert på artikkelen over. Jeg vil også ta for meg prediksjonsfeil i estimatene mine, for å sammenlikne den robuste Chain-Ladder metoden med den klassiske metoden for et par ulike datasett. I Kapittel 7 har jeg tatt for meg en del resultater etter å ha brukt den klassiske og robuste Chain-Ladder metoden på tre ulike datasett. I tillegg har jeg også fått tilgang til to aktuelle og reelle trekanter fra DnB NOR Skadeforsikring, som jeg har sett på i kapittel 8. I appendix ligger R-kode som jeg har brukt i oppgaven, i tillegg til et par små bevis.

2. Bakgrunn

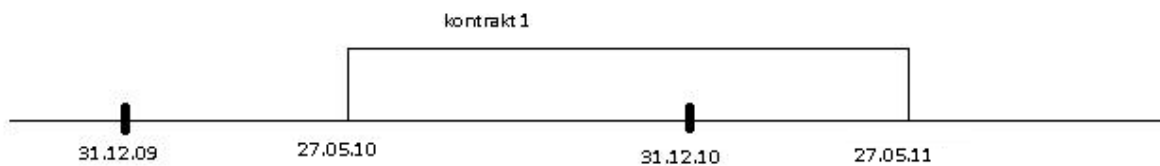
Når en skade inntreffer, hender det ofte at det går tid før skaden oppdages, eller tid før skaden rapporteres inn til forsikringsselskapene. Det kan være ulike årsaker til dette, og hvor lang tid det går før skaden rapporteres kan også variere stort. Ofte er det også snakk om skader der kravene blir utbetalt over flere år. I Norge og flere andre land er det slik at alle forsikringsselskaper hvert år er pålagt å sette av reserver som skal dekke fremtidige krav på skader som har skjedd det aktuelle året, men som enda ikke er rapportert inn, og for skader der kravet ikke er lukket og en forventer flere utbetalinger fremover. IBNR skader heter det ene, og står for Incurred But Not Reported, og det andre er RBNS skader, Reported But Not Settled. Hvert år er altså forsikringsselskapene pålagt å reservere beløp for IBNR og RBNS skader, i det året skaden faktisk skjedde. Dette for å sikre at selskapene har nok midler til å dekke kravene fra forsikringstakerne når skaden oppdages og rapporteres. Det er også en skattemessig hensikt bak dette, i tillegg til at aksjonærer skal få innblikk i hvordan selskapet er finansiert, og hvilken forpliktelse de står ovenfor.

For regnskapsårets slutt (som regel 31.12), må forsikringsselskapene lage regnskap med oversikt over selskapets balanse (eiendeler, forpliktelser og egenkapital). I balansen fremgår selskapets reserver, eller erstatningsavsetninger som det heter, under forpliktelser. Selskapets totale erstatningsavsetninger er summert over avsetningene for hver av bransjene som selskapet opererer i, for eksempel motor, brann / kombinert, ansvar, yrkesskade, marine osv. Hvor mye av selskapets totale erstatningsavsetning som er IBNR / RBNS finner man sjelden i regnskapet (heller ikke i notene til regnskapet), men dette har selskapene god oversikt over i egne systemer. En kan ellers merke seg at når aktuarer flest snakker om IBNR, så menes det IBNR + IBNeR, altså Incurred But Not Reported og Incurred But Not enough Reported. Dette gjør at man også må ta hensyn til skader som er rapportert, men ikke godt nok til at de faller inn under RBNS.

I tillegg til erstatningsavsetning, er premieavsetning en viktig forpliktelse for selskapene. Premieavsetningen er ment å dekke erstatningene for skader som enda ikke er oppstått, men vil oppstå i løpet av kontraktstiden.

La oss ta et eksempel, se Figur 2.1. Vi tegner en motorvogn forsikring (kontrakt 1) 27.05.10, som løper i ett år. Ved årsskifte, 31.12.10, må selskapene gjøre opp årsregnskapet sitt. Det selskapet da vil sette av i erstatningsavsetning vil være forventet beløp for fremtidig utbetaling av denne kontrakten, for skader som har skjedd i perioden 27.05.10 – 31.12.10, men som enda ikke er rapportert, eller ikke er ferdigbehandlet og dermed ikke fullstendig utbetalt. Vi ser at kontrakten gjelder fra dato til dato, og ikke per kalenderår. Dette er vanlig

praksis, noe som gjør at man i tillegg til erstatningsavsetningen må sette av en premieavsetning også. Premieavsetningen er ment å dekke skader som enda ikke har skjedd, men som vil skje innenfor kontraktstiden, altså i perioden 01.01.11 – 27.05.11. For selskaper som har kontrakter som er riktig priset, er det svært vanlig å sette premieavsetningen lik uopptjent premie ved regnskapsårets slutt. Hva som er riktig priset, er vanskelig å si uten å ta en nærmere titt på selskapet prising, men de fleste selskaper gjør denne likheten per i dag. Det antas da at premie opptjenes jevnt over kontraktstiden.



Figur 2.1

Som nevnt flere ganger nå, deles erstatningsavsetningen inn i to, nemlig RBNS og IBNR. Når jeg senere i denne oppgaven bruker ordet reserve, mener jeg erstatningsavsetning for IBNR skader. RBNS skader er skader som er rapportert til selskapene, noe som gjør at man bør ha en relativt god formening om hva de totale kostnadene i forbindelse med disse skadene vil være. Alle selskaper har ansatt egne skadebehandlere / takstmenn eller bruker eksterne skadebehandlere / takstmenn, som er ansvarlig for å dømme omfanget av en skade etter at den er rapportert, og hvor mye skaden totalt vil koste. Det er altså mye lettere å estimere utbetalingsprofilen til kravene når de først er rapportert, ettersom man i det minste vet *noe* om omfanget av skaden, og det er da ikke like aktuelt å bruke statistiske metoder til å beregne de samlede avsetningene.

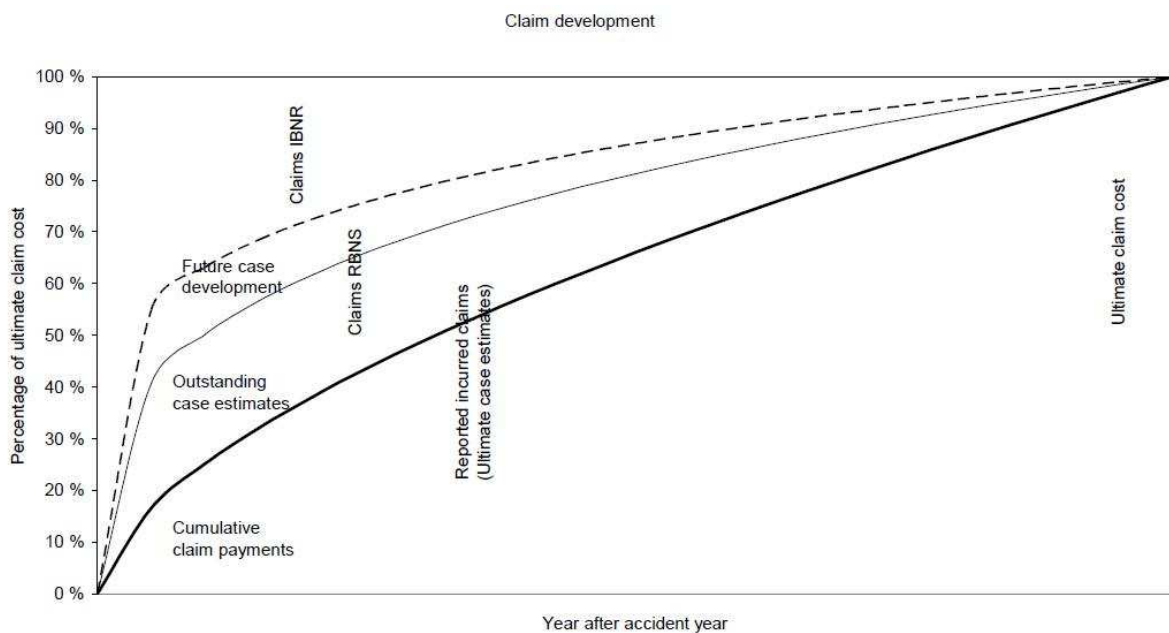
IBNR skader er derimot mye vanskeligere å estimere, ettersom ingenting er rapportert inn til selskapene. Det er her den klassiske Chain-Ladder metoden kommer inn, ettersom det har vist seg at denne metoden ofte gir gode estimater på IBNR skader, spesielt for langhalede bransjer. Med langhalede bransjer mener jeg her bransjer der det forventes utbetaling for det aktuelle skadeår i flere år fremover, mens korthalede bransjer ofte blir gjort opp i løpet av få år.

Dataene som aktuarene har tilgjengelig for estimering av fremtidige krav er vanligvis oversikt over kumulative skadeutbetalinger og rapporterte inntruffet skader (*incurred claims / ultimate case estimate* som det gjerne kalles), alt for ulike skadeår og utviklingsår, og selvsagt per

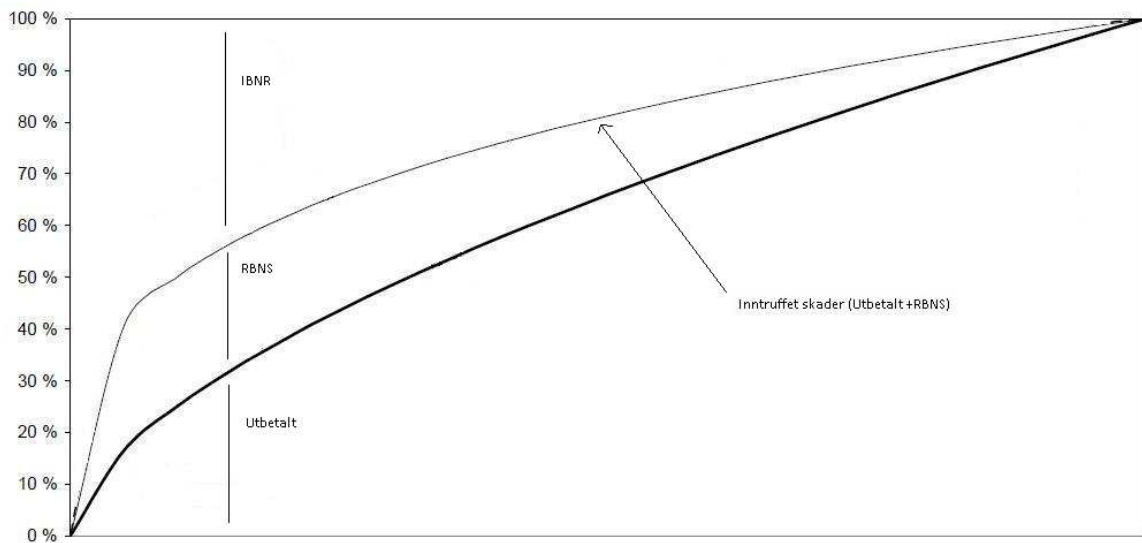
bransjenivå og samlet. Etter hvert som årene utvikler seg, vil andelen av rapporterte skader i forhold til totale skader for angitt skadeår gå mot 100 %, mens utestående IBNR reserve vil gå mot 0. Figur 2.2 [9] under viser typisk utvikling av et skadeår.

Når IBNR avsetningen skal estimeres, er det vanlig å gjøre dette på bakgrunn av trekanten av rapporterte skader (incurred claims). Dette er trekanten som inneholder utbetalte krav + skadebehandlerens estimat på fremtidig utbetaling av skader som så langt er rapportert. Ved å fullføre trekanten til en firkant finner man da estimat på IBNR reserven.

Som vi skal se fra et eksempel senere i oppgaven, hender det at trekanten for rapporterte krav er ufullstendig og ubrukelig for estimering av IBNR reserven. Det kan da være aktuelt å ta for seg trekanten for betalte krav, noe som gjør at man finner estimat på IBNR + RBNS.

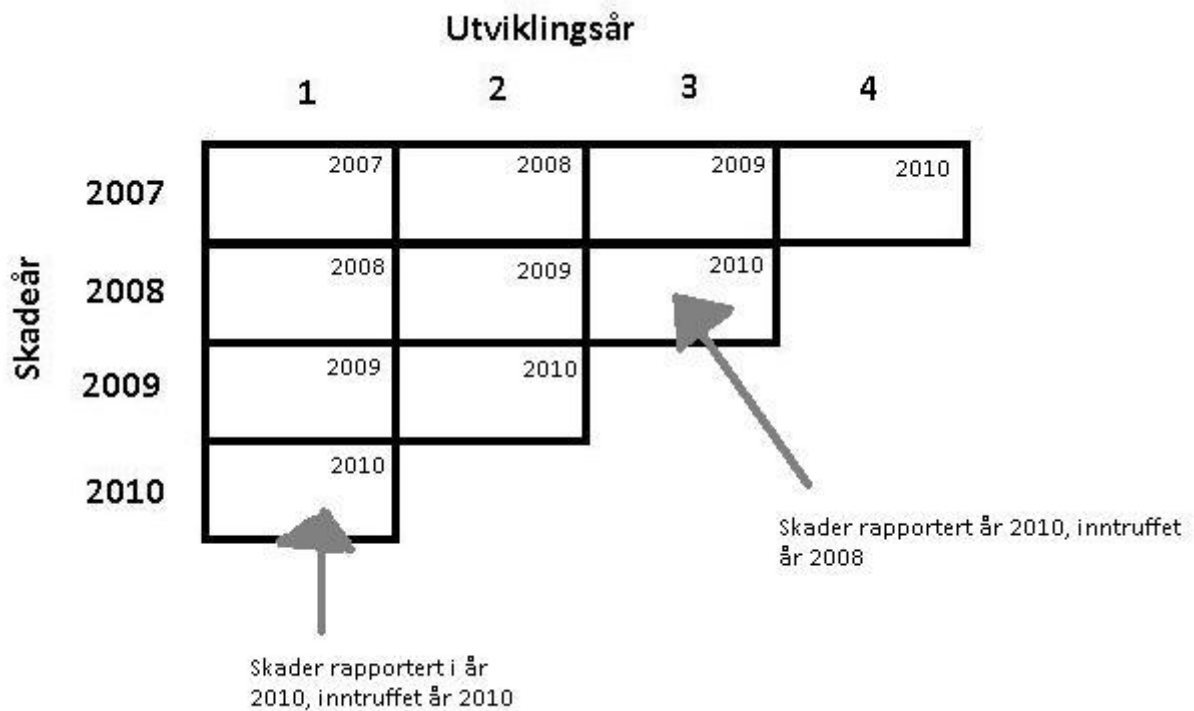


Figur 2.2



Figur 2.2 (alternativ)

Chain-Ladder metoden er en av de mest vanlige måtene å estimere fremtidige IBNR krav på. I kapittel 2.1 vil jeg kort forklare hvordan Chain-Ladder metoden er bygget opp. Chain-Ladder metoden er en deterministisk metode som har fått mye kritikk av statistikere tidligere, men det har blitt vist at flere stokastiske modeller underbygger Chain-Ladder metoden og gir samme resultater. Så selv om Chain-Ladder metoden virker veldig enkel i seg selv, kan Chain-Ladder estimatene bli nøyaktig reproduisert ved for eksempel å bruke Poisson baserte generaliserte lineære modeller (GLM).



Figur 2.3

Figur 2.3 over angir hvordan en typisk trekant for IBNR estimering er satt opp. Når man bruker statistiske metoder, f.eks Chain-Ladder og estimerer utestående reserve, vil man sitte igjen med et estimat på hvordan utbetalingsprofilen på selskapets reserver vil se ut. Tabell 2.1 viser eksempel på et estimat av en utbetalingsprofil basert på datasettet fra tabell 7.9. Vi ser at selskapet må sette av totalt 18 690 856 for å dekke IBNR skader for kontraktene som datasettet gjelder for. Merk ellers at det er vanlig å kombinere utbetalingsprofilen for IBNR og RBNS når data først er estimert, for å komme frem til den totale erstatningsavsetningen.

Skadeår	SUM	Utbetalingsår								
		2011	2012	2013	2014	2015	2016	2017	2018	2019
2002	94 634	94 634	0	0	0	0	0	0	0	0
2003	469 511	375 833	93 678	0	0	0	0	0	0	0
2004	709 638	247 190	370 179	92 268	0	0	0	0	0	0
2005	984 889	334 148	226 674	339 456	84 611	0	0	0	0	0
2006	1 419 459	383 287	351 548	238 477	357 132	89 016	0	0	0	0
2007	2 177 641	605 548	424 501	389 349	264 121	395 534	98 588	0	0	0
2008	3 920 301	1 310 258	725 788	508 792	466 660	316 566	474 073	118 164	0	0
2009	4 278 972	1 018 834	1 089 616	603 569	423 113	388 076	263 257	394 241	98 266	0
2010	4 625 811	856 804	897 410	959 756	531 636	372 687	341 826	231 882	347 255	86 555
SUM	18 680 856	5 226 536	4 179 394	3 131 668	2 127 272	1 561 879	1 177 744	744 287	445 521	86 555
<i>Diskontert</i>	<i>16 998 911</i>	<i>5 092 898</i>	<i>3 940 580</i>	<i>2 852 243</i>	<i>1 867 127</i>	<i>1 318 483</i>	<i>954 876</i>	<i>578 897</i>	<i>332 047</i>	<i>61 761</i>

Tabell 2.1

Den nederste raden, "Diskontert", er rett og slett bare for å vise at reservene vil bli en del lavere i det nye Solvens II regelverket, som vil tre i kraft fra 01.01.13. Med det nye regelverket er selskapene pliktig til å diskontere reservene. Det nye solvens II regelverket er svært omfattende, og jeg har ikke tenkt å ta for meg noe mer av det nye regelverket i denne oppgaven, men ønsker kjapt å ha nevnt dette med diskontering av reservene.

2.1 Chain-Ladder metoden

Chain-Ladder metoden er en metode for å estimere fremtidige krav basert på et vektet gjennomsnitt av tidligere krav vi har historiske data for. Den vanlige måte å gjøre dette på, er å sette opp en såkalt "run-off triangle" eller trekant, med oversikt over utviklingen av krav som har kommet fra ulike skadeår.

Development Year							
Accident Year	1	2	...	$n - i + 1$...	$n - 1$	n
1	X_{11}	X_{12}	...	$X_{1,n-i+1}$...	$X_{1,n-1}$	X_{1n}
2	X_{21}	X_{22}	...	$X_{2,n-i+1}$...	$X_{2,n-1}$	
⋮		
i	X_{i1}	X_{i2}	...	$X_{i,n-i+1}$			
⋮				
n	X_{n1}						

Figur 2.4

Ser vi på Figur 2.4, ser vi hvordan vi setter opp en slik trekant for å få oversikten over utviklingen av krav fra ulike skadeår. De ulike kolonnene kaller vi for utviklingsår, og de

ulike radene for skadeår. Utviklingsår vil da si hvordan kravene blir fordelt på de ulike årene etter skaden fant sted. For eksempel vil utviklingsår 1 være kravet som ble rapportert i samme året som skaden skjedde, utviklingsår 2 vil være kravet som ble rapportert året etter, osv.

$X_{i,j}$ vil være kravet fra skadeår i , rapportert utviklingsår j . (Ofte blir utviklingsår 1 sett på som utviklingsår 0 i stedet, men av hensyn til programmering senere osv, lar jeg utviklingsårene gå fra 1,2,...)

I datasettene jeg kommer til å ta for meg senere, opererer vi med like mange skadeår som utviklingsår. Vi er i dag ved slutten av år n , og har historiske data for alle krav som er rapportert tom år n . For skadeår $n-1$ har vi da altså kun krav for to utviklingsår. Med andre ord, når vi setter opp en slik trekant, får vi en *firkant* som kun er fylt ut på øvre halvdel. Målet videre blir da å fullføre trekanten til en firkant, med estimater på hvordan kravene vil bli for årene fremover.

For å kunne bruke Chain-Ladder metoden er det nødvendig at antall utviklingsår for det første skadeåret i trekanten er fullt utviklet.

Vi antar videre at vi har et sett med inkrementelle krav,

$$\{X_{i,j} \mid i = 1, \dots, n; j = 1, \dots, n - i + 1\}$$

Videre får vi akkumulerte krav,

$$C_{i,j} = \sum_{k=1}^j X_{i,k}$$

Forsikringsselskapene er videre interessert i å estimere den totale reserven de må sette av for fremtidige utbetalinger, R . Vi får da

$$R_i = C_{i,n} - C_{i,n-i+1}$$

$$R = \sum_{i=1}^n R_i$$

for henholdsvis skadeår i , og total reserve, der C må estimeres for den nedre trekanten.

Chain-Ladder metoden baserer seg nettopp på akkumulerte krav. Metoden går ut på å se på hvordan de kumulative kravene endrer seg for de ulike utviklingsårene for historiske krav, og dermed lage forholdstall for å estimere kumulative krav for den nedre trekanten vår.

Disse såkalte forholdstallene vi ønsker å lage, blir kalt utviklingsfaktorer, og vi skriver dem ofte som \hat{f}_j for utviklingsår j . Utviklingsfaktorene kan da videre estimeres med Chain-Ladder metoden på følgende måte:

$$(2.1) \quad \hat{f}_j = \frac{\sum_{i=1}^{n-j+1} C_{i,j}}{\sum_{i=1}^{n-j+1} C_{i,j-1}} \quad 2 \leq j \leq n$$

Når vi videre skal estimere de ukjente kumulative kravene i nedre del av trekanten, finner vi dem på følgende måte ved hjelp av de estimerte utviklingsfaktorene:

$$\begin{aligned} \hat{C}_{i,n-i+2} &= C_{i,n-i+1} \cdot \hat{f}_{n-i+2} & 2 \leq i \leq n \\ \hat{C}_{i,k} &= \hat{C}_{i,k-1} \cdot \hat{f}_k & 2 \leq i \leq n, \quad n-i+3 \leq k \leq n \end{aligned}$$

Et av problemene med Chain-Ladder metoden, er at den er svært sensitiv for uteliggende data (outliers). En verdi i datasettet som er svært forskjellig fra resten av datasettet, blir kalt en uteligger. En uteligger kan komme av målefeil, kopieringsfeil og lignende (type feilplassering av komma f.eks), men som regel vil det skyldes data som kommer fra helt spesielle situasjoner, og som skjer i svært sjeldne tilfeller. Eksempler på slike ekstreme hendelser / katastrofegivenheter kan for eksempel være:

- Rocknes ulykken i 2004
- Sleipner forliset i 1999
- En ekstrem hard og lang vinter (dårlig føre, frostskafer, osv)
- Flom på Østlandet
- Vindstorm på Vestlandet
- Store skred/ras
- Epidemier
- Andre sjeldne hendelser

Når det gjelder punktene over, så er det vært å nevne Norsk Naturskade Pool, NNP. Alle bygninger og ting som brannforsikres dekkes også av naturskadepoolens ordning ved naturskader. Dette er en ordning som alle skadeforsikringselskap som tegner brannforsikring er medlem av, og hvert selskap må betale inn en andel av premien til et naturskadefond hvert år. Selv om denne ordningen eksisterer, må reservene estimeres på brutto nivå, da det er selskapene selv som står for behandling av skadene og utbetalingene. Ved en eventuell

naturkatastrofe vil de samlede erstatningene utliknes mellom selskapene i Norge. En annen ting som er vært å nevne, er at ved slike katastrofe- og sjeldne begivenheter, blir som regel de fleste skadene oppdaget og rapportert med en gang, og IBNR reservering høres kanskje ikke så aktuelt ut. Husk da, at som tidligere nevnt, vil IBNR også omfatte IBNeR (not enough reported), noe som gjør at ved katastrofebegivenheter som oppdages med en gang, kan det gå lang tid før alt av nødvendig informasjon for et ultimatum er rapportert. For eksempel når det gjelder båtforlis eksemplene over, er det naturlig at det vil gå lang tid før man har en fullstendig oversikt over det endelige omfanget av ulykken, og at alt av relevant skadeomfang er rapportert (skade på skip, last, mannskap, natur osv...). Slik ekstreme krav må som regel gjennom rettsvesenet også, noe som kan gjøre at skadeomfanget kan komme både gunstigere og dårligere ut for selskapene.

Merk ellers at ulykke-, yrkesskade- og gruppelivforsikringer kan få en sentral rolle også, for eksempel ved personskader der skaden for eksempel ikke merkes før det har gått litt tid. Skader på bygg oppdages ikke alltid med en gang heller, for eksempel ved vannskader som kan føre til sopp, eller frostskafer osv. En katastrofebegivenhet som omfatter mange personer og stort område kan altså få store konsekvenser, også ved IBNR reservering.

Slike katastrofebegivenheter eller sjeldne hendelser kan medføre at den klassiske Chain-Ladder metoden gir estimater på reserven der data fra katastrofe år gir for stort bidrag i forhold til hva vi kan forvente oss fremover. Dette gjør at vi gjerne ønsker å se på mer robuste metoder for beregning av reserven. Robuste metoder i dette tilfellet, vil si metoder der vi kommer frem til noenlunde samme resultat som vi ville gjort med det samme datasettet, men uten uteliggerne.

Jeg ønsker å fremheve at det som er ønskelig med en mer robust Chain-Ladder metode er ikke å erstatte den klassiske metoden, men heller å kunne ha et hjelpeverktøy som kan brukes i tillegg, **til å oppdage uteliggere**, og kanskje gi en idé om hvordan en alternativt kan legge opp reserven.

2.2 Robust statistikk

Robust statistikk, eller en robust statistisk metode, er en av flere statistiske metoder som har som mål å gjengi en klassisk statistisk metode, men som videre vil være mindre sensitiv mot uteliggende data (outliers). En klassisk statistisk metode baserer seg nesten alltid på antakelser som en ofte ikke vil møte i praksis. Typisk er gjerne en metode som baserer seg på at residualer / feilledd er normalfordelte, i hvert fall noenlunde, men det er ikke alltid dette stemmer 100 %. Kanskje viser deg seg at 95 % av feilleddene er noenlunde normalfordelt, men videre at de resterende 5 % er normalfordelt med en høyere varians.

Slike uteliggere kan være med på å gjøre klassiske metoder type dårligere enn hva de ville vært uten uteliggerne. Det er her robust statistikk kommer inn.

En robust statistisk metode har som mål å kunne tilpasse data som inneholder uteliggere, og gi noenlunde samme resultat som vi ville fått uten uteliggere i datasettet.

Medianen er et eksempel på en robust metode, mens middelveiden/gjennomsnittet (mean) ikke er det. Har vi svært store eller lave verdier i et datasett, vil middelveiden påvirkes av disse, noe medianen ikke vil gjøre. Median og middelveid er noe som blant annet vil bli diskutert senere i forbindelse med influens funksjoner.

3. GLM og reservering

Flere stokastiske metoder innenfor reserveringsbransjen gir samme estimater som Chain-Ladder metoden, og vi kan si at disse metodene underbygger Chain-Ladder metoden. En av de enkleste stokastiske modellene som har denne egenskapen, er å tilpasse historiske krav som en Poisson modell, eventuelt overdispersert Poisson. Andre stokastiske modeller som også underbygger Chain-Ladder metoden, og er sterkt linket til Poisson modellen, er negativ binomisk, den multiplikative modellene, og Mack sin modell, se Hess [11]. Poisson og negativ binomisk er antakeligvis de mest brukte stokastiske modellene når det gjelder reserveringsbransjen.

Jeg vil senere i oppgaven ta for meg Poisson antakelsen når jeg skal finne influensfunksjoner. Først vil jeg kort introdusere GLM i sin helhet, og videre vise hvordan Poisson modellen som en GLM kan brukes til å finne Chain-ladder estimatene.

3.1 GLM – generaliserte lineære modeller

GLM er rett og slett en generalisering av vanlig lineær regresjon, der det nå er mulig å la den lineære modellen være linket til responsvariabelen via en link funksjon, annet enn på identitet skala.

En GLM er bygget opp av tre komponenter, nemlig en stokastisk komponent, en systematisk komponent og en link funksjon, [15].

1. Responsvariablene Y_1, Y_2, \dots, Y_N er de stokastiske komponentene, og må tilhøre den eksponentielle familie. Hvis fordelingen til hver av Y_i 'ene, $f(y_i, \theta_i)$, kan skrives på formen

$$f(y_i, \theta_i) = \exp[y_i b(\theta_i) + c(\theta_i) + d(y_i)]$$

der fordelingen til hver y_i avhenger av en parameter θ_i , mens b , c og d er kjente funksjoner, tilhører Y_i 'ene den eksponentielle familie.

2. Den systematiske komponenten er selve modellen. Dette er funksjonen av prediksjonsvariablene x_i 'ene, som er relatert til forventingen til Y_i . Vi angir vanligvis den systematiske komponenten som en lineær prediktor η , som vi har på formen

$$\eta = X\beta = \beta_1 x_1 + \beta_2 x_2 + \dots$$

3. Det siste vi trenger er en link funksjon g som knytter denne lineære prediktoren sammen med forventningen til responsvariabelen Y ,

$$g(\mu) = \eta$$

$$\mu = E[Y] = g^{-1}(\eta)$$

Link funksjonen kan f.eks være log-link (Poisson) der $\log \mu = \eta$, eventuelt andre kjente funksjoner som f.eks identitet, inverse og logit.

Når vi har alle disse tre komponentene på plass, har vi en GLM modell, og vi kan kjøre modellen f.eks i R og finner estimater for alle koeffisientene β_1, β_2, \dots til de ulike prediksjonsvariablene x_1, x_2, \dots

3.2 Chain-Ladder metoden som GLM og marginale summer

Videre ønsker vi å se på sammenhengen mellom Chain-Ladder metoden, og en Poisson basert GLM modell.

Hvis vi lar X_{ij} være de inkrementelle kravene fra trekanten vår, og antar videre at vi har en multiplikativ modell som vi kan skrive

$$(3.1) \quad X_{ij} = \alpha_i \beta_j$$

,som har parametere for hver rad i og hver kolonne j . Ved å anta at X_{ij} er uavhengige og Poisson fordelte, vil X_{ij} 'ene være medlem av den eksponentielle familie, og vi har en GLM. Forventningen til X_{ij} kan da skrives

$$E[X_{ij}] = \exp(\log(\alpha_i \beta_j)) = \exp(\log \alpha_i + \log \beta_j)$$

, noe som gir oss en logaritmisk link funksjon. Chain-Ladder metoden kan altså utledes fra (3.1) hvis følgende antakelser om fordelingen er oppfylt:

$$X_{ij} \sim \text{Poisson}(\alpha_i \beta_j)$$

Når vi videre estimerer parameterne $\alpha_i > 0$ og $\beta_j > 0$ ved MLE, får vi en multiplikativ GLM med Poisson fordeling og en log-link funksjon. Siden en av parameterne er overflødig, legger vi på en ekstra restriksjon. Det kan f.eks være $\beta_1 = 1$, men det er mer vanlig å kreve at

$\sum_{j=1}^n \beta_j = 1$. Dette gjør at vi kan tolke β_j som en andel av hvordan kravene fordeler seg

mellom utviklingsårene, mens α_i da kan tolkes som et slags "volummål" på skadeår i .

Å estimere α_i og β_j ved MLE gir samme resultater som marginale summer metoden, Kaas [12]. Dette kan enkelt vises, se Appendix A.

Vi kan altså finne estimatene for α_i og β_j når vi har restriksjonen $\sum_{j=1}^n \beta_j = 1$ ved metoden marginale summer, noe vi vil se videre på nå, for å bruke senere når vi skal finne influens funksjonene for Chain-Ladder metoden.

Når vi tilpasser trekanten vår med marginale summer, kan vi skrive

$$RS_i = \sum_{j=1}^{n-i+1} X_{ij} = \sum_{j=1}^{n-i+1} \hat{\alpha}_i \hat{\beta}_j$$

$$CS_j = \sum_{i=1}^{n-j+1} X_{ij} = \sum_{i=1}^{n-j+1} \hat{\alpha}_i \hat{\beta}_j$$

(RS = row sum, CS = column sum)

Og siden $\sum_{j=1}^n \beta_j = 1$, får vi at for en gitt $l < n$, der estimatene $\hat{\beta}_{l+1}, \dots, \hat{\beta}_n$ og $\hat{\alpha}_1, \dots, \hat{\alpha}_{n-l}$

allerede er funnet, at de to uttrykkene over leder til:

$$\hat{\alpha}_1 = RS_1$$

$$\hat{\alpha}_2 = \frac{RS_2}{\sum_{j=1}^{n-1} \hat{\beta}_j} = \frac{RS_2}{1 - \hat{\beta}_n}$$

$$\hat{\alpha}_3 = \frac{RS_2}{\sum_{j=1}^{n-2} \hat{\beta}_j} = \frac{RS_2}{1 - (\hat{\beta}_n + \hat{\beta}_{n-1})},$$

..., OSV

som gir oss

$$(3.2) \quad \hat{\alpha}_l = \frac{RS_l}{1 - \sum_{j=l+2}^n \hat{\beta}_j}$$

Tilsvarende får vi for $\hat{\beta}_l$, som vi skriver som

$$(3.3) \quad \hat{\beta}_{n-l+1} = \frac{CS_{n-l+1}}{\sum_{i=1}^l \hat{\alpha}_i}$$

Det vi har gjort her er rett og slett å bruke metoden marginale summer og en rekursiv fremgangsmåte for å finne disse uttrykkene, *Verbeek's algorithm for chain ladder* [12]. Vi har nå to uttrykk som vi kan bruke til å finne estimater for α_i og β_j , som vi vil få bruk for i neste kapittel.

4. Influens funksjoner

Influensfunksjonen for en estimator beskriver effekten av en liten endring i et av datapunktene som brukes i beregningen av estimatoren. Jeg ønsker å se på hvordan endringer i kravene i reserveringsbransjen vil påvirke den totale reserven som en ønsker å estimere. For å se på dette på et mer matematisk nivå, tar jeg for meg influensfunksjoner.

4.1 Influens funksjoner

Influens funksjoner er en måte å undersøke om en estimator er *begrenset/ubegrenset*. At en estimator er *begrenset*, vil si at estimatoren er avgrenset, og at uteliggere vil ha liten effekt på selve estimatet. Er estimatoren derimot *ubegrenset*, kan uteliggere få stor effekt på selve estimatet en kommer frem til.

Jeg vil nå ta for meg selve definisjonen på influens funksjoner, videre vise noen eksempler der vi ser på hvorvidt gjennomsnittet og medianen er robust, før jeg vil ta for meg influens funksjonen i forbindelse med Chain-Ladder metoden.

I stedet for å se på en estimator med vanlig input, x_1, \dots, x_n , kan en estimator sees på som en funksjon av den kumulative fordelingen som dataene følger, F_n . Slike funksjoner, som har andre funksjoner som argument, blir kalt *funksjonaler*. F.eks hvis T_n er en estimator for et sample på størrelse n , så er en *statistisk funksjonal* som svarer til denne estimatoren en funksjon T , slik at $T(F_n(x_1, \dots, x_n)) = T_n(x_1, \dots, x_n)$. Denne er videre Fisher konsistent hvis $T(F_\theta) = \theta$, se under.

Merk at det for et sample X_1, X_2, \dots, X_n , er det likegyldig om vi har kjennskap til sample-dataene, eller kjennskap til den empiriske fordelingsfunksjonen F_n . Siden F_n har hopp på størrelse $1/n$ ved hver X_i , kan vi lese av verdien for X_i hver gang funksjonen har hopp på denne størrelsen. Dermed kan en estimator $T = T(X_1, X_2, \dots, X_n)$ skrives som $T(F_n)$.

Fisher konsistent

Anta at vi har et sample av data, x_1, \dots, x_n der hver x_i følger en kumulativ fordeling F_θ som avhenger av en ukjent parameter θ . Hvis en estimator for θ basert på vårt sample av data kan representeres som en funksjon av den empiriske fordelingsfunksjonen \hat{F}_n :

$$\hat{\theta} = T(\hat{F}_n)$$

da sies estimatoren å være Fisher-konsistent dersom:

$$T(F_\theta) = \theta$$

Definisjon – Influens funksjon

Hvis vi har et sample av data, X_1, X_2, \dots, X_n , fra en populasjon med cdf F , så er influens funksjonen til en estimator $T = T(F_n)$ i et punkt x gitt ved,

$$(4.1) \quad IF(T, x) = \lim_{\delta \rightarrow 0} \frac{1}{\delta} [T(F_\delta) - T(F)]$$

$$(4.2) \quad = \frac{d}{d\delta} T(F_\delta) |_{\delta=0}$$

der $X \sim F_\delta$ hvis

$$X \sim F, P = 1 - \delta$$

$$X = x, P = \delta$$

Altså, $X \sim F$ med sannsynlig $1 - \delta$, og $X = x$ med sannsynlig δ , det vil si at F_δ er en blanding av F og et punkt x . Vi ønsker altså å se hvordan en estimator endrer seg med x for en ny populasjon med cdf F_δ , der F_δ består av $1 - \delta$ fra F , og δ fra x .

Fra (4.2) ser vi at influensfunksjonen kan tolkes som den deriverte av $T(F_\delta)$.

Jeg vil nå ta for meg gjennomsnitt og median, og ønsker å benytte definisjonen ovenfor til å finne influensfunksjonene, Casella & Berger [10]. At medianen er en robust estimator, mens gjennomsnittet ikke er det er noe alle vet, og jeg skal nå forsøke å vise dette ved hjelp av influensfunksjoner.

Influens funksjon for gjennomsnitt

Antar at vi har en populasjon med kontinuerlig cdf F , der μ er populasjonsgjennomsnittet, mens \bar{X} er snittet vi regner ut fra vårt datasample X_1, X_2, \dots, X_n . Vi lar $T(\cdot)$ være en funksjonal som beregner gjennomsnittet av en populasjon, som gjør at vi har $T(F_n) = \bar{X}$, og $T(F) = \mu$,

Fra definisjonen av influens funksjoner, finner vi da vår nye *populasjon*,

$$F_\delta = F(1 - \delta) + x\delta$$

Siden $T(\cdot)$ er en funksjonal som beregner gjennomsnittet av en populasjon, får vi

$$T(F_\delta) = \mu(1 - \delta) + x\delta = \mu + \delta(x - \mu)$$

Vi bruker videre (4.1) og finner influensfunksjonen som

$$\begin{aligned} \text{IF}(\bar{X}, x) &= \lim_{\delta \rightarrow 0} \frac{1}{\delta} [T(F_\delta) - T(F)] \\ &= \lim_{\delta \rightarrow 0} \frac{1}{\delta} [\mu(1 - \delta) + x\delta - \mu] \\ &= \lim_{\delta \rightarrow 0} \frac{1}{\delta} [\delta(x - \mu)] \\ &= x - \mu \end{aligned}$$

Eventuelt ved å bruke (4.2) og videre derivere $T(F_\delta)$ med hensyn på δ gir også

$$\text{IF}(\bar{X}, x) = x - \mu.$$

Det går da frem at når x øker, vil også dens influens på \bar{X} øke. Altså vil influensfunksjonen til gjennomsnittet være ubegrenset. Det vil si at store observasjoner, uteliggere, vil ha stor innflytelse på gjennomsnittet.

Influens funksjon for medianen

For medianen har vi at $T(F) = m$ hvis $P(X \leq m) = \frac{1}{2}$, eller $m = F^{-1}(\frac{1}{2})$.

Hvis $X \sim F_\delta$ får vi da fra definisjonen over at

$$\begin{aligned}
 P(X \leq a) &= P(X \leq a | X \sim F)(1 - \delta) + P(X \leq a | X = x)\delta \\
 &= (1 - \delta)F(a) + I[x \leq a]\delta
 \end{aligned}$$

Der a antas å være medianen til data som følger fordeling F_δ .

Altså,

$$(4.3) \quad P(X \leq a) = \begin{cases} (1 - \delta)F(a), & x > a \\ (1 - \delta)F(a) + \delta, & \text{ellers} \end{cases}$$

som kan vises på følgende måte:

$$P(X \leq a | X = x) = \frac{P(X \leq a, X = x)}{P(X = x)} = 0 \quad \text{når } x > a$$

og

$$P(X \leq a | X = x) = \frac{P(X \leq a, X = x)}{P(X = x)} = 1 \quad \text{når } x \leq a$$

Hvis $P(X \leq m) = \frac{1}{2}$ for ekte populasjon F , så vil $P(X \leq a) = \frac{1}{2}$ for den nye populasjonen F_δ .

Det gir fra (4.3) at

$$\begin{aligned}
 (1 - \delta)F(a) = \frac{1}{2} &\Rightarrow a = F^{-1}\left(\frac{1}{2(1 - \delta)}\right) \\
 (1 - \delta)F(a) + \delta = \frac{1}{2} &\Rightarrow a = F^{-1}\left(\frac{\frac{1}{2} - \delta}{1 - \delta}\right)
 \end{aligned}$$

Og vi får da følgende uttrykk for en estimator til medianen for den nye populasjonen vi ser på:

$$T(F_\delta) = F^{-1}\left(\frac{1}{2(1 - \delta)}\right) \quad \text{når } x > F^{-1}\left(\frac{1}{2(1 - \delta)}\right)$$

$$T(F_\delta) = F^{-1}\left(\frac{\frac{1-\delta}{2}}{1-\delta}\right) \quad \text{ellers}$$

For å bruke (4.1) for å finne influensfunksjonen, skriver vi $a_\delta = F^{-1}\left(\frac{1}{2(1-\delta)}\right)$, og finner influens funksjonen som

$$\begin{aligned} \text{IF}(M, x) &= \lim_{\delta \rightarrow 0} \frac{1}{\delta} [T(F_\delta) - T(F)] \\ &= \lim_{\delta \rightarrow 0} \frac{1}{\delta} [a_\delta - a_0] \\ &= a'_\delta |_{\delta=0} \end{aligned}$$

der

$$F(a_\delta) = \frac{1}{2(1-\delta)},$$

Ved å ta den deriverte på begge siden av likhetstegnet, får vi

$$\frac{d}{d\delta} F(a_\delta) = \frac{d}{d\delta} \frac{1}{2(1-\delta)}$$

Som da blir

$$\begin{aligned} f(a_\delta) a'_\delta &= \frac{1}{2(1-\delta)^2} \\ \Rightarrow a'_\delta &= \frac{1}{2(1-\delta)^2 f(a_\delta)} \end{aligned}$$

og vi får da

$$\begin{aligned} IF(M, x) &= a'_\delta \Big|_{\delta=0} \\ &= \frac{1}{2(1-\delta)^2 f(a_\delta)} \Big|_{\delta=0} \\ &= \frac{1}{2f(a_0)} \\ &= \frac{1}{2f(m)} \end{aligned}$$

siden $a_0 = m$, ettersom $\delta = 0 \Rightarrow F_\delta = F$.

Vi gjør tilsvarende for den andre grensen, og får tilslutt

$$IF(M, x) = \begin{cases} \frac{1}{2f(m)}, & x > m \\ -\frac{1}{2f(m)}, & \text{ellers} \end{cases}$$

Det er da klart at influensfunksjonen for medianen er *begrenset*, som vil si at når x blir veldig stor, vil innflytelsen x har på medianen være begrenset.

Vi har altså vist ved influensfunksjoner at gjennomsnittet for et data sample ikke er robust, mens medianen for det samme data sample vil være robust.

Jeg tar videre for meg Chain-Ladder metoden, og ønsker å finne influensfunksjoner for denne metoden, og undersøke hvorvidt den er robust eller ikke, Verdonck [7].

4.2 Influens funksjon for Chain-Ladder

Vi ser nå på trekanten vår, hvor vi har data $X = x_{11}, x_{12}, \dots, x_{ij}, \dots, x_{n1}$, der vi antar $X_{ij} \sim \text{Poisson}(\alpha_i \beta_j)$, slik at $E(X_{ij}) = \alpha_i \beta_j$.

For å finne influens funksjonen for CL metoden, må vi finne statistiske funksjonaler for å bruke definisjonen av influensfunksjoner. I stedet for å se på hvilken influens et punkt har på den totale reserven, ser vi i første omgang på hvilken influens et punkt har på beregning av parameterne α og β , som videre gir oss estimat på utestående krav og da også den totale reserven.

Vi starter ved å beregne statistiske funksjonaler for α og β ved å bruke de rekursive uttrykkene vi kom frem til tidligere, 3.2 og 3.3, der vi viste at en Poisson modell og marginale summer metoden gir samme resultater som CL.

Vi lar $T_{\alpha_1}(P_{\alpha\beta})$ være en funksjonal som beregner α_1 på bakgrunn av fordelingsfunksjonen $P_{\alpha\beta}$. (Merk at vi tidligere har brukt F for fordelingsfunksjonen, der vi nå bruker P)

Ved å la $P_{\alpha\beta}$ være den *ekte* fordelingsfunksjon, mens P_n er den empiriske fordelingsfunksjonen for data i trekanten vår, får vi ved å bruke 3.2 for α_1 at:

$$\hat{\alpha}_1 = RS_1 = \sum_{j=1}^n X_{1j}$$

Siden vi ønsker å basere vår funksjonal på P_n , får vi at

$$T_{\alpha_1}(P_n) = E_{P_n} \left(\sum_{j=1}^n X_{1j} \right) = \sum_{j=1}^n E_{P_n}(X_{1j}) = \sum_{j=1}^n \hat{\alpha}_1 \hat{\beta}_j = \hat{\alpha}_1$$

og $T(P_n)$ er altså en tilfredsstillende funksjonal for α_1 .

Hva så med generell α_l ?

Vi gjør som vi gjorde tidligere med marginal totals, og antar at det er OK for

$T_{\alpha_1}(P_n), \dots, T_{\alpha_{l-1}}(P_n)$ og $T_{\beta_n}(P_n), \dots, T_{\beta_{n-l+2}}(P_n)$ for en gitt $l < n$. Vi får da ved å bruke (3.2)

$$T_{\alpha_l}(P_n) = \frac{E_{P_n}(\sum_{j=1}^{n-l+1} X_{lj})}{1 - \sum_{j=n-l+2}^n T_{\beta_j}(P_n)} = \frac{\sum_{j=1}^{n-l+1} E_{P_n}(X_{lj})}{1 - \sum_{j=n-l+2}^n \hat{\beta}_j} = \frac{\sum_{j=1}^{n-l+1} \hat{\alpha}_l \hat{\beta}_j}{1 - \sum_{j=n-l+2}^n \hat{\beta}_j} = \hat{\alpha}_l \frac{\sum_{j=1}^{n-l+1} \hat{\beta}_j}{\sum_{j=1}^{n-l+1} \hat{\beta}_j} = \hat{\alpha}_l$$

siden

$$1 - \sum_{j=n-l+2}^n \hat{\beta}_j = \sum_{j=1}^n \hat{\beta}_j - \sum_{j=n-l+2}^n \hat{\beta}_j = \sum_{j=1}^{n-l+1} \hat{\beta}_j$$

Tilsvarende får vi ved å bruke (3.3) og de samme antakelsene som over,

$$T_{\beta_{n-l+1}}(P_n) = \frac{E_{P_n}(\sum_{i=1}^l X_{i,n-l+1})}{\sum_{i=1}^l T_{\alpha_i}(P_n)} = \frac{\sum_{i=1}^l E_{P_n}(X_{i,n-l+1})}{\sum_{i=1}^l \hat{\alpha}_i} = \frac{\sum_{i=1}^l \hat{\alpha}_i \hat{\beta}_{n-l+1}}{\sum_{i=1}^l \hat{\alpha}_i} = \hat{\beta}_{n-l+1} \frac{\sum_{i=1}^l \hat{\alpha}_i}{\sum_{i=1}^l \hat{\alpha}_i} = \hat{\beta}_{n-l+1}$$

Vi har nå altså statistiske funksjonaler for α og β , med empirisk fordelingsfunksjon P_n , og det kan også vises at disse er Fisher konsistent, slik at vi får fra (3.2) og (3.3) at

$$T_{\alpha_l}(P) = \frac{E_P(\sum_{j=1}^{n-l+1} X_{lj})}{1 - \sum_{j=n-l+2}^n T_{\beta_j}(P)} = \alpha_l$$

$$T_{\beta_{n-l+1}}(P) = \frac{E_P(\sum_{i=1}^l X_{i,n-l+1})}{\sum_{i=1}^l T_{\alpha_i}(P)} = \beta_{n-l+1}$$

Som begge gjelder for alle $l = 1, \dots, n$.

Når vi nå har statistiske funksjonaler, kan vi gå videre for å finne influens funksjonene for α og β .

Vi skriver nå influens funksjonen som $IF([z, p, q]; T_{\alpha_l}, P)$, hvor denne sier noe om hvor mye en endring z i kravet $X_{p,q}$ påvirker T_{α_l} , der vi antar kravene $(X_{1,1}, \dots, X_{n,1})$ følger en

fordelingsfunksjon P , $(X_{1,1}, \dots, X_{n,1}) \sim P$, og $P_{i,j}$ er fordelingen til selve $X_{i,j}$. Vi antar $P_{i,j} \sim \text{Poisson}(\alpha_i \beta_j)$.

Fra definisjonen av influens funksjoner, får vi da

$$IF([z, p, q]; T_{\alpha_i}, P) = \lim_{\varepsilon \rightarrow 0} \frac{T_{\alpha_i}(P_{p,q,\varepsilon,z}) - T_{\alpha_i}(P)}{\varepsilon} = \frac{d}{d\varepsilon} T_{\alpha_i}(P_{p,q,\varepsilon,z}) \Big|_{\varepsilon=0}$$

$$IF([z, p, q]; T_{\beta_j}, P) = \lim_{\varepsilon \rightarrow 0} \frac{T_{\beta_j}(P_{p,q,\varepsilon,z}) - T_{\beta_j}(P)}{\varepsilon} = \frac{d}{d\varepsilon} T_{\beta_j}(P_{p,q,\varepsilon,z}) \Big|_{\varepsilon=0}$$

Der $X_{ij} \sim P_{p,q,\varepsilon,z}$ hvis

$$(4.4) \quad \begin{cases} X_{ij} \sim P_{ij} - \forall (i, j) \neq (p, q) \\ X_{ij} \sim (1 - \varepsilon)P_{ij} + \varepsilon z \end{cases}$$

Der $\varepsilon > 0, 1 \leq p \leq n$ og $1 \leq q \leq n$

$P_{p,q,\varepsilon,z}$ er altså den nye fordelingsfunksjonen, som er en blanding av P_{ij} og z med sannsynlighetene for dette hhv $1 - \varepsilon$ og ε .

Ifølge definisjonen av influensfunksjoner skal den nye fordelingsfunksjonen, ved en endring z i data, være en blanding av den tidligere fordelingsfunksjonen og endringen z . Hvis et krav $X_{i,j}$ følger fordelingsfunksjonen $P_{i,j}$ og vi gjør en endring i punktet (i,j) , så vil dette gjøre at $X_{i,j}$ nå følger den nye fordelingsfunksjonen $(1 - \varepsilon)P_{ij} + \varepsilon z$. Hvis vi gjør en endring i et annet punkt enn (i,j) , så vil $X_{i,j}$ fremdeles følge den samme marginale fordelingsfunksjonen $P_{i,j}$ som før.

For å finne influens funksjonene, trenger vi altså $T_{\alpha_i}(P_{p,q,\varepsilon,z})$ og $T_{\beta_j}(P_{p,q,\varepsilon,z})$. Vi ser først på $T_{\alpha_i}(P_{p,q,\varepsilon,z})$ og $T_{\beta_n}(P_{p,q,\varepsilon,z})$, altså tilfellet ved $i = 1$ og $j = n$.

Fra (3.2) over får vi da

$$T_{\alpha_i}(P_{p,q,\varepsilon,z}) = E_{P_{p,q,\varepsilon,z}} \left(\sum_{j=1}^n X_{1j} \right)$$

Som gir ved å bruke (4.4):

$$\begin{aligned}
\underline{p=1} \Rightarrow T_{\alpha_1}(P_{1,q,\varepsilon,z}) &= E_{P_{1,q,\varepsilon,z}} \left[\sum_{j=1}^n X_{1j} \right] \\
&= E_{P_{1,q,\varepsilon,z}} \left[\sum_{j=1, j \neq q}^n X_{1j} + X_{1q} \right] \\
&= \sum_{j=1, j \neq q}^n E_{P_{1,q,\varepsilon,z}} [X_{1j}] + E_{P_{1,q,\varepsilon,z}} [X_{1q}] \\
&= \sum_{j=1, j \neq q}^n \alpha_1 \beta_j + \alpha_1 \beta_q (1 - \varepsilon) + z\varepsilon \\
&= \sum_{j=1}^n \alpha_1 \beta_j - \alpha_1 \beta_q \varepsilon + z\varepsilon \\
&= T_{\alpha_1}(P) + \varepsilon(z - \alpha_1 \beta_q)
\end{aligned}$$

$$\begin{aligned}
\underline{p \neq 1} \Rightarrow T_{\alpha_1}(P_{p,q,\varepsilon,z}) &= E_{P_{p,q,\varepsilon,z}} \left[\sum_{j=1}^n X_{1j} \right] \\
&= E_{P_{p,q,\varepsilon,z}} \left[\sum_{j=1}^n X_{1j} \right] \\
&= T_{\alpha_1}(P)
\end{aligned}$$

Vi finner videre influens funksjonen ved å bruke (4.2):

$$\begin{aligned}
IF([z, p, q]; T_{\alpha_1}, P) &= \frac{d}{d\varepsilon} T_{\alpha_1}(P_{p,q,\varepsilon,z}) \Big|_{\varepsilon=0} \\
p=1 \Rightarrow \frac{d}{d\varepsilon} T_{\alpha_1}(P_{1,q,\varepsilon,z}) \Big|_{\varepsilon=0} &= z - \alpha_1 \beta_q \\
p \neq 1 \Rightarrow \frac{d}{d\varepsilon} T_{\alpha_1}(P_{p,q,\varepsilon,z}) \Big|_{\varepsilon=0} &= 0
\end{aligned}$$

Influens funksjonen for T_{α_1} er altså ubegrenset for skadeår 1 ($p=1$), noe som vil si at uteliggere her kan få stor innflytelse på α_1 .

Vi gjør nå det samme for $T_{\beta_n}(P_{p,q,\varepsilon,z})$, og får da ved (3.3):

$$T_{\beta_n}(P_{p,q,\varepsilon,z}) = \frac{E_{P_{p,q,\varepsilon,z}}(X_{1,n})}{T_{\alpha_1}(P_{p,q,\varepsilon,z})}$$

Som gir ved bruk av (4.4)

$$p=1, q=n \Rightarrow T_{\beta_n}(P_{1,n,\varepsilon,z}) = \frac{E_{P_{1,n,\varepsilon,z}}[(1-\varepsilon)X_{1,n} + \varepsilon z]}{T_{\alpha_1}(P_{1,n,\varepsilon,z})} = \frac{(1-\varepsilon)\alpha_1 \beta_n + \varepsilon z}{T_{\alpha_1}(P) + \varepsilon(z - \alpha_1 \beta_q)}$$

$$p = 1, q \neq n \Rightarrow T_{\beta_n}(P_{1,q,\varepsilon,z}) = \frac{E_{P_{1,q,\varepsilon,z}}[X_{1,n}]}{T_{\alpha_1}(P_{1,q,\varepsilon,z})} = \frac{\alpha_1 \beta_n}{T_{\alpha_1}(P) + \varepsilon(z - \alpha_1 \beta_q)}$$

$$p \neq 1 \Rightarrow T_{\beta_n}(P_{p,q,\varepsilon,z}) = \frac{E_{P_{p,q,\varepsilon,z}}[X_{1,n}]}{T_{\alpha_1}(P_{p,q,\varepsilon,z})} = \frac{\alpha_1 \beta_n}{T_{\alpha_1}(P)} = \beta_n$$

Vi finner så influens funksjonen:

$$IF([z, p, q]; T_{\beta_n}, P) = \frac{d}{d\varepsilon} T_{\beta_n}(P_{p,q,\varepsilon,z}) \Big|_{\varepsilon=0}$$

Som gir følgende:

$$p = 1, q = n \Rightarrow \frac{d}{d\varepsilon} T_{\beta_n}(P_{1,n,\varepsilon,z}) \Big|_{\varepsilon=0} = \frac{d}{d\varepsilon} \frac{(1-\varepsilon)\alpha_1\beta_n + \varepsilon z}{T_{\alpha_1}(P) + \varepsilon(z - \alpha_1\beta_q)} \Big|_{\varepsilon=0} = \frac{(1-\beta_n)(z - \alpha_1\beta_n)}{\alpha_1}$$

$$p = 1, q \neq n \Rightarrow \frac{d}{d\varepsilon} T_{\beta_n}(P_{1,q,\varepsilon,z}) \Big|_{\varepsilon=0} = \frac{d}{d\varepsilon} \frac{\alpha_1\beta_q}{T_{\alpha_1}(P) + \varepsilon(z - \alpha_1\beta_q)} \Big|_{\varepsilon=0} = \frac{-\beta_q(z - \alpha_1\beta_q)}{\alpha_1}$$

$$p \neq 1 \Rightarrow \frac{d}{d\varepsilon} T_{\beta_n}(P_{p,q,\varepsilon,z}) \Big|_{\varepsilon=0} = 0$$

Vi ser at influens funksjonen for skadeår 1 ($p=1$) og utviklingsår n ($q=n$) er ubegrenset, og uteliggere i dette punktet kan få stor innflytelse på β_n . For de andre utviklingsårene ($q \neq n$), ser vi at en økning i z vil føre til at β_n blir mindre, men fremdeles ubegrenset. Her må vi huske på at dette gjelder influens funksjonen for innflytelsen et punkt (p,q) har på β_n , der β_n altså sier oss noe om hvor stor andel av de totale kravene er fordelt mellom utviklingsårene. Det gir altså mening at influens funksjonen gir økt negativ bidrag når z økes, ettersom andelen av krav som ikke hører til utviklingsår n øker, og da må andelen som tilhører utviklingsår n bli mindre.

Vi har nå altså funnet influens funksjoner for T_{α_l} og T_{β_n} som følge av en endring z i punktet (p,q) . Vi ønsker videre å se på influens funksjoner for alle T_{α_l} og T_{β_j} . Vi gjør som vi har gjort tidligere, og antar at for $l < n$ har vi allerede funnet estimater for

$T_{\alpha_1}(P_{p,q,\varepsilon,z}), \dots, T_{\alpha_{l-1}}(P_{p,q,\varepsilon,z})$ og $T_{\beta_{n-l+2}}(P_{p,q,\varepsilon,z}), \dots, T_{\beta_n}(P_{p,q,\varepsilon,z})$. Vi bruker nå (3.2) og (3.3) for å finne uttrykk for $T_{\alpha_l}(P_{p,q,\varepsilon,z})$ og $T_{\beta_{n-l+1}}(P_{p,q,\varepsilon,z})$.

Ser først på $T_{\alpha_l}(P_{p,q,\varepsilon,z})$. Ved å bruke (3.2) kan vi da skrive

$$T_{\alpha_l}(P_{p,q,\varepsilon,z}) = \frac{E_{P_{p,q,\varepsilon,z}} \left[\sum_{j=1}^{n-l+1} X_{lj} \right]}{1 - \sum_{j=n-l+2}^n T_{\beta_j}(P_{p,q,\varepsilon,z})}, \forall l = 1, \dots, n$$

Og vi får da ved å bruke (4.4)

$$p = 1 \Rightarrow T_{\alpha_1}(P_{1,q,\varepsilon,z}) = \frac{E_{P_{1,q,\varepsilon,z}} \left[\sum_{j=1, j \neq q}^{n-l+1} X_{lj} + (1-\varepsilon)X_{lq} + \varepsilon z \right]}{1 - \sum_{j=n-l+2}^n T_{\beta_j}(P_{1,q,\varepsilon,z})}$$

$$= \frac{E_{P_{1,q,\varepsilon,z}} \left[\sum_{j=1}^{n-l+1} X_{lj} - \varepsilon X_{lq} + \varepsilon z \right]}{1 - \sum_{j=n-l+2}^n T_{\beta_j}(P_{1,q,\varepsilon,z})}$$

$$= \frac{\alpha_1 \sum_{j=1}^{n-l+1} \beta_j + \varepsilon z - \varepsilon \alpha_1 \beta_q}{1 - \sum_{j=n-l+2}^n T_{\beta_j}(P_{1,q,\varepsilon,z})}$$

$$p \neq 1 \Rightarrow T_{\alpha_1}(P_{p,q,\varepsilon,z}) = \frac{E_{P_{p,q,\varepsilon,z}} \left[\sum_{j=1}^{n-l+1} X_{lj} \right]}{1 - \sum_{j=n-l+2}^n T_{\beta_j}(P_{p,q,\varepsilon,z})}$$

$$= \frac{\alpha_1 \sum_{j=1}^{n-l+1} \beta_j}{1 - \sum_{j=n-l+2}^n T_{\beta_j}(P_{p,q,\varepsilon,z})}$$

Vi har nå $T_{\alpha_l}(P_{p,q,\varepsilon,z})$, og kan videre finne influens funksjonen til denne estimatoren som

$$IF([z, p, q]; T_{\alpha_l}, P) = \frac{d}{d\varepsilon} T_{\alpha_l}(P_{p,q,\varepsilon,z}) \Big|_{\varepsilon=0}$$

Vi får da:

$$\begin{aligned}
p=l &\Rightarrow \frac{d}{d\varepsilon} T_{\alpha_l}(P_{p,q,\varepsilon,z})|_{\varepsilon=0} = \frac{d}{d\varepsilon} \frac{\alpha_l \sum_{j=1}^{n-l+1} \beta_j + \varepsilon z - \varepsilon \alpha_l \beta_q}{1 - \sum_{j=n-l+2}^n T_{\beta_j}(P_{l,q,\varepsilon,z})} |_{\varepsilon=0} \\
&= \frac{(z - \alpha_l \beta_q)(1 - \sum_{j=n-l+2}^n T_{\beta_j}(P_{l,q,\varepsilon,z})) + (\sum_{j=n-l+2}^n IF([z, p, q]; T_{\beta_j}, P))(\alpha_l \sum_{j=1}^{n-l+1} \beta_j + \varepsilon z - \varepsilon \alpha_l \beta_q)}{(1 - \sum_{j=n-l+2}^n T_{\beta_j}(P_{l,q,\varepsilon,z}))^2} |_{\varepsilon=0} \\
&= \frac{(z - \alpha_l \beta_q)(1 - \sum_{j=n-l+2}^n T_{\beta_j}(P)) + (\sum_{j=n-l+2}^n IF([z, p, q]; T_{\beta_j}, P))(\alpha_l \sum_{j=1}^{n-l+1} \beta_j)}{(1 - \sum_{j=n-l+2}^n T_{\beta_j}(P))^2} \\
&= \frac{(1 - \sum_{j=n-l+2}^n T_{\beta_j}(P))(z - \alpha_l \beta_q + \alpha_l \sum_{j=n-l+2}^n IF([z, p, q]; T_{\beta_j}, P))}{(1 - \sum_{j=n-l+2}^n T_{\beta_j}(P))^2}
\end{aligned}$$

Siden $T_{\beta_j}(P) = \beta_j$ (Fisher konsistent) og $\sum_{j=1}^{n-l+1} \beta_j = (1 - \sum_{j=n-l+2}^n \beta_j)$

Og vi får da for $p=l$

$$IF([z, p, q]; T_{\alpha_l}, P) = \frac{(z - \alpha_l \beta_q + \alpha_l \sum_{j=n-l+2}^n IF([z, p, q]; T_{\beta_j}, P))}{(1 - \sum_{j=n-l+2}^n T_{\beta_j}(P))}$$

I motsatt tilfelle, får vi for $p \neq l$

$$\begin{aligned}
\frac{d}{d\varepsilon} T_{\alpha_l}(P_{p,q,\varepsilon,z}) \Big|_{\varepsilon=0} &= \frac{d}{d\varepsilon} \frac{\alpha_l \sum_{j=1}^{n-l+1} \beta_j}{1 - \sum_{j=n-l+2}^n T_{\beta_j}(P_{p,q,\varepsilon,z})} \Big|_{\varepsilon=0} \\
&= \frac{(\alpha_l \sum_{j=1}^{n-l+1} \beta_j) (\sum_{j=n-l+2}^n IF([z, p, q]; T_{\beta_j}, P))}{(1 - \sum_{j=n-l+2}^n T_{\beta_j}(P_{p,q,\varepsilon,z}))^2} \Big|_{\varepsilon=0} \\
&= \frac{\alpha_l (1 - \sum_{j=n-l+2}^n \beta_j) (\sum_{j=n-l+2}^n IF([z, p, q]; T_{\beta_j}, P))}{(1 - \sum_{j=n-l+2}^n T_{\beta_j}(P))^2} \\
&= \frac{\alpha_l (\sum_{j=n-l+2}^n IF([z, p, q]; T_{\beta_j}, P))}{(1 - \sum_{j=n-l+2}^n T_{\beta_j}(P))}
\end{aligned}$$

Vi gjør tilsvarende for å finne $IF([z, p, q]; T_{\beta_{n-l+1}}, P)$, og starter først med å se på

$T_{\beta_{n-l+1}}(P_{p,q,\varepsilon,z})$ ved å bruke (3.3).

Vi får da

$$T_{\beta_{n-l+1}}(P_{p,q,\varepsilon,z}) = \frac{E_{P_{p,q,\varepsilon,z}} [\sum_{i=1}^l X_{i,n-l+1}]}{\sum_{i=1}^l T_{\alpha_i}(P_{p,q,\varepsilon,z})}, \forall l = 1, \dots, n$$

For $q = n - l + 1$ får vi da ved (4.4)

$$T_{\beta_{n-l+1}}(P_{p,q,\varepsilon,z}) = \frac{E_{P_{p,q,\varepsilon,z}} [\sum_{i=1, i \neq p}^1 X_{i,n-l+1} + (1-\varepsilon)X_{p,n-l+1} + \varepsilon Z]}{\sum_{i=1}^1 T_{\alpha_i}(P_{p,q,\varepsilon,z})}$$

$$\begin{aligned}
&= \frac{E_{P_{p,q,\varepsilon,z}} \left[\sum_{i=1}^1 X_{i,n-l+1} - \varepsilon X_{p,n-l+1} + \varepsilon z \right]}{\sum_{i=1}^1 T_{\alpha_i}(P_{p,q,\varepsilon,z})} \\
&= \frac{\beta_{n-l+1} \sum_{i=1}^1 \alpha_i - \varepsilon \alpha_p \beta_{n-l+1} + \varepsilon z}{\sum_{i=1}^1 T_{\alpha_i}(P_{p,q,\varepsilon,z})}
\end{aligned}$$

Og for $q \neq n-l+1$

$$\begin{aligned}
T_{\beta_{n-l+1}}(P_{p,q,\varepsilon,z}) &= \frac{E_{P_{p,q,\varepsilon,z}} \left[\sum_{i=1}^1 X_{i,n-l+1} \right]}{\sum_{i=1}^1 T_{\alpha_i}(P_{p,q,\varepsilon,z})} \\
&= \frac{\beta_{n-l+1} \sum_{i=1}^1 \alpha_i}{\sum_{i=1}^1 T_{\alpha_i}(P_{p,q,\varepsilon,z})}
\end{aligned}$$

Influens funksjonen for $T_{\beta_{n-l+1}}(P_{p,q,\varepsilon,z})$ finner vi på lik linje som over, nemlig

$$IF([z, p, q]; T_{\beta_{n-l+1}}, P) = \frac{d}{d\varepsilon} T_{\beta_{n-l+1}}(P_{p,q,\varepsilon,z}) \Big|_{\varepsilon=0}$$

For $q = n-l+1$ får vi da

$$\begin{aligned}
IF([z, p, q]; T_{\beta_{n-l+1}}, P) &= \frac{d}{d\varepsilon} \frac{\beta_{n-l+1} \sum_{i=1}^l \alpha_i - \varepsilon \alpha_p \beta_{n-l+1} + \varepsilon z}{\sum_{i=1}^l T_{\alpha_i}(P_{p,q,\varepsilon,z})} \Big|_{\varepsilon=0} \\
&= \frac{(z - \alpha_p \beta_{n-l+1}) (\sum_{i=1}^l T_{\alpha_i}(P_{p,q,\varepsilon,z})) - (\beta_{n-l+1} \sum_{i=1}^l \alpha_i - \varepsilon \alpha_p \beta_{n-l+1} + \varepsilon z) (\sum_{i=1}^l IF([z, p, q]; T_{\alpha_i}, P))}{(\sum_{i=1}^l T_{\alpha_i}(P_{p,q,\varepsilon,z}))^2} \Big|_{\varepsilon=0} \\
&= \frac{(z - \alpha_p \beta_{n-l+1}) (\sum_{i=1}^l T_{\alpha_i}(P)) - (\beta_{n-l+1} \sum_{i=1}^l \alpha_i) (\sum_{i=1}^l IF([z, p, q]; T_{\alpha_i}, P))}{(\sum_{i=1}^l T_{\alpha_i}(P))^2} \\
&= \frac{(z - \alpha_p \beta_{n-l+1}) - \beta_{n-l+1} \sum_{i=1}^l IF([z, p, q]; T_{\alpha_i}, P)}{\sum_{i=1}^l T_{\alpha_i}(P)}
\end{aligned}$$

Og for $q \neq n-l+1$ får vi

$$\begin{aligned}
IF([z, p, q]; T_{\beta_{n-l+1}}, P) &= \frac{d}{d\varepsilon} \frac{\beta_{n-l+1} \sum_{i=1}^l \alpha_i}{\sum_{i=1}^l T_{\alpha_i}(P_{p,q,\varepsilon,z})} \Big|_{\varepsilon=0} \\
&= \frac{-(\beta_{n-l+1} \sum_{i=1}^l \alpha_i) (\sum_{i=1}^l IF([z, p, q]; T_{\alpha_i}, P))}{(\sum_{i=1}^l T_{\alpha_i}(P_{p,q,\varepsilon,z}))^2} \Big|_{\varepsilon=0} \\
&= \frac{-(\beta_{n-l+1} \sum_{i=1}^l \alpha_i) (\sum_{i=1}^l IF([z, p, q]; T_{\alpha_i}, P))}{(\sum_{i=1}^l T_{\alpha_i}(P))^2} \\
&= \frac{-\beta_{n-l+1} \sum_{i=1}^l IF([z, p, q]; T_{\alpha_i}, P)}{\sum_{i=1}^l T_{\alpha_i}(P)}
\end{aligned}$$

Mens når $p \neq l$, vil $T_{\beta_{n-l+1}}(P_{p,q,\varepsilon,z}) = \beta_{n-l+1}$, og influensfunksjon blir da lik null.

Vi kan nå oppsummere det vi har gjort, og kommer frem til følgende uttrykk for influensfunksjonene:

$$(4.5) \quad \text{IF}([z, p, q]; T_{\alpha_1}, P) = \begin{cases} \frac{(z - \alpha_1 \beta_q + \alpha_1 \sum_{j=n-l+2}^n \text{IF}([z, l, q]; T_{\beta_j}, P))}{(1 - \sum_{j=n-l+2}^n \beta_j)}, & p = l \\ \frac{\alpha_1 (\sum_{j=n-l+2}^n \text{IF}([z, p, q]; T_{\beta_j}, P))}{(1 - \sum_{j=n-l+2}^n \beta_j)}, & p \neq l \end{cases}$$

$$(4.6) \quad \text{IF}([z, p, q]; T_{\beta_{n-l+1}}, P) = \begin{cases} \frac{z - \alpha_p \beta_{n-l+1} - \beta_{n-l+1} \sum_{i=1}^l \text{IF}([z, p, n-l+1]; T_{\alpha_i}, P)}{\sum_{i=1}^l \alpha_i}, & q = n-l+1 \\ \frac{-\beta_{n-l+1} \sum_{i=1}^l \text{IF}([z, p, q]; T_{\alpha_i}, P)}{\sum_{i=1}^l \alpha_i}, & q \neq n-l+1 \end{cases}$$

Uttrykkene over gir oss influensen på estimatorene for α og β ved en endring z av kravet i punkt (p, q) . At influens funksjonen for α_l og β_{n-l+1} er ubegrenset når $p = l$ og når $q = n-l+1$, leser vi greit ut av uttrykkene, ettersom når z øker vil verdien til influensfunksjonen også øke. For $p \neq l$ og for $q \neq n-l+1$ er det ikke så lett å se direkte ut av uttrykkene når influens funksjonene er ubegrenset, men nærmere studie gir at uttrykkene for T_{α_l} og $T_{\beta_{n-l+1}}$ er ubegrenset når $l \geq p$.

Når vi har influens funksjoner for funksjonalene T_{α_l} og $T_{\beta_{n-l+1}}$, kan vi også finne influens funksjonene for $T_{\alpha_i \beta_j}$, $T_{\sum_{j=n-i+2}^n \alpha_i \beta_j}$ og $T_{\sum_{i=2}^n \sum_{j=n-i+2}^n \alpha_i \beta_j}$, som da vil være influensfunksjonen for hhv.

utestående krav for punkt (i, j) , utestående krav for skadeår i , R_i , og samlet utestående krav, eller total reserve, R .

Influens funksjonene for disse funksjonalene vil da bli som under, [7], der utledning ligger i appendix B.

$$(4.7) \quad \text{IF}([z, p, q]; T_{\alpha_i \beta_j}, P) = \alpha_i \text{IF}([z, p, q]; T_{\beta_j}, P) + \beta_j \text{IF}([z, p, q]; T_{\alpha_i}, P)$$

$$(4.8) \quad \text{IF}([z, p, q]; T_{\sum_{j=n-i+2}^n \alpha_i \beta_j}, P) = \sum_{j=n-i+2}^n \text{IF}([z, p, q]; T_{\alpha_i \beta_j}, P)$$

$$(4.9) \quad \text{IF}([z, p, q]; T_{\sum_{i=2}^n \sum_{j=n-i+2}^n \alpha_i \beta_j}, P) = \sum_{i=2}^n \text{IF}([z, p, q]; T_{\sum_{j=n-i+2}^n \alpha_i \beta_j}, P)$$

4.3 Influens funksjon for Bornhuetter-Ferguson

I tillegg til Chain-Ladder metoden, er Bornhuetter-Ferguson (BF) metoden også en mye brukt deterministisk reserveringsmetode i forsikringsbransjen. Jeg tar nå for meg influens funksjoner for denne metoden også, for å vise at denne i likhet med CL heller ikke er robust.

I Bornhuetter-Ferguson metoden, Neuhaus [9], antar vi at det er en konstant skade intensitet over tid, i motsetning til CL metoden hvor dette ikke er tilfellet. CL metoden kan underbygges med at det antas $X_{i,j} \sim \text{Poisson}(\alpha_i \beta_j)$. Hvis vi antar at det i α_i inngår både eksponering og skade intensitet, kan vi i stedet skrive $X_{i,j} \sim \text{Poisson}(p_i \theta_i \beta_j)$, der $\alpha_i = p_i \theta_i$.

p_i - apriori verdi – kjent eksponering for skadeår i

θ_i - skadeintensitet skadeår i

β_j - andel av total krav rapportert utviklingsår j

For CL modellerer vi altså eksponering og skadeintensitet sammen, som en parameter.

For BF kan vi da la $X_{i,j} \sim \text{Poisson}(p_i \theta_i \beta_j)$, altså har vi en konstant skadeintensitet. På lik linje som for CL metoden, kan vi nå finne influens funksjoner for metoden.

Vi finner da følgende MLE estimatorer for θ og β_j :

$$\hat{\theta} = \sum_{j=1}^n \frac{\sum_{i=1}^{n-j+1} X_{i,j}}{\sum_{i=1}^{n-j+1} p_i} \quad \hat{\beta}_j = \frac{\sum_{i=1}^{n-j+1} X_{i,j}}{\hat{\theta} \sum_{i=1}^{n-j+1} p_i}$$

Som da gir oss følgende estimat for IBNR krav i trekanten,
 Når vi har $i=1, \dots, n$ og $j = 1, \dots, n$

$$(4.10) \quad \hat{X}_{i,j} = p_i \hat{\theta} \hat{\beta}_j = p_i \hat{\theta} \frac{\sum_{i=1}^{n-j+1} X_{i,j}}{\sum_{i=1}^{n-j+1} p_i} = p_i \frac{\sum_{i=1}^{n-j+1} X_{i,j}}{\sum_{i=1}^{n-j+1} p_i}, \quad \text{for } i+j > n+1$$

Ved å bruke samme fremgangsmåte som for influens funksjonen for Chain-Ladder, kommer vi da frem til følgende uttrykk for influens funksjonen for Bornhuetter-Ferguson, der utledning ligger i Appendix. (Uttrykkene sier oss, som tidligere, hvor stor influens en endring z i krav $X_{p,q}$ får på estimatorene θ og β_j)

$$IF([z, p, q]; T_\theta, P) = \frac{z - p_p \hat{\theta} \hat{\beta}_q}{\sum_{i=1}^{n-q+1} p_i}$$

$$IF([z, p, q]; T_{\beta_j}, P) = \begin{cases} \frac{(1 - \hat{\beta}_q)}{\hat{\theta}} \cdot IF([z, p, q]; T_\theta, P), & j = q \\ \frac{-\hat{\beta}_j}{\hat{\theta}} \cdot IF([z, p, q]; T_\theta, P), & j \neq q \end{cases}$$

Som vi ville tenkt oss på forhånd er influens funksjonen for Bornhuetter-Ferguson metoden også ubegrenset, noe som gjør at uteliggere kan få stor innflytelse på reserven her også.

4.4 Influens analyse for datasett – Chain-Ladder

Jeg har videre beregnet verdien av influens funksjonene for et av datasettene (tabell 7.9) jeg har sett på, etter Chain-Ladder metoden. For Bornhuetter-Ferguson metoden er vi avhengig av en a priori verdi for å gjøre dette, noe jeg ikke har tilgang til. En forenkling kunne vært og satt denne a priori verdien lik for alle skadeårene (for eksempel 1000), men dette vil igjen føre til lik innfluens på den totale reserven for alle utviklingsårene. Ettersom det er en robust Chain-Ladder metode jeg tar for meg i neste kapittel, har jeg dermed kun valgt å se på influensen for Chain-Ladder her.

Jeg velger å se på selve influensen som hvert av kravene har på den totale reserven R , (4.9). I stedet for å se på selve verdien til influens funksjonene, har jeg valgt å se på verdien i prosent av den totale reserven med den klassiske metoden. Dette gir en bedre forståelse av hvilken influens hvert av kravene har på den totale reserven, kontra det å se på selve verdien av influens funksjonen.

Når influens funksjonene for datasettet skal beregnes, har jeg brukt robuste estimater på parameterne α_i og β_{n-i+1} , estimert ved robust GLM metode i R. Jeg bruker robuste estimater i stedet for klassiske estimater, ettersom klassiske estimater kan maskere eventuelle uteliggende krav som allerede har påvirket estimatene, [13].

Når det gjelder valg av z -verdi i formelen til influens funksjonene, har jeg valgt å se på ulike z -verdier. Jeg har derfor tatt med en oversikt over beregninger med $z = \alpha_i \beta_j \cdot 1.1$, $z = \alpha_i \beta_j \cdot 2$ og $z = \alpha_i \beta_j \cdot 10$. Verdien av influens funksjonen når $z = \alpha_i \beta_j \cdot 1.1$, gir oss altså endringen i estimatet (total reserve i dette tilfellet), når vi øker kravet i posisjon (i,j) med 10%. Ved å la $z = \alpha_i \beta_j$ gir dette altså oss ingen endring, noe som gir oss at influens funksjonen er lik 0 for alle krav (ingen *perturbering* i fordelingen til kravene, se (4.4)).

		Utviklingsår									
		1	2	3	4	5	6	7	8	9	10
S k a d e å r	1	-0,53 %	-0,54 %	-0,36 %	-0,16 %	0,01 %	0,08 %	0,18 %	0,21 %	0,65 %	0,47 %
	2	-0,69 %	-0,65 %	-0,39 %	-0,10 %	0,09 %	0,17 %	0,31 %	0,33 %	0,99 %	
	3	-0,58 %	-0,44 %	-0,17 %	0,13 %	0,24 %	0,27 %	0,41 %	0,40 %		
	4	-0,44 %	-0,29 %	-0,05 %	0,20 %	0,26 %	0,27 %	0,39 %			
	5	-0,38 %	-0,18 %	0,08 %	0,34 %	0,35 %	0,33 %				
	6	-0,33 %	-0,06 %	0,22 %	0,49 %	0,45 %					
	7	-0,23 %	0,16 %	0,49 %	0,79 %						
	8	0,08 %	0,79 %	1,21 %							
	9	0,65 %	1,64 %								
	10	2,44 %									

Tabell 4.1

Tabell 4.1 viser influensen i prosent av den totale reserven, beregnet etter den klassiske metoden, som hvert av kravene i trekanten har på den totale reserven når $z = \alpha_i \beta_j \cdot 1.1$. Vi ser at dette ikke gir de store utslagene, men vi legger merke til at det har dannet seg et mønster der vi får negativ innflytelse i øvre venstre hjørne, og innflytelsen på den totale reserven øker når vi nærmer oss de to andre hjørnene. Hjørnepunktet øverst til venstre kan altså virke litt merkelig ved første øyekast, ettersom det vanligvis gir negativt bidrag til den totale reserven. Årsaken til det negative bidraget er rett og slett at hvis man øker kravet her, vil de kumulative kravene for utviklingsårene fremover også økes, uten å påvirke de inkrementelle kravene fremover. Dette gjør at utviklingsfaktorene blir redusert, noe som gir lavere reserve, og dermed den negative innflytelsen.

Gir så verdiene for influens funksjonen i dette tilfellet mening? La oss se på influensen fra krav (1,1) i dette tilfellet, som gir en influens på -0,53 % på den totale reserven når vi øker dette kravet med 10 %. Dette høres veldig lite ut, men ser vi på datasettet dette er gjort på (tabell 7.9), ser vi at 10 % av kravet i posisjon (1,1) tilsvarer ca 0,2 % av den totale reserven. Og siden utviklingsfaktorene blir redusert som følge av økningen i dette kravet, gir en negativ influens på totalt reserve på -0,53 % altså mening.

		Utviklingsår									
		1	2	3	4	5	6	7	8	9	10
S k a d e å r	1	-5,32 %	-5,35 %	-3,64 %	-1,63 %	0,06 %	0,82 %	1,80 %	2,08 %	6,50 %	4,67 %
	2	-6,93 %	-6,48 %	-3,93 %	-1,04 %	0,90 %	1,71 %	3,11 %	3,32 %	9,85 %	
	3	-5,75 %	-4,40 %	-1,66 %	1,26 %	2,38 %	2,71 %	4,10 %	3,98 %		
	4	-4,38 %	-2,87 %	-0,49 %	1,99 %	2,60 %	2,68 %	3,86 %			
	5	-3,82 %	-1,76 %	0,80 %	3,37 %	3,51 %	3,31 %				
	6	-3,27 %	-0,59 %	2,19 %	4,88 %	4,53 %					
	7	-2,30 %	1,65 %	4,91 %	7,91 %						
	8	0,83 %	7,89 %	12,10 %							
	9	6,50 %	16,40 %								
	10	24,43 %									

Tabell 4.2

		Utviklingsår									
		1	2	3	4	5	6	7	8	9	10
S k a d e å r	1	-47,86 %	-48,17 %	-32,74 %	-14,63 %	0,57 %	7,39 %	16,18 %	18,69 %	58,50 %	42,07 %
	2	-62,33 %	-58,29 %	-35,37 %	-9,40 %	8,11 %	15,42 %	27,96 %	29,88 %	88,69 %	
	3	-51,79 %	-39,60 %	-14,95 %	11,38 %	21,41 %	24,35 %	36,93 %	35,83 %		
	4	-39,39 %	-25,85 %	-4,39 %	17,95 %	23,38 %	24,10 %	34,71 %			
	5	-34,36 %	-15,81 %	7,19 %	30,30 %	31,59 %	29,83 %				
	6	-29,42 %	-5,27 %	19,68 %	43,92 %	40,78 %					
	7	-20,69 %	14,82 %	44,15 %	71,19 %						
	8	7,43 %	70,98 %	108,94 %							
	9	58,49 %	147,60 %								
	10	219,91 %									

Tabell 4.3

I tabell 4.2 har vi økt kravene med 100 %, $z = \alpha_i \beta_j \cdot 2$, mens i tabell 4.3 har vi økt med 1000 %. Dette gjør vi for å se hvordan influens til hvert av kravene øker, når z blir stor. Vi ser at mønsteret er lik som i tabell 4.1, og vi ser tydelig hvilke krav som har stor influens på den totale reserven. Forståelig nok er det slik at krav som befinner seg lengst ut i hjørnene i trekanten vil ha størst innflytelse på reserven. For eksempel vil kravet øverst i høyre hjørne påvirke utviklingsfaktoren for alle skadeår i det siste utviklingsåret, og blir dermed veldig innflytelsesrik. Kravet nederst i venstre hjørne vil ikke påvirke noen av utviklingsfaktorene,

men derimot vil siste skadeåret bli utviklet for alle utviklingsårene, noe som også gir stor innflytelse. Alle krav blir altså mer og mer innflytelsesrike jo lengre ut i hjørnene man befinner seg.

Det som er interessant å legge merke til, er krav hvor innflytelsen fra et krav er større (mindre) enn kravene som befinner seg et skadeår/utviklingsår nærmere aktuelt hjørne, når man går opp (ned) i aktuelt kalenderår. For eksempel fra tabell 4.3 legger vi merke til krav fra punkt(2,9) som gir større innflytelse på den totale reserven enn kravet i (1,10). Dette tyder på at kravet i punkt (2,9) kanskje er en uteliggere, og bør undersøkes nærmere? Vi vil i kapittel 7 se at ifølge den robuste Chain-Ladder metoden jeg har sett på, vil kravet i punkt(2,9) ikke betraktes som uteligger.

Jeg har også beregnet influens funksjonen for et datasett (tabell 7.7) hvor det er tydelig at det forekommer uteliggere i datasettet.

		Utviklingsår									
		1	2	3	4	5	6	7	8	9	10
S k a d e å r	1	-4,35 %	-1,33 %	-0,79 %	-0,43 %	-0,17 %	0,04 %	0,26 %	0,40 %	1,58 %	4,78 %
	2	-5,85 %	-1,55 %	-0,60 %	0,03 %	0,37 %	0,89 %	1,67 %	2,00 %	3,29 %	
	3	-41,84 %	-9,96 %	-2,10 %	3,13 %	5,47 %	10,41 %	18,26 %	21,27 %		
	4	-1,35 %	-0,21 %	0,15 %	0,39 %	0,46 %	0,74 %	1,22 %			
	5	-0,78 %	0,25 %	0,82 %	1,20 %	1,21 %	1,77 %				
	6	1,01 %	1,32 %	2,17 %	2,72 %	2,59 %					
	7	3,03 %	2,03 %	2,72 %	3,17 %						
	8	8,58 %	4,49 %	5,26 %							
	9	18,41 %	8,57 %								
	10	23,13 %									

Tabell 4.4

Tabell 4.4 viser influensen på total reserve fra datasettet 7.7, ved $z = \alpha_i \beta_j \cdot 2$ (100 % økning i angitt krav). Vi ser tydelig at det er i skadeår 3 det befinner seg uteliggere. Merk spesielt krav (3,1) som gir en betydelig negativ influens i dette tilfellet.

Oppsummering influens funksjoner

Med influens funksjoner til Chain-Ladder metoden, har vi altså vist at influens funksjonen er ubegrenset, og at uteliggere kan få stor innflytelse på estimatene som estimeres med den klassiske metoden. Vi har videre vist at ved å beregne influens funksjonene for datasett vi ønsker å studere, kan vi få et inntrykk av hvilke krav som er mest innflytelsesrike, og dermed hvilke krav som er aktuelle uteliggere i trekanten vi ser på.

5. Robust Chain-Ladder metode

Fremt til nå har vi sett på hvordan Chain-Ladder metoden ikke er robust, og at uteliggere i datasettet for beregning av IBNR reserve ved den klassiske Chain-Ladder metoden kan gi store utslag i reserven.

Jeg vil nå ta for meg fremgangsmåten i artikkelen "A robustification of the Chain-Ladder method", skrevet av Tim Verdonck, Martine Van Wouwe og Jan Dhaene, [1].

5.1 Robust Chain-Ladder

Den Robuste Chain-Ladder metoden som Verdonck, Van Wouwe og Dhaene foreslår, er i hovedsak delt opp i 5 steg som jeg har tenkt å ta for meg nå.

Metoden baserer seg på den klassiske Chain-Ladder metoden, som videre blir modifisert for å få en robust metode.

Det vi ønsker med en robust Chain-Ladder metode er først og fremst å oppdage eventuelle uteliggere i datasettet, og videre modifisere datasettet slik at vi får et estimat på hva reserven ville vært uten disse eventuelle uteliggerne i datasettet. Vi ønsker ikke å erstatte den klassiske CL-metoden, men bruke metoden som et hjelpeverktøy i jakten på et beste estimat av reserven.

Siden Chain-Ladder metoden baserer seg på akkumulerte krav, vil en uteligger i første kolonne påvirke alle utviklingsfaktorene. Dette kan medføre at estimatet på reserven blir for stort / for lite i forhold til hva den egentlig burde ha vært. Som en løsning på dette, foreslår Verdonck å basere utviklingsfaktorene på inkrementelle krav i stedet, noe som gjør at en uteligger ikke kan påvirke mer enn maksimalt to utviklingsfaktorer. Altså hvis det befinner seg en uteligger i utviklingsår j , er det kun utviklingsfaktoren f_j og f_{j+1} som vil påvirkes av uteliggeren i dette tilfellet.

Verdonck foreslår da følgende i sin artikkel,

$$(5.1) \quad \hat{f}_j = \text{median} \left(\frac{X_{i,j}}{X_{i,j-1}} \mid i = 1, \dots, n - j + 1 \right) \quad 2 \leq j \leq n$$

Utviklingsfaktorene basert på denne måten vil brukes for å gjøre datasettet robust, før den klassiske Chain-Ladder metoden tas i bruk. Det er altså ikke ment å erstatte beregningen av de ordnære utviklingsfaktorene i selve estimeringen, men kun bruke de alternative utviklingsfaktorene når datasettet gjøres robust.

Det vi ønsker er altså å undersøke om det befinner seg uteliggere i datasettet vårt. Finner vi uteliggere, må datasettet modifiseres slik at det ikke lenger inneholder uteligger, før vi tilslutt kan gjøre vanlig Chain-Ladder estimering på datasettet.

Jeg tar videre for meg de ulike stegene i den robuste Chain-Ladder metoden, og bruker data fra et av eksemplene i artikkelen til å forklare fremgangsmåten, Tabell 5.1. Resultatene fra bruk av den robuste metoden finnes igjen i kapittel 7.

		Utviklingsår									
		1	2	3	4	5	6	7	8	9	10
Skadeår	1	701 848	232 585	194 470	148 488	98 600	61 875	47 145	32 260	25 628	18 173
	2	1 864 592	856 348	441 065	256 385	139 112	103 032	62 855	47 355	33 132	
	3	1 152 332	2 381 638	2 545 868	2 613 448	2 310 415	2 712 015	3 662 850	3 704 750		
	4	966 722	168 570	149 128	140 050	38 410	9 548	12 308			
	5	789 602	485 170	192 082	149 400	140 052	43 518				
	6	1 154 888	475 018	619 605	330 220	91 025					
	7	1 053 622	459 830	419 665	273 385						
	8	1 956 875	368 372	244 525							
	9	1 568 152	966 498								
	10	1 322 485									

Tabell 5.1 – inkrementelle krav

Oversikt over de ulike stegene i metoden, [1]

Step 1

- Compute development factors

$$\tilde{r}_j = \text{median} \left\{ \frac{C_j}{C_{j-1}} \mid j = 1, \dots, n - j + 1 \right\} \quad 2 \leq j \leq n.$$

- Obtain Pearson residuals, r_{ij} as in England and Verrall (1999).

Step 2

- Test for normality.
- Apply classical boxplot rejection rule on residuals.

Step 3

- If outlier in first column; suppose r_{k1}
 - If r_{k2} is not outlying,

$$C_{k1} = \frac{X_{k2}}{\text{median} \left\{ \frac{X_{i2}}{X_{i1}} \mid i = 1, \dots, n - 1 \right\}} \quad 2 \leq k \leq n.$$

- If r_{k2} is outlying,

$$C_{k1} = \text{median} \{C_{i1} \mid i = 1, \dots, n\}.$$

Step 4

- Compute development factors

$$\hat{r}_j = \text{median} \left\{ \frac{X_{ij}}{X_{i1}} \mid i = 1, \dots, n - j + 1 \right\} \quad 2 \leq j \leq n.$$

- Calculate fitted incremental claims

$$\hat{X}_{i,n-i+2}^j = X_{i1} \hat{r}_{n-i+2}^j \quad 1 \leq j \leq n, 2 \leq i \leq n.$$

- Obtain residuals, r_{ij}^1 as in (5.2).
- Apply classical boxplot rejection rule (after testing for normality).
- If outlier; suppose r_{ij}^1
 - $r_{ij}^1 = \text{median} \{r_{ij}^1 \mid i = 1, \dots, n - 1; j = 2, \dots, n - i + 1\}$.
 - Backtransform residuals r_{ij}^1 to data matrix X_{ij}^r .

Step 5

- Apply classical chain-ladder method on the robustified data X_{ij}^r .

Steg 1:

Det første som gjøres, er å tilpasse nye verdier (fitted values) til dataene våre, som vi kan sammenligne med de originale data, og dermed finne uteliggere ut ifra residualene vi finner her.

Måten vi gjør dette på, er å basere oss på kumulative data. Dette fordi at de kumulative kravene i dag tilsvarer de tilpassede kravene i dag. Mao, den siste diagonalen av de observerte data kan overføres til den siste diagonalen tilhørende de tilpassede kumulative data.

		Utviklingsår									
		1	2	3	4	5	6	7	8	9	10
Skadeår	1	701 848	934 433	1 128 903	1 277 391	1 375 991	1 437 866	1 485 011	1 517 271	1 542 899	1 561 072
	2	1 864 592	2 720 940	3 162 005	3 418 390	3 557 502	3 660 534	3 723 389	3 770 744	3 803 876	
	3	1 152 332	3 533 970	6 079 838	8 693 286	11 003 701	13 715 716	17 378 566	21 083 316		
	4	966 722	1 135 292	1 284 420	1 424 470	1 462 880	1 472 428	1 484 736			
	5	789 602	1 274 772	1 466 854	1 616 254	1 756 306	1 799 824				
	6	1 154 888	1 629 906	2 249 511	2 579 731	2 670 756					
	7	1 053 622	1 513 452	1 933 117	2 206 502						
	8	1 956 875	2 325 247	2 569 772							
	9	1 568 152	2 534 650								
	10	1 322 485									

Tabell 5.2

		Utviklingsår									
		1	2	3	4	5	6	7	8	9	10
Skadeår	1	693 056	995 525	1 179 804	1 334 987	1 413 674	1 454 616	1 490 952	1 523 341	1 542 899	1 561 072
	2	1 708 665	2 454 374	2 908 699	3 291 288	3 485 283	3 586 224	3 675 806	3 755 658	3 803 876	
	3	9 592 015	13 778 238	16 328 699	18 476 461	19 565 499	20 132 153	20 635 046	21 083 316		
	4	690 166	991 374	1 174 885	1 329 421	1 407 780	1 448 552	1 484 736			
	5	857 531	1 231 781	1 459 793	1 651 804	1 749 165	1 799 824				
	6	1 309 342	1 880 776	2 228 922	2 522 099	2 670 756					
	7	1 145 501	1 645 429	1 950 011	2 206 502						
	8	1 509 569	2 168 387	2 569 772							
	9	1 764 551	2 534 650								
	10	1 322 485									

Tabell 5.3

Tabell 5.2 er de kumulative dataene fra datasettet, mens tabell 5.3 er tilpassede data fra datasettet. Vi ser at den siste diagonalen er lik for kumulative data og tilpassede kumulative data, slik som beskrevet over.

Måten resten av verdiene i de tilpassede kumulative data er laget, er å bruke baklengs rekursiv deling av de kumulative tilpassede verdiene ved tid t , på utviklingsfaktoren ved tid $t-1$.

Utviklingsfaktorene har vi altså laget av det originale kumulative datasettet, på samme måte som i (2.1). På bakgrunn av at vi skal gjøre denne baklengs rekursive delingen, må utviklingsfaktorene være laget på kumulative data, og ikke inkrementelle data, England [2].

Videre finner vi de inkrementelle tilpassede dataene fra de kumulative tilpassede dataene vi nå har laget oss. Dette medfører at hjørnepunktene i punktene (1,10) og (10,1) alltid vil være like for inkrementelle tilpassede og observerte data, og tilsvarende vil residualene her alltid være lik null.

Steg 2:

Når vi nå har de observerte inkrementelle data, og tilpassede inkrementelle data, kan vi finne residualer og videre uteliggere. Vi finner Pearson residualer på følgende måte:

$$r_{i,j} = \frac{X_{i,j} - m_{i,j}}{\sqrt{\phi \cdot m_{i,j}}}$$

Der $X_{i,j}$ er de observerte inkrementelle data, $m_{i,j}$ er de inkrementelle tilpassede verdiene, og ϕ er en scale paramter (dispersjonsparameter). I denne oppgaven setter vi denne lik en, ettersom Pearson residualene kun brukes for å teste for uteliggere, og dermed vil ikke denne dispersjonen spille noen rolle for resultatet vi kommer frem til. Tabell 5.4 under er oversikt over Pearson residualene vi har kommet frem til ved datasettet vi nå ser på.

		Utviklingsår									
		1	2	3	4	5	6	7	8	9	10
Skadeår	1	10,56	-127,07	23,74	-17,00	70,99	103,45	56,71	-0,72	43,41	0,00
	2	119,29	128,12	-19,67	-204,04	-124,61	6,58	-89,30	-115,00	-68,70	
	3	-2 725,03	-882,00	-2,88	317,76	1 170,38	2 849,98	4 455,98	4 863,83		
	4	332,89	-241,68	-80,26	-36,85	-142,71	-154,63	-125,52			
	5	-73,35	181,31	-75,25	-97,24	136,82	-31,73				
	6	-134,98	-127,55	460,07	68,41	-149,48					
	7	-85,85	-56,71	208,53	33,36						
	8	364,06	-357,83	-247,59							
	9	-147,85	223,80								
	10	0,00									

Tabell 5.4

Når vi videre skal undersøke hvilken av residualene som er uteliggere, brukes klassisk boxplot-rejection. Vi betrakter det som mest sannsynlig at en verdi er uteligger hvis tilsvarende residual ligger utenfor boxplot intervallet:

$$[Q_1 - 3IQR, Q_3 + 3IQR]$$

$$IQR = \text{Interquartile range} = Q_3 - Q_1$$

Boxplot-rejection forutsetter at data er normalfordelte, noe som viser seg å stemme noenlunde bra med Pearson residualene jeg har funnet. Dette kan eventuelt testes f.eks ved en Shapiro-Wilk test. For alle datasettene jeg har sett på, har ikke dette vært et problem, og jeg har dermed brukt klassisk boxplot hele veien. Alternativet vil være å bruke en form for justert boxplot, når residualene er mer skjevfordelte, men dette har jeg valgt å utelate foreløpig.

Skadeår	Utviklingsår									
	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	1	1	0	0	1	1	1	1		
4	0	0	0	0	0	0	0			
5	0	0	0	0	0	0				
6	0	0	0	0	0					
7	0	0	0	0						
8	0	0	0							
9	0	0								
10	0									

Tabell 5.5

Tabell 5.5 viser en oversikt over hvilken av residualene som ligger utenfor boxplottet, og blir betraktet som uteliggere i dette eksempelet. Vi ser at det er skadeår 3 som gir problemer med dette datasettet.

I første omgang er det kun uteliggere i første kolonne vi skal se på, der uteliggere i resten av datasettet blir tatt opp i steg 4.

Steg 3:

Når vi finner en uteliggende residual i første kolonne, vil tilsvarende krav antas å være uteligger, og dette kravet vil da måtte endres på en av følgende to måter:

- Hvis residualen i neste kolonne i samme rad også er uteligger, skal kravet i første kolonne erstattes med medianen av alle krav i første kolonne. (avhengig av data, kan man eventuelt erstatte kravet med gjennomsnittet av kravet over og under, men dette har jeg ikke gjort i denne oppgave).

Altså, hvis $r_{k,2}$ uteligger:

$$X_{k,1} = \text{median}(X_{i,1} | i = 1, \dots, n)$$

- I den andre situasjonen, skal vi dividere kravet i neste kolonne i samme rad på en robust utviklingsfaktor, og erstatte det uteliggende kravet i første kolonne med denne verdien.

Hvis $r_{k,2}$ ikke uteligger:

$$X_{k,1} = \frac{X_{k,2}}{\text{median}\left(\frac{X_{i,2}}{X_{i,1}} \mid i = 1, \dots, n-1\right)}$$

Vi har nå gjort den første kolonnen fri for uteliggere, og vi kan videre ta for oss resten av datasettet.

Steg 4:

Måten vi sjekker resten av data for uteliggere er igjen å basere oss på residualer fra tilpassede data, men tilpassede data er nå laget på en ny måte.

Siden vi nå vet at første kolonne er fri for uteliggere, lager vi nye inkrementelle utviklingsfaktorer som er basert på første kolonne:

$$(5.2) \quad \hat{f}_j^1 = \text{median}\left(\frac{X_{i,j}}{X_{i,1}} \mid i = 1, \dots, n-j+1\right) \quad 2 \leq j \leq n$$

Og videre finner vi nye tilpassede verdier for våre data:

$$\hat{X}_{i,j} = X_{i,1} \cdot \hat{f}_j^1 \quad 2 \leq i \leq n, \quad 2 \leq j \leq n-i+1$$

På tilsvarende måte som tidligere finner vi nye Pearson residualer:

$$r_{i,j} = \frac{X_{i,j} - \hat{X}_{i,j}}{\sqrt{\hat{X}_{i,j}}}$$

Alle residualene i første kolonne vil nå være lik null. Som tidligere finner vi uteliggere i resten av data ved å se på boxplot intervallet $[Q_1 - 3IQR, Q_3 + 3IQR]$.

Residualer utenfor intervallet vil som tidligere betraktes som uteliggere, se Tabell 5.6.

Skadeår	Utviklingsår									
	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	1	1	1	1	1	1	1		
4	0	0	0	0	0	0	0			
5	0	0	0	0	0	0				
6	0	0	0	0	0					
7	0	0	0	0						
8	0	0	0							
9	0	0								
10	0									

Tabell 5.6

Alle residualer som betraktes som uteliggere erstatter vi med medianen av alle residualene, og når dette er gjort, tilbaketransformerer vi residualene til robuste krav:

$$X_{i,j} = r_{i,j} \cdot \sqrt{\hat{X}_{i,j} + \hat{X}_{i,j}}$$

Skadeår	Utviklingsår									
	1	2	3	4	5	6	7	8	9	10
1	701 848	232 585	194 470	148 488	98 600	61 875	47 145	32 260	25 628	18 173
2	1 864 592	856 348	441 065	256 385	139 112	103 032	62 855	47 355	33 132	
3	1 153 610	503 468	300 139	244 066	126 495	63 745	58 190	53 025		
4	966 722	168 570	149 128	140 050	38 410	9 548	12 308			
5	789 602	485 170	192 082	149 400	140 052	43 518				
6	1 154 888	475 018	619 605	330 220	91 025					
7	1 053 622	459 830	419 665	273 385						
8	1 956 875	368 372	244 525							
9	1 568 152	966 498								
10	1 322 485									

Tabell 5.7

Tabell 5.7 angir det robuste datasettet (inkrementelle data) vi har kommet frem til, etter å ha modifisert kravene som blir betraktet som uteliggere.

Steg 5:

Siste steg blir å kjøre vanlig Chain-Ladder metoden på det nye datasettet vårt.

En kan ellers merke seg at metoden så langt ikke vil klare å oppdage uteliggere i hjørnepunktene $X_{1,n}, X_{n,1}$ og de to kravene i nest siste kolonne, hhv $X_{1,n-1}$ og $X_{2,n-1}$. Verdonck foreslår å løse dette på følgende måte:

For $X_{n,1}$, så tar vi medianen til alle kravene i første kolonne, og tester for om $X_{n,1}$ avviker stort fra denne verdien. Dette gjør vi ved å teste om $X_{n,1}$ ligger i boxplottet for kravene i den første kolonnen. Blir $X_{n,1}$ sett på som uteligger, erstatter vi verdien med medianen i første kolonne.

For kravene $X_{1,n-1}$, $X_{2,n-1}$ og $X_{1,n}$ blir ting litt verre. For de to kravene i nest siste kolonne, blir medianen og gjennomsnittet det samme, og det er vanskelig å si hvilken av verdiene som eventuelt er uteliggere. Verdonck foreslår å løse dette ved å ekstrapolere hhv nest siste og siste utviklingsfaktor, basert på de tidligere utviklingsfaktorene. En ekstrapolerer da utviklingsfaktorer basert på (2.1), som i vanlig CL, for det nye robuste datasettet vi nå har. Verdonck foreslår to ulike modeller for ekstrapolasjonen, der andre modeller også kan brukes, men dette er ofte avhengig av datasettet.

Inverse modell:

$$(5.3) \quad \tilde{f}_j = b_0 + b_1 \frac{1}{j}$$

Eksponentiell modell:

$$(5.4) \quad \tilde{f}_j = b_0 + b_1 e^{-j}$$

For datasettene mine har jeg testet med begge modellene, men for datasettet jeg nå ser på (Tabell 5.1) spiller dette ingen rolle for resultatet. For datasett fra Tabell 7.1 vil det bli en minimal forskjell om man bruker 5.3 fremfor 5.4, mens for datasett fra Tabell 7.6 vil forskjellen bli litt større. I denne oppgaven har jeg for enkelthets skyld kun basert meg på den eksponentielle modellen når det gjelder resultatene jeg oppgir. En nærmere undersøkelse av hvilken modell som passer utviklingsfaktorene best vil være nødvendig når man jobber med reelle datasett, og det er mistanke om uteliggere øverst til høyre i trekanten.

For kravene $X_{1,n-1}$ og $X_{2,n-1}$, sammenligner vi de kumulative utviklingsfaktorene (5.5), med den estimerte utviklingsfaktorene vi får fra ekstrapolasjonen, (5.4) i denne oppgaven.

$$(5.5) \quad \hat{f}_j = \frac{C_{i,j}}{C_{i,j-1}}$$

Det er litt uklart hvordan vi skal sammenlikne to verdier i dette tilfellet, og på bakgrunn av denne sammenlikningen si noe om vi har uteliggere eller ikke. Dette er et av problemene jeg har hatt med artikkelen til Verdonck.

Jeg foreslår rett og slett å bestemme seg for et signifikansnivå α , f. eks $\alpha = 0.05$. Hvis avviket mellom de kumulative utviklingsfaktorene og utviklingsfaktoren vi har ekstrapolert er større enn $100 \cdot \alpha\%$, så antar vi at de kumulative utviklingsfaktorene er uteliggere. Hvis de kumulative utviklingsfaktorene for både rad 1 og 2 er uteliggere, erstatter vi begge to med den ekstrapolerte utviklingsfaktoren. Hvis kun en av dem er uteligger, erstatter vi den med den andre kumulative utviklingsfaktoren.

For $X_{1,n}$ gjør vi det så enkelt at vi ekstrapolerer de tidligere utviklingsfaktorene som over, og hvis den kumulative utviklingsfaktoren laget ved (5.5) avviker mer enn $100 \cdot \alpha\%$ fra den ekstrapolerte, så bruker vi den ekstrapolerte utviklingsfaktoren i stedet.

Her må vi også passe på at kravet $X_{1,n}$ må justeres hvis utviklingsfaktoren for $X_{1,n-1}$ er uteligger, uten at utviklingsfaktoren for $X_{1,n}$ er uteligger.

Som tatt opp i kapittel 4.4, er hjørnepunktene og nærliggende verdier i trekanten svært sensitiv for uteliggere, og hvis det viser seg at metoden fanger opp uteliggere i hjørnepunktene, eller de nærliggende kravene $X_{1,n-1}$ $X_{2,n-1}$, er det spesielt viktig at man graver dypere i datasettet. Uteliggere gir som tidligere vist store utslag på reserven, slik at det kan være naturlig å sette α enda mindre, for eksempel $\alpha = 0.025$. Dette gjør at det skal enda mindre til for at punktene $X_{1,n-1}$, $X_{1,n}$ og $X_{2,n-1}$ betraktes som uteliggere, noe som gir økt sannsynlig for at vi fanger dette opp, og vil gå nærmere inn i datasettet for å gjøre en grundigere vurdering av kravene og hvordan vi skal håndtere de.

Når det gjelder den robuste metoden jeg har tatt for meg steg for steg nå, har jeg programmert hele metoden i R. Koden finnes i kapittel 11.

6. Prediksjonsfeil

Når man lager estimater, er det også ønskelig å vite noe om hvor gode estimatene er. I vårt tilfelle ønsker vi å kunne si noe om usikkerheten på estimatet av reserven som vi kommer frem til. Det finnes flere mål på usikkerhet, men i skadereservering er det vanlig å betrakte prediksjonsfeilen (prediction error) av den totale reserven. Vi ønsker som nevnt å legge opp et best mulig estimat på reserven en må ha tilgjengelig for å dekke IBNR krav. Prediksjonsfeilen sier noe om hvor mye estimatet vårt avviker fra den "sanne" verdien. Mao kan dette oversettes til: desto lavere prediksjonsfeil, desto bedre estimat. I forsikringsbransjen er det ikke fullt så enkelt, der det kan være andre faktorer som spiller inn også. Blant annet vil selskapets risikoappetitt spille en sentral rolle her. Hvor stor risiko skal selskapet egentlig ta? På den ene siden står eierne som ønsker størst mulig avkastning på den investering de har gjort, og på den andre siden har du kundene som ønsker et solvent selskap som kan tilbakebetale de krav som rapporteres. Det er altså flere ting å ta hensyn til, men jeg velger likevel å ta for meg prediksjonsfeilen i denne oppgaven.

Prediksjonsfeilen for en variable X , der \hat{X} er estimatet av X , er generelt gitt som

$$MSEP = E[(X - \hat{X})^2]$$
$$RMSEP = \sqrt{E[(X - \hat{X})^2]}$$

Prediksjonsfeilen som vi snakker om er på fagspråket kjent som *Root Mean Square Error of Prediction*, RMSEP, og er altså kvadratroten av *Mean Square Error of Prediction*, MSEP. Ofte benevnes dette som RMSE og MSE også, men det er det samme vi mener i dette tilfellet.

Hvis vi videre ser på uttrykket for MSEP, kan vi gjøre følgende

$$\begin{aligned} E[(X - \hat{X})^2] &= E[((X - E[X]) - (\hat{X} - E[X]))^2] \\ &= E[(X - E[X])^2] - 2E[(X - E[X])(\hat{X} - E[X])] + E[(\hat{X} - E[X])^2] \\ &= E[(X - E[X])^2] + E[(\hat{X} - E[X])^2] \\ &= \text{Var}[X] + E[(\hat{X} - E[X])^2] \end{aligned}$$

Dette gir mening, siden kryssleddet kan vises at er lik null ved å anta $E[X] \approx E[\hat{X}]$:

$$\begin{aligned}
E[(X - E[X])(\hat{X} - E[X])] &\approx E[E[(X - E[X])(\hat{X} - E[\hat{X}]) | \mathcal{S}]] \\
&= E[(\hat{X} - E[\hat{X}])E[(X - E[X]) | \mathcal{S}]] \\
&= E[(\hat{X} - E[\hat{X}]) \cdot 0] \\
&= 0
\end{aligned}$$

Her er \mathcal{S} "historien" som vi kjenner til, som i dette tilfellet vil si det vi ikke skal predikere. Å anta $E[X] \approx E[\hat{X}]$, forventningen til X tilsvarer forventningen til estimatet av X , bør være en ok antakelse, ettersom målet i utgangspunktet å finne et best mulig estimat på X .

Det siste leddet i uttrykket for MSEP, $E[(\hat{X} - E[X])^2]$ kan vi se nærmere på. Vi skriver det som

$$\begin{aligned}
E[(\hat{X} - E[X])^2] &= E[(\hat{X} - E[\hat{X}] - (E[X] - E[\hat{X}]))^2] \\
&= E[(\hat{X} - E[\hat{X}])^2] + E[(E[X] - E[\hat{X}])^2] \\
&= \text{Var}[\hat{X}] + (E[X] - E[\hat{X}])^2 \\
&\approx \text{Var}[\hat{X}]
\end{aligned}$$

Kryssleddet vil her også forsvinne, siden $E[\hat{X} - E[\hat{X}]] = 0$. Leddet $(E[X] - E[\hat{X}])^2$ er differansen mellom forventningen til estimatoren og den sanne verdien, kvadrert. Denne differansen blir ofte kalt *bias*, der en forventningsrett estimator har en *bias* = 0. Vi antar at *biasen* er svært liten, og vi setter dermed dette leddet lik 0 her, noe som gjør at vi ender opp med

$$MSEP = \text{Var}[X] + \text{Var}[\hat{X}]$$

eller

$$RMSEP = \sqrt{\text{Var}[X] + \text{Var}[\hat{X}]}$$

Uttrykket for MSEP kan da sees på som summen av to ledd, nemlig variansen i dataene (prosessvariansen), og variansen som skyldes estimering (estimeringsvariansen).

Når man videre skal estimere prediksjonsfeilen av reserven med klassiske statistiske metoder, er problemet redusert til å finne prosessvariansen og estimeringsvariansen. Ved en Poisson / overdispersert Poisson tilpasning, som enkelt kan modelleres som en GLM, er dette ganske rett frem, [3].

Det er ellers verdt å nevne at mange har lett for å blande prediksjonsfeil med standardfeil. Standardfeil er kvadratroten til estimeringsvariansen, mens prediksjonsfeilen er som vi har vist, kvadratroten av summen av prosessvariansen og estimeringsvariansen. Prediksjonsfeilen sier altså noe om variasjonen til en prediksjon, hvor det tas hensyn til usikkerhet i parameterestimering og variasjon i data. Denne forskjellen er viktig å huske på, slik at en ikke blander de to tingene sammen.

Når jeg videre skal se på prediksjonsfeilen til estimerer jeg kommer frem til, har jeg beregnet prediksjonsfeilen med Mack sin formel, analytisk som GLM og ved bruk av bootstrap.

For den vanlige Chain-Ladder metoden har Thomas Mack laget en formel for beregning av prediksjonsfeilen for den totale reserven, [4]. M. Sagosen har blant annet studert denne formelen og dens bakgrunn i sin masteroppgave, [5]. Jeg har ikke tenkt å gjenta hva som har blitt gjort og vist tidligere, men vil i denne oppgaven kun bruke Mack sin formel for å beregne prediksjonsfeil når jeg sammenligner resultater fra Chain-ladder metoden, med en robust Chain-Ladder metode.

Det har også blitt vist at Chain-Ladder metoden kan underbygges av flere stokastiske modeller, blant annet en overdispersert Poisson modell. Ved å tilpasse trekanten vår med historiske krav i en GLM modell, får vi samme reserver som med Chain-Ladder metoden, og kan da enkelt beregne prediksjonsfeilen analytisk. De Silva [3] har blant annet laget et program for dette i R, som vi kommer til å bruke i denne oppgaven.

6.1 Bootstrap

Bootstrap går i all hovedsak ut på å bruke dataene man har, lage et nytt like stort datasett ved å trekke tilfeldig tall med tilbakelegging fra det originale datasettet, og så lage nye estimerer på dette datasettet. Dette gjør man N antall ganger, og man kan tilslutt finne gjennomsnitt, standard error, konfindensintervall osv. Fordelen med bootstrap er at det er en veldig enkel måte å lage estimerer og finne usikkerhet på, spesielt når fordelingen til dataene er ukjent. Bootstrap metoden har tidligere fått en del kritikk, men i dagens samfunn hvor vi har sterke og raske datamaskiner, lar det seg enkelt gjøre å la N være veldig stor, slik at vi får gode og stabile estimerer på usikkerheten.

I denne oppgaven ønsker vi å bruke bootstrap metoden på trekantene våre, for å finne usikkerheten til estimatene på den totale reserven som vi kommer frem til. Spesielt for den robuste Chain-Ladder metoden ønsker vi å gjøre dette, ettersom det er vanskelig å argumentere for at bruk analytiske metoder for å beregne prediksjonsfeilen her gir mening.

I bransjen som vi nå ser på, er det vanlig å bootstrappe residualene, i stedet for selve kravene. Hadde vi resamplet selve kravene i stedet, hadde vi flere ganger endt opp med en trekant med store svingninger i hvordan kravene fordeler seg på utviklingsårene. Vi unngår altså dette i stor grad ved å resample residualer.

Det finnes flere måter å regne residualer på, slik at det er viktig at vi bruker den best egnede metoden, og er konsistent i valget vårt. I vanlige regresjonsmodeller der feilledet er antatt å være Normalfordelt, er residualene observert verdi minus tilpasset verdi, (observed less fitted). Ettersom en Poisson GLM underbygger CL metoden, velger å vi å bruke Pearson residualer som passer til denne modellen:

$$r_{i,j} = \frac{X_{i,j} - m_{i,j}}{\sqrt{m_{i,j}}}$$

Som tidligere i oppgaven, tar vi for oss "unscaled" pearson residualer her, og lar heller dispersjonsparameteren komme inn når vi regner ut selve prediksjonsfeilen.

Vi har som nevnt tidligere at $RMSEP = \sqrt{E[(X - \hat{X})^2]} = \sqrt{Var[X] + Var[\hat{X}]}$, altså kvadratroten av prosessvariansen og estimeringsvariansen.

Siden en overdispenserte Poisson GLM gir samme reserver som CL metoden, regner vi ut reserven for hver av de N nye bootstrap-samples'ene med CL metoden. Vi finner bootstrap standardfeilen ved å ta standardavviket til våre N bootstrap reserver. Dette vil da være et estimat på kvadratroten av estimeringsvariansen. Problemet er at denne ikke kan direkte sammenlignes med den analytiske estimeringsvariansen vi ville fått fra en GLM, siden bootstrap standardfeilen ikke tar hensyn til antall parametere brukt til å tilpasse modellen. Bootstrap metoden bruker bare residualene uten å ta hensyn til hvordan de er beregnet. Dette må vi nå ta hensyn til, for å kunne bruke dette med at den overdispenserte Poisson GLM og CL-metoden gir samme reserver. Vi løser dette med å gjøre en liten justering på bootstrap estimeringsvariansen, slik at de to metodene kan sammenlignes. Justeringen blir å multiplisere bootstrap estimeringsvariansen med $\frac{n}{n-p}$. Der n er antall residualer (n=55 for 10x10 trekant), og p er antall estimerte parametere i GLM modellen (p=19).

Dette gir oss at bootstrap estimeringsvariansen blir:

$$\text{Var}_{BS}[\hat{R}] = \frac{n}{n-p} (\text{SE}_{BS}(R))^2$$

Videre må vi ha et mål for bootstrap prosessvariansen, for å beregne hele bootstrap prediksjonsfeilen. I GLM modellen vil prosessvariansen være dispersjonsparameteren multiplisert med estimatet på reserven. Det samme blir tilfellet når vi beregner bootstrap prosessvariansen.

Tar vi alt dette i betraktning, får vi da følgende uttrykk for prediksjonsfeilen beregnet med bootstrap metoden, fra England [2]:

$$PE_{BS} = \sqrt{\varphi_p \cdot R + \frac{n}{n-p} (\text{SE}_{BS}(R))^2}$$

Der et estimat på skala parameteren φ er gitt ved

$$\varphi_p = \frac{\sum r^2}{n-p}$$

Her er r residualene som vi bruker når vi gjør bootstrap. For datasettet brukt i [2], se Tabell 7.9, blir for eksempel dispersjonsparameteren 52 601, som vi også finner igjen når vi kjører vanlig Poisson GLM i R.

Jeg har laget en egen funksjon i R for bruk av bootstrap metoden, der R-koden ligger tilgjengelig i kapittel 11.

7. Resultater

I denne oppgaven har jeg sett på de samme datasettene som Verdonck har brukt i sin artikkel. RunOffTriangle1 og 2 har jeg kalt to av datasettene, der sistnevnte ble brukt som eksempel i kapittel 5. Begge disse to trekantene er reelle data fra et skadeforsikringsselskap i Belgia, Verdonck [1]. RunOffTriangle3 er også fra et skadeforsikringsselskap i Belgia, hentet fra Verdonck [6]. Verdonck har ikke tatt med noe forklaring på hva datasettene egentlig er, om det er antall skader, rapporterte skader, utbetalt, eller hva. Etter å ha studert datasettene litt, har jeg konkludert med meg selv at det kan være snakk om rapporterte skader, der RunOffTriangle1 trolig er for hele forsikringsbransjen, mens RunOffTriangle2 og 3 kan være for bransjenivå (f.eks reiseforsikring eller lignende). Dette spiller uansett ikke noe rolle når det gjelder fremgangsmåte og resultatene mine, men det kan være greit å vite hva man egentlig studerer.

Jeg vil nå ta for meg ett og ett datasett, se hvorvidt vi finner uteliggere i datasettene, hvordan reserven kan legges opp ved bruk av Chain-Ladder metoden, samt den Robuste Chain-Ladder metoden, og ellers prediksjonsfeilen på estimatene.

Først tar jeg for meg RunOffTriangle1 fra [1], Tabell 7.1.

I dette datasettet var det ingen av kravene som ble sett på som uteliggere i min metode. Verdonck kom dog frem til at kravet i utviklingsår 9 fra skadeår 2 var uteligger. Hvordan han kom frem til dette er fremdeles ikke klart for meg, men han har nok ikke sammenlignet utviklingsfaktorene mot et signifikansnivå slik som jeg har gjort. Dette er ellers kravet som jeg nevnte som en mulig uteligger da vi så på influens funksjoner i kapittel 4.3. I tabell 7.2 og 7.3 ser vi resultatene av hhv. den klassiske CL metoden, og den robuste CL metoden jeg har sett på. Som vi ser er resultatene helt like for dette datasettet, da ingen uteliggere ble oppdaget.

Skadeår	Utviklingsår									
	1	2	3	4	5	6	7	8	9	10
1	135338126	90806681	68666715	55736215	46967279	35463367	30477244	24838121	18238489	14695083
2	125222434	89639978	70697962	58649114	46314227	41369299	34394512	26554172	24602209	
3	136001521	91672958	78246269	62305193	49115673	36631598	30210729	29882359		
4	135277744	103604885	78303084	61812683	48720135	39271861	32029697			
5	143540778	109316613	79092473	65603900	51226270	44408236				
6	132095863	88862933	69269383	57109637	48818781					
7	127299710	92979311	61379607	50317305						
8	120660241	89469673	71570718							
9	134132283	87016365								
10	131918566									

Tabell 7.1

Skadeår	CL reserve	Bootstrap reserve	Bootstrap pred. %	GLM pred. %	Mack pred %
1	15 011 643	15 047 094	16	16	19
2	38 011 251	38 062 381	10	10	17
3	67 704 116	67 747 166	7	7	10
4	106 779 775	106 890 704	6	6	8
5	131 407 908	131 353 498	5	5	6
6	168 979 637	168 955 909	5	5	5
7	226 403 952	225 996 197	4	4	4
8	304 821 202	305 229 507	4	4	4
9	404 269 458	404 309 545	5	5	5
10	1 463 388 942	1 463 592 000	3	3	3

Tabell 7.2

Skadeår	CL reserve	Bootstrap reserve	Bootstrap pred. %	GLM pred. %	Mack pred %
1	15 011 643	15 047 094	16	16	19
2	38 011 251	38 062 381	10	10	17
3	67 704 116	67 747 166	7	7	10
4	106 779 775	106 890 704	6	6	8
5	131 407 908	131 353 498	5	5	6
6	168 979 637	168 955 909	5	5	5
7	226 403 952	225 996 197	4	4	4
8	304 821 202	305 229 507	4	4	4
9	404 269 458	404 309 545	5	5	5
10	1 463 388 942	1 463 592 000	3	3	3

Tabell 7.3

Skadeår	Utviklingsår									
	1	2	3	4	5	6	7	8	9	10
1	701848	232585	194470	148488	98600	61875	47145	32260	25628	18173
2	1864592	856348	441065	256385	139112	103032	62855	47355	33132	
3	1152332	2381638	2545868	2613448	2310415	2712015	3662850	3704750		
4	966722	168570	149128	140050	38410	9548	12308			
5	789602	485170	192082	149400	140052	43518				
6	1154888	475018	619605	330220	91025					
7	1053622	459830	419665	273385						
8	1956875	368372	244525							
9	1568152	966498								
10	1322485									

Tabell 7.4

Skadeår	CL reserve	Bootstrap reserve	Bootstrap pred. %	GLM pred. %	Mack pred %
1	44 804	57 413	1 354	473	2 271
2	485 365	530 225	848	246	1 026
3	288 670	287 360	207	116	226
4	751 042	739 666	115	79	131
5	1 693 424	1 667 286	75	58	92
6	1 933 478	1 846 718	76	57	83
7	3 341 620	3 354 465	68	49	67
8	5 155 799	5 123 221	60	46	59
9	4 979 105	5 000 697	69	56	70
10	18 673 307	18 607 052	77	31	76

Tabell 7.5

Skadeår	CL reserve	Bootstrap reserve	Bootstrap pred. %	GLM pred. %	Mack pred %
1	44 804	46 138	155	156	71
2	57 616	59 001	106	107	49
3	60 488	61 141	88	88	38
4	111 050	111 790	66	65	30
5	241 007	242 100	47	47	22
6	320 896	319 771	39	39	23
7	718 049	721 898	28	27	16
8	1 337 748	1 338 889	22	22	22
9	1 511 924	1 508 826	22	23	27
10	4 403 582	4 409 553	17	17	13

Tabell 7.6

Tabell 7.4 viser datasettet RunOffTriangle2 [1], og krav som blir påvist å være uteliggere med den robuste CL metoden er uthevet. Tabell 7.5 og 7.6 gir reserve og prediksjonsfeil for hhv den klassiske og robuste CL metoden.

Det er altså en betydelig forskjell i reservene her. I skadeår 3 for dette datasettet har det antakeligvis vært en spesiell sjelden hendelse / katastrofe som har ført til de økte skaderapporteringene i utviklingsårene deretter. I et slikt tilfelle der reserven fra den robuste metoden avviker såpass mye fra klassisk CL (over 4 ganger så liten), er det nødvendig å ta en nærmere titt på datasettet, og gjøre en videre analyse om hvordan en skal legge opp reserven. I et slikt tilfelle bør en absolutt ta i bruk andre metoder og undersøke hvordan estimatet på reserven blir da.

Skadeår	Utviklingsår									
	1	2	3	4	5	6	7	8	9	10
1	2709472	2086828	1373262	902270	633120	422288	278132	86070	66002	86710
2	1791958	1682388	1879652	1179850	1080628	878815	642640	650442	447882	
3	1519338	694782	523490	451785	252450	77355	47990	18150		
4	2183012	1014815	778818	701680	506295	189452	251968			
5	3864165	2484822	1567012	1047242	366640	229945				
6	2444540	1335810	906415	863665	409018					
7	2439000	1592875	983065	688808						
8	3525685	2085190	1249468							
9	3578580	1875010								
10	3046332									

Tabell 7.6

Skadeår	CL reserve	Bootstrap reserve	Bootstrap pred. %	GLM pred. %	Mack pred %
1	103 701	103 349	144	140	461
2	138 153	137 233	95	93	208
3	425 677	422 375	57	56	113
4	1 215 321	1 200 355	36	36	62
5	1 120 528	1 110 018	36	35	58
6	1 684 062	1 673 500	30	30	44
7	3 471 686	3 447 816	23	23	28
8	5 033 489	5 020 318	21	21	25
9	6 428 518	6 373 781	23	23	22
10	19 621 134	19 488 746	14	14	21

Tabell 7.7

Skadeår	CL reserve	Bootstrap reserve	Bootstrap pred. %	GLM pred. %	Mack pred %
1	89 199	90 089	91	92	32
2	146 803	147 062	57	57	135
3	286 670	287 638	43	43	92
4	984 808	982 018	25	25	48
5	845 897	845 842	25	25	44
6	1 300 945	1 307 365	20	21	32
7	2 925 502	2 919 214	15	15	19
8	4 237 802	4 242 529	14	14	14
9	5 569 501	5 558 055	15	15	13
10	16 387 128	16 379 812	9	9	12

Tabell 7.8

Tabell 7.6 viser datasett som jeg har kalt RunOffTriangle3. Dette datasettet er også fra et skadeforsikringsselskap i Belgia som jeg har hentet fra [2]. Vi ser at det er 4 krav fra skadeår 2 som betraktes som uteliggere, og videre gir utslag i reserven. Avviket i reserven er på ca 16.5 % ved den robuste metoden, og det er igjen absolutt nødvendig å ta en nærmere titt på datasettet, spesielt kravene fra skadeår 2.

Merk ellers hvordan kravet i punkt (1,10) ikke betraktes som uteligger, men er like stort som kravet to år tidligere. Hvis vi setter $\alpha = 0.025$ i stedet, vil dette kravet også betraktes som

uteligger, og vi får en utestående reserve med den robuste CL metoden på 18 460 305 som er ca 2M mer enn fra tabell 7.8 over.

Skadeår	Utviklingsår									
	1	2	3	4	5	6	7	8	9	10
1	357 848	766 940	610 542	482 940	527 326	574 398	146 342	139 950	227 229	67 948
2	352 118	884 021	933 894	1 183 289	445 745	320 996	527 804	266 172	425 046	
3	290 507	1 001 799	926 219	1 016 654	750 816	146 923	495 992	280 405		
4	310 608	1 108 250	776 189	1 562 400	272 482	352 053	206 286			
5	443 160	693 190	991 983	769 488	504 851	470 639				
6	396 132	937 085	847 498	805 037	705 960					
7	440 832	847 631	1 131 398	1 063 269						
8	359 480	1 061 648	1 443 370							
9	376 686	986 608								
10	344 014									

Tabell 7.9

Tabell 7.9 er et datasett hentet fra [2], som er et mye brukt datasett i studiene rundt skadereservering. Verdonck bruker dette datasettet i sin artikkel [1] også, der han ønsker å se på hvor godt den robuste CL-metoden virker når man multipliserer ett og ett krav med 10, sammenliknet med den klassiske CL-metoden. Jeg har gjort tilsvarende som Verdonck, men jeg har sammenliknet den robuste og klassiske metoden ved å multiplisere ett og ett krav opp med både 2, 5 og 10. Dette for å kunne sammenlikne metoden i ulike scenarioer, og også få et innblikk av hvor mye vi egentlig må multiplisere opp kravene med, for at de skal betraktes som uteliggere. Jeg har også tatt med kolonner for prediksjonsfeil, beregnet med ulike metoder. Prediksjonsfeil med bootstrap metoden gir antakeligvis et mer riktig bilde av hva prediksjonsfeilen er, ettersom det er vanskelig å argumentere for hvordan Mack sin metode og analytisk prediksjonsfeil (GLM) kan brukes etter å ha modifisert datasettet ved den robuste CL-metoden. Jeg har likevel tatt med beregninger for sammenlikning.

I tabell 7.10 er ett og ett krav multiplisert med 10, og vi ser at alle kravene fra trekanten blir betraktet som uteliggere. For svært mange av kravene er det også stor forskjell mellom den klassisk CL-metoden og den robuste når det gjelder den totale reserven. Den øverste raden er resultatene uten uteliggere, og er et slags utgangspunkt vi ønsker å sammenligne de resterende dataene med. Vi ser at den robuste metoden gir total reserve som er svært likt utgangspunktet (uten uteliggere), og det er tydelig at denne metoden presterer svært bra for dette datasettet. Dette fremgår også tydelig av prediksjonsfeilen, som er lavere og mer stabil for den robuste metoden. Kolonnen helt til høyre angir hvorvidt kravet som multipliseres med 10 betraktes som uteligger av den robuste metoden, noe vi ser alle kravene ble med en tidobling av kravet.

I Tabell 7.11 er kravene femdoblet, noe som gir at faktisk ikke alle kravene betraktes som uteliggere, nemlig kravene i punktene (1,8), (1,10) og (3,6). Dette er de tre minste kravene i

trekanten vår, og dermed forståelig. Fremdeles presterer den robuste metoden svært bra sammenliknet med den klassiske.

Ved å bare doble kravene, som vi har gjort i tabell 7.12, er det enda færre krav som blir betraktet som uteliggere. Igjen ser vi at for kravene som betraktes som uteliggere, gir den robuste metoden er estimat på reserven som er mye nærmere utgangspunktet, enn hva den klassiske metoden gir.

Outlier	CL reserve	Mack err.	GLM Err.	Bootstrap reserve	Boots. Err	Rob. CL reserve	Mack err.	GLM err.	Rob. bootstrap res.	Boots err.	Outlier
-	18 680 856	13 %	16 %	18 852 681	16 %	18 680 856	13 %	16 %	18 852 681	16 %	-
X1,1	12 603 783	32 %	35 %	12 802 758	35 %	18 487 959	12 %	14 %	18 586 588	14 %	1
X1,2	12 851 784	32 %	48 %	13 764 729	55 %	18 411 731	13 %	16 %	18 584 221	16 %	1
X1,3	15 813 130	26 %	38 %	16 508 563	42 %	18 370 569	13 %	16 %	18 520 702	16 %	1
X1,4	18 169 959	22 %	29 %	18 732 450	31 %	18 419 406	14 %	15 %	18 565 626	16 %	1
X1,5	20 132 751	20 %	36 %	20 816 133	38 %	18 681 093	13 %	16 %	18 852 378	16 %	1
X1,6	22 709 179	18 %	38 %	23 655 351	42 %	18 584 734	12 %	15 %	18 750 177	15 %	1
X1,7	20 329 847	20 %	20 %	20 682 297	21 %	18 879 228	12 %	14 %	19 026 769	14 %	1
X1,8	21 616 873	19 %	20 %	21 944 365	20 %	19 149 029	14 %	16 %	19 341 377	16 %	1
X1,9	28 068 004	14 %	19 %	28 450 341	19 %	20 910 974	12 %	16 %	21 019 733	17 %	1
X1,10	26 382 875	15 %	16 %	26 658 211	16 %	20 410 445	12 %	16 %	20 608 879	16 %	1
X2,1	13 064 239	31 %	32 %	13 366 286	33 %	18 619 218	13 %	16 %	18 789 881	16 %	1
X2,2	13 080 595	31 %	49 %	14 392 639	52 %	18 628 484	13 %	16 %	18 801 838	16 %	1
X2,3	16 044 692	25 %	45 %	17 454 608	48 %	18 713 353	13 %	16 %	18 886 731	16 %	1
X2,4	20 313 528	20 %	45 %	22 575 257	47 %	18 757 158	13 %	16 %	18 917 970	16 %	1
X2,5	20 318 298	20 %	30 %	20 889 073	30 %	18 713 874	13 %	16 %	18 877 377	16 %	1
X2,6	21 142 008	19 %	26 %	21 611 883	27 %	18 740 127	13 %	16 %	18 907 650	16 %	1
X2,7	25 833 401	16 %	30 %	26 785 860	30 %	18 526 323	13 %	16 %	18 699 127	16 %	1
X2,8	24 770 914	16 %	22 %	25 191 921	22 %	18 865 708	14 %	16 %	19 035 040	16 %	1
X2,9	36 975 225	11 %	20 %	37 789 414	20 %	18 128 896	13 %	16 %	18 213 788	17 %	1
X3,1	14 594 660	28 %	28 %	14 798 483	28 %	16 911 913	12 %	14 %	16 962 266	14 %	1
X3,2	14 881 519	27 %	51 %	16 728 790	60 %	18 942 881	13 %	16 %	19 104 568	16 %	1
X3,3	18 190 072	22 %	44 %	19 486 134	45 %	18 744 757	13 %	16 %	18 926 295	16 %	1
X3,4	22 260 738	18 %	40 %	23 916 152	39 %	18 608 427	14 %	16 %	18 780 538	16 %	1
X3,5	23 910 232	17 %	39 %	25 375 415	42 %	18 437 900	13 %	16 %	18 593 996	16 %	1
X3,6	20 281 482	20 %	17 %	20 489 160	18 %	18 899 668	13 %	15 %	19 061 497	15 %	1
X3,7	26 993 251	15 %	29 %	27 848 878	31 %	18 345 131	13 %	16 %	18 520 101	16 %	1
X3,8	26 124 080	16 %	22 %	26 636 178	23 %	18 501 515	13 %	16 %	18 671 282	16 %	1
X4,1	14 890 583	27 %	29 %	15 222 507	30 %	18 344 006	13 %	16 %	18 511 466	16 %	1
X4,2	16 119 238	25 %	52 %	17 532 044	52 %	18 927 113	13 %	16 %	19 122 238	16 %	1
X4,3	19 166 515	21 %	37 %	20 590 635	38 %	18 677 034	13 %	16 %	18 850 269	16 %	1
X4,4	26 779 838	15 %	50 %	31 039 739	56 %	18 270 950	12 %	14 %	18 373 567	14 %	1
X4,5	20 950 173	19 %	22 %	21 125 322	21 %	18 892 896	13 %	15 %	19 037 736	15 %	1
X4,6	23 178 356	17 %	26 %	23 767 610	27 %	18 649 556	13 %	16 %	18 821 117	16 %	1
X4,7	22 505 925	18 %	20 %	22 851 656	20 %	18 950 015	12 %	14 %	19 055 740	14 %	1
X5,1	14 864 015	27 %	36 %	15 379 975	37 %	19 021 397	13 %	16 %	19 196 489	16 %	1
X5,2	17 682 939	23 %	37 %	18 387 506	37 %	17 927 809	13 %	16 %	18 030 094	17 %	1
X5,3	21 424 577	19 %	43 %	23 566 282	47 %	18 706 467	13 %	16 %	18 887 641	16 %	1
X5,4	23 930 482	17 %	33 %	25 327 877	33 %	19 199 746	13 %	16 %	19 402 566	16 %	1
X5,5	24 047 427	17 %	31 %	24 736 006	32 %	18 934 684	13 %	16 %	19 098 632	16 %	1
X5,6	25 775 382	16 %	31 %	26 672 307	33 %	18 733 871	13 %	16 %	18 906 576	16 %	1
X6,1	15 906 579	25 %	32 %	16 102 002	32 %	18 362 703	12 %	14 %	18 458 269	14 %	1
X6,2	19 712 830	21 %	41 %	21 294 633	44 %	18 663 571	13 %	16 %	18 839 404	16 %	1
X6,3	22 724 309	18 %	36 %	23 897 090	36 %	18 761 396	13 %	16 %	18 925 244	16 %	1
X6,4	26 011 071	16 %	31 %	26 949 080	33 %	19 204 133	13 %	16 %	19 345 645	16 %	1
X6,5	27 889 017	15 %	35 %	29 722 466	37 %	18 444 105	13 %	16 %	18 590 997	16 %	1
X7,1	17 251 335	23 %	31 %	17 559 213	32 %	18 786 451	13 %	16 %	18 961 636	16 %	1
X7,2	22 268 233	18 %	33 %	23 610 687	35 %	18 761 392	13 %	16 %	18 949 763	16 %	1
X7,3	28 111 881	14 %	37 %	30 394 373	38 %	18 608 619	13 %	16 %	18 776 718	16 %	1
X7,4	32 254 649	13 %	31 %	33 861 580	33 %	19 273 972	13 %	16 %	19 431 810	16 %	1
X8,1	20 451 046	20 %	23 %	20 633 775	24 %	18 679 791	13 %	16 %	18 850 520	16 %	1
X8,2	32 812 737	12 %	27 %	34 318 629	29 %	18 463 426	13 %	16 %	18 638 383	16 %	1
X8,3	44 155 256	9 %	31 %	47 832 454	32 %	17 631 819	13 %	16 %	17 771 404	16 %	1
X9,1	27 531 995	15 %	21 %	28 105 108	22 %	18 643 601	13 %	16 %	18 813 350	16 %	1
X9,2	50 350 360	8 %	19 %	51 628 207	19 %	18 336 128	13 %	16 %	18 508 711	16 %	1
X10,1	60 313 152	7 %	16 %	61 120 574	16 %	19 004 501	13 %	16 %	19 181 691	16 %	1

Tabell 7.10

Outlier	CL reserve	Mack err.	GLM Err.	Bootstrap reserve	Boots. Err	Rob. CL reserve	Mack err.	GLM err.	Rob. bootstrap res.	Boots err.	Outlier
-	18 680 856	13 %	16 %	18 852 681	16 %	18 680 856	13 %	16 %	18 852 681	16 %	-
X1,1	15 257 168	16 %	25 %	15 417 044	25 %	18 487 959	12 %	14 %	18 586 588	14 %	1
X1,2	15 152 043	16 %	31 %	15 555 006	33 %	18 411 731	13 %	16 %	18 584 221	16 %	1
X1,3	16 910 645	14 %	25 %	17 175 357	26 %	18 370 569	13 %	16 %	18 520 702	16 %	1
X1,4	18 181 943	13 %	19 %	18 406 525	20 %	18 419 406	14 %	15 %	18 565 626	16 %	1
X1,5	19 068 319	13 %	26 %	19 363 968	27 %	18 681 093	13 %	16 %	18 852 378	16 %	1
X1,6	20 228 179	12 %	30 %	20 621 336	31 %	18 584 734	12 %	15 %	18 750 177	15 %	1
X1,7	19 388 049	13 %	16 %	19 636 867	17 %	19 089 985	13 %	14 %	19 258 759	14 %	1
X1,8	19 967 636	12 %	17 %	20 201 169	17 %	19 967 636	25 %	17 %	20 201 169	17 %	0
X1,9	22 833 006	11 %	17 %	23 076 157	17 %	20 910 974	12 %	16 %	21 019 733	17 %	1
X1,10	22 103 975	11 %	16 %	22 325 469	16 %	22 103 975	12 %	16 %	22 325 469	16 %	0
X2,1	15 540 079	16 %	22 %	15 748 399	23 %	18 619 218	13 %	16 %	18 789 881	16 %	1
X2,2	15 227 592	16 %	30 %	15 809 930	32 %	18 628 484	13 %	16 %	18 801 838	16 %	1
X2,3	16 814 550	15 %	29 %	17 331 295	31 %	18 713 353	13 %	16 %	18 886 731	16 %	1
X2,4	18 763 002	13 %	32 %	19 613 641	32 %	18 757 158	13 %	16 %	18 917 970	16 %	1
X2,5	19 278 194	13 %	21 %	19 516 177	21 %	18 713 874	13 %	16 %	18 877 377	16 %	1
X2,6	19 714 304	12 %	19 %	19 940 899	20 %	18 740 127	13 %	16 %	18 907 650	16 %	1
X2,7	21 764 145	11 %	24 %	22 147 235	24 %	18 526 323	13 %	16 %	18 699 127	16 %	1
X2,8	21 368 876	11 %	19 %	21 593 537	19 %	18 865 708	14 %	16 %	19 035 040	16 %	1
X2,9	26 811 686	9 %	18 %	27 177 234	18 %	18 128 896	13 %	16 %	18 213 788	17 %	1
X3,1	16 431 739	15 %	20 %	16 563 439	20 %	16 911 913	12 %	14 %	16 962 266	14 %	1
X3,2	16 157 812	15 %	33 %	16 878 589	35 %	18 942 881	13 %	16 %	19 104 568	16 %	1
X3,3	18 032 754	14 %	29 %	18 504 686	29 %	18 744 757	13 %	16 %	18 926 295	16 %	1
X3,4	20 003 216	12 %	28 %	20 624 056	28 %	18 608 427	14 %	16 %	18 780 538	16 %	1
X3,5	20 896 512	12 %	29 %	21 476 332	30 %	18 437 900	13 %	16 %	18 593 996	16 %	1
X3,6	19 388 384	13 %	15 %	19 550 536	15 %	19 388 384	12 %	15 %	19 550 536	15 %	0
X3,7	22 361 317	11 %	23 %	22 712 926	24 %	18 345 131	13 %	16 %	18 520 101	16 %	1
X3,8	21 988 955	11 %	19 %	22 268 051	19 %	18 501 515	13 %	16 %	18 671 282	16 %	1
X4,1	16 545 029	15 %	21 %	16 762 854	21 %	18 344 006	13 %	16 %	18 511 466	16 %	1
X4,2	16 723 879	15 %	34 %	17 197 069	34 %	18 927 113	13 %	16 %	19 122 238	16 %	1
X4,3	18 638 097	13 %	25 %	19 204 691	25 %	18 677 034	13 %	16 %	18 850 269	16 %	1
X4,4	21 968 145	11 %	37 %	23 448 425	38 %	18 325 796	12 %	14 %	18 420 226	14 %	1
X4,5	19 679 332	12 %	16 %	19 789 639	16 %	18 892 896	13 %	15 %	19 037 736	15 %	1
X4,6	20 673 602	12 %	20 %	20 965 014	20 %	18 649 556	13 %	16 %	18 821 117	16 %	1
X4,7	20 380 886	12 %	16 %	20 593 108	17 %	18 950 015	12 %	14 %	19 055 740	14 %	1
X5,1	16 307 623	15 %	26 %	16 611 346	27 %	19 021 397	13 %	16 %	19 196 489	16 %	1
X5,2	17 871 671	14 %	25 %	18 105 397	25 %	17 927 809	13 %	16 %	18 030 094	17 %	1
X5,3	19 628 489	12 %	30 %	20 403 216	31 %	18 706 467	13 %	16 %	18 887 641	16 %	1
X5,4	20 951 372	12 %	23 %	21 465 181	23 %	19 199 746	13 %	16 %	19 402 566	16 %	1
X5,5	21 056 359	12 %	23 %	21 319 070	23 %	18 934 684	13 %	16 %	19 098 632	16 %	1
X5,6	21 833 978	11 %	24 %	22 213 900	25 %	18 733 871	13 %	16 %	18 906 576	16 %	1
X6,1	16 889 434	14 %	23 %	17 046 543	23 %	18 362 703	12 %	14 %	18 458 269	14 %	1
X6,2	18 671 137	13 %	29 %	19 280 054	29 %	18 663 571	13 %	16 %	18 839 404	16 %	1
X6,3	20 322 360	12 %	25 %	20 706 548	26 %	18 761 396	13 %	16 %	18 925 244	16 %	1
X6,4	21 909 682	11 %	22 %	22 188 341	23 %	19 204 133	13 %	16 %	19 345 645	16 %	1
X6,5	22 773 372	11 %	27 %	23 524 188	28 %	18 444 105	13 %	16 %	18 590 997	16 %	1
X7,1	17 453 076	14 %	23 %	17 645 768	24 %	18 814 552	13 %	16 %	18 990 812	16 %	1
X7,2	19 955 362	12 %	24 %	20 498 388	25 %	18 761 392	13 %	16 %	18 949 763	16 %	1
X7,3	22 734 420	11 %	28 %	23 515 303	28 %	18 684 274	13 %	16 %	18 856 245	16 %	1
X7,4	24 713 653	10 %	24 %	25 167 739	24 %	19 273 972	13 %	16 %	19 431 810	16 %	1
X8,1	19 076 718	13 %	18 %	19 183 040	19 %	18 314 712	12 %	14 %	18 399 312	14 %	1
X8,2	24 752 661	10 %	21 %	25 248 666	22 %	18 466 070	13 %	16 %	18 640 866	16 %	1
X8,3	30 002 811	8 %	24 %	31 204 181	25 %	17 631 819	13 %	16 %	17 771 404	16 %	1
X9,1	22 304 806	11 %	18 %	22 625 872	18 %	18 765 080	13 %	16 %	18 941 512	16 %	1
X9,2	32 756 191	7 %	17 %	33 200 891	17 %	36 049 793	11 %	12 %	36 273 262	13 %	1
X10,1	37 184 098	7 %	16 %	37 645 035	16 %	19 004 501	13 %	16 %	19 181 691	16 %	1

Tabell 7.11

Outlier	CL reserve	Mack err.	GLM Err.	Bootstrap reserve	Boots. Err	Rob. CL reserve	Mack err.	GLM err.	Rob. bootstrap res.	Boots err.	Outlier
-	18 680 856	13 %	16 %	18 852 681	16 %	18 680 856	13 %	16 %	18 852 681	16 %	-
X1,1	17 648 223	14 %	17 %	17 802 797	18 %	17 648 223	15 %	17 %	17 802 797	18 %	0
X1,2	17 556 761	14 %	18 %	17 751 655	19 %	16 826 940	16 %	18 %	16 985 164	20 %	1
X1,3	18 119 044	14 %	16 %	18 263 900	17 %	18 119 044	14 %	16 %	18 263 900	17 %	0
X1,4	18 493 337	13 %	15 %	18 639 173	15 %	18 493 337	13 %	15 %	18 639 173	15 %	0
X1,5	18 715 224	13 %	18 %	18 889 560	18 %	18 715 224	15 %	18 %	18 889 560	18 %	0
X1,6	19 005 723	13 %	20 %	19 213 426	21 %	18 584 734	12 %	15 %	18 750 177	15 %	1
X1,7	18 852 939	13 %	15 %	19 042 663	16 %	18 852 939	13 %	15 %	19 042 663	16 %	0
X1,8	18 999 202	13 %	16 %	19 184 541	16 %	18 634 048	12 %	14 %	18 742 511	14 %	1
X1,9	19 714 568	12 %	16 %	19 903 535	16 %	20 280 409	11 %	14 %	20 311 796	15 %	1
X1,10	19 536 636	13 %	16 %	19 722 279	16 %	19 536 636	13 %	16 %	19 722 279	16 %	0
X2,1	17 740 446	14 %	16 %	17 906 649	16 %	17 352 055	13 %	15 %	17 437 920	15 %	1
X2,2	17 575 533	14 %	18 %	17 811 577	18 %	17 321 622	15 %	18 %	17 434 931	19 %	1
X2,3	18 041 870	14 %	18 %	18 229 939	18 %	18 041 870	15 %	18 %	18 229 939	18 %	0
X2,4	18 525 327	13 %	19 %	18 799 784	19 %	18 757 158	13 %	16 %	18 917 970	16 %	1
X2,5	18 803 711	13 %	16 %	18 959 400	16 %	18 803 711	13 %	16 %	18 959 400	16 %	0
X2,6	18 927 725	13 %	16 %	19 091 139	16 %	18 927 725	13 %	16 %	19 091 139	16 %	0
X2,7	19 430 911	13 %	18 %	19 623 395	18 %	19 430 911	15 %	18 %	19 623 395	18 %	0
X2,8	19 349 381	13 %	16 %	19 518 957	16 %	19 349 381	17 %	16 %	19 518 957	16 %	0
X2,9	20 713 563	12 %	16 %	20 911 838	16 %	18 128 896	13 %	16 %	18 213 788	17 %	1
X3,1	18 019 090	14 %	16 %	18 160 251	16 %	16 547 400	12 %	14 %	16 581 603	14 %	1
X3,2	17 841 401	14 %	19 %	18 098 827	19 %	18 942 881	13 %	16 %	19 104 568	16 %	1
X3,3	18 420 988	13 %	18 %	18 601 165	18 %	18 744 757	13 %	16 %	18 926 295	16 %	1
X3,4	18 949 551	13 %	17 %	19 174 733	18 %	18 608 427	14 %	16 %	18 780 538	16 %	1
X3,5	19 211 816	13 %	19 %	19 449 755	19 %	18 437 900	13 %	16 %	18 593 996	16 %	1
X3,6	18 857 108	13 %	15 %	19 020 253	15 %	18 857 108	13 %	15 %	19 020 253	15 %	0
X3,7	19 598 254	12 %	17 %	19 787 303	18 %	18 345 131	13 %	16 %	18 520 101	16 %	1
X3,8	19 507 881	13 %	16 %	19 697 037	17 %	19 507 881	18 %	16 %	19 697 037	17 %	0
X4,1	18 040 364	14 %	16 %	18 208 952	16 %	18 040 364	13 %	16 %	18 208 952	16 %	0
X4,2	17 980 950	14 %	20 %	18 151 299	20 %	18 927 113	13 %	16 %	19 122 238	16 %	1
X4,3	18 615 495	13 %	16 %	18 843 174	16 %	18 615 495	13 %	16 %	18 843 174	16 %	0
X4,4	19 422 518	13 %	23 %	19 829 462	23 %	18 325 796	12 %	14 %	18 420 226	14 %	1
X4,5	18 928 757	13 %	15 %	19 069 998	15 %	18 573 570	12 %	13 %	18 642 881	13 %	1
X4,6	19 177 962	13 %	16 %	19 363 730	16 %	19 177 962	13 %	16 %	19 363 730	16 %	0
X4,7	19 105 863	13 %	15 %	19 278 334	15 %	18 767 294	12 %	14 %	18 868 288	14 %	1
X5,1	17 905 805	14 %	18 %	18 097 751	18 %	17 905 805	15 %	18 %	18 097 751	18 %	0
X5,2	18 398 194	13 %	16 %	18 533 734	17 %	17 841 069	13 %	16 %	17 958 259	17 %	1
X5,3	18 856 678	13 %	19 %	19 116 067	19 %	18 856 678	16 %	19 %	19 116 067	19 %	0
X5,4	19 235 912	13 %	16 %	19 442 428	16 %	19 235 912	13 %	16 %	19 442 428	16 %	0
X5,5	19 272 969	13 %	17 %	19 435 466	17 %	19 272 969	14 %	17 %	19 435 466	17 %	0
X5,6	19 469 136	13 %	17 %	19 667 557	18 %	19 469 136	14 %	17 %	19 667 557	18 %	0
X6,1	18 087 670	14 %	17 %	18 246 196	17 %	18 087 670	15 %	17 %	18 246 196	17 %	0
X6,2	18 564 589	13 %	18 %	18 799 451	18 %	18 002 301	15 %	18 %	18 145 313	20 %	1
X6,3	19 057 986	13 %	17 %	19 221 276	17 %	19 057 986	14 %	17 %	19 221 276	17 %	0
X6,4	19 482 248	13 %	16 %	19 618 642	16 %	19 482 248	13 %	16 %	19 618 642	16 %	0
X6,5	19 703 985	12 %	18 %	19 996 884	19 %	18 444 105	13 %	16 %	18 590 997	16 %	1
X7,1	18 211 711	13 %	17 %	18 378 930	17 %	17 853 074	14 %	15 %	17 938 522	15 %	1
X7,2	18 923 782	13 %	16 %	19 153 694	17 %	18 358 101	14 %	16 %	18 506 801	18 %	1
X7,3	19 661 966	12 %	18 %	19 914 283	19 %	19 661 966	16 %	18 %	19 914 283	19 %	0
X7,4	20 189 055	12 %	17 %	20 349 786	17 %	20 189 055	14 %	17 %	20 349 786	17 %	0
X8,1	18 678 323	13 %	15 %	18 812 937	16 %	18 315 401	12 %	14 %	18 378 924	14 %	1
X8,2	20 143 913	12 %	16 %	20 336 152	16 %	18 466 070	13 %	16 %	18 640 866	16 %	1
X8,3	21 511 345	11 %	18 %	21 838 849	18 %	17 635 809	13 %	16 %	17 775 406	16 %	1
X9,1	19 502 543	13 %	16 %	19 714 723	16 %	19 502 543	14 %	16 %	19 714 723	16 %	0
X9,2	22 199 689	11 %	16 %	22 381 982	16 %	21 591 039	14 %	16 %	21 675 356	17 %	1
X10,1	23 306 666	10 %	16 %	23 553 303	16 %	19 004 501	13 %	16 %	19 181 691	16 %	1

Tabell 7.12

8. Hvordan fungerer ting i praksis?

Trekantene jeg har tatt for meg så langt har vært svært enkle trekanter med null forekomst av inkrementelle negative krav, og liten eller ingen vekst i tilhørende portefølje til trekantene. Det er altså veldig enkle og *rene* trekanter jeg så langt har sett på, når jeg har undersøkt trekantene med influens funksjoner og den robuste Chain-Ladder metode på jakt etter uteliggere.

Etter å ha vært i kontakt med ansvarshavende aktuar i DnB NOR Skadeforsikring (DNS), Kari Halvorsen Holter, har jeg fått tilgang til to av hennes trekanter for å kunne se nærmere på hvordan ting fungerer i virkeligheten. Når det gjelder selve tall og tabeller jeg oppgir som tilhører DNS, så er alt anonymisert og dermed ikke selskapets virkelige tall.

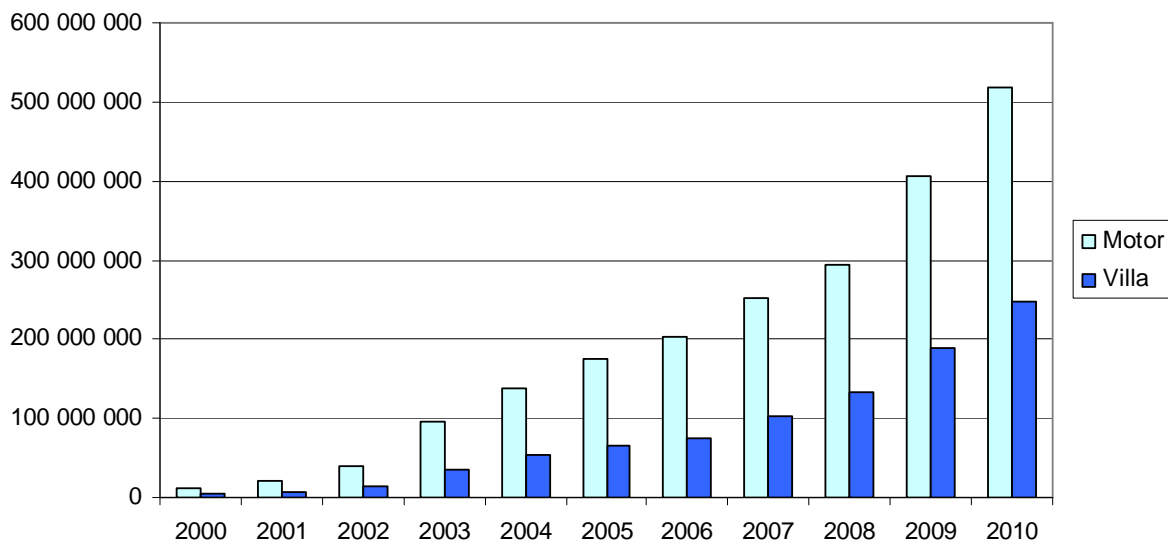
De to trekantene jeg har fått tilgang til er for bransjene motor og villa. Motor og villa står samlet for omkring 65 % av selskapets portefølje, og er altså de viktigste og største bransjene når det gjelder selskapets samlede avsetninger. Trekantene jeg har oppgitt under er rapporterte inkrementelle krav, og er da et skadeestimat på det totale skadeomfanget av rapporterte krav. Trekantene er ellers delt opp i kvartaler, og ikke år som jeg har sett på tidligere. Ofte kan det være lurt å se på kvartaler eller kanskje måneder i stedet for år, da dette gjør at selskapene kan få oversikt over hvordan reservene ligger an etter hver måned/ kvartal i året. For DNS er dette en nødvendighet, ettersom de må rapportere tall til ledelsen og avlegge regnskap hver måned. Dette er en mindre rapportering, både i tid og omfang, som kommer i tillegg til de kvartalsvise og årlige rapporteringene. På grunn av store trekanter, har jeg kun tatt med trekantene for de 6 første utviklingsårene (24 kvartalene) i tabell 9.1 og 9.2. Etter de 6 første utviklingsårene er det svært få rapporterte krav, slik at trekantene jeg har oppgitt dekker stort sett det som er av rapporterte krav.

Når det gjelder kravene i trekantene, ser vi spesielt at trekanten for villa (tabell 9.1) er en *stygg* trekant, og vi ser at den inneholder svært mange negative inkrementelle krav. Brann, og oftest storbrann, er den største forklaringen på dette, da krav knyttet til brann ofte tar lang tid før de er helt ferdigbehandlet. Etter at DNS overtok porteføljen i 2009 er for eksempel ingen av kravene fra storbrann, inntruffet etter overtakelse, lukket per i dag. Det er altså i stor grad brann som gjør at trekanten til villa er såpass stygg, mye pga ting tar tid, man oppdager nye ting som øker kostnadene, og at ting ofte koster mer enn hva man forventet.

Mye negative krav er også tilfellet for motor (tabell 9.2), men for motor er det mer vanlig at en har en del negative inkrementelle krav 2-5 utviklingskvartaler etter skaden skjedde. Dette skyldes i stor grad tilbakeføringer fra en 3.part, der det er andre selskaper eller involverte som

ender opp med å måtte dekke erstatningskostnadene for den skadelidte, og DNS får da tilbakeført utbetalingene de allerede har gjort. Vi ser av trekanten for *motor incurred* at det er et mønster av negative inkrementelle krav i 2-5 utviklingskvartal. I utviklingskvartalene etter dette (6+) er det også en del negative krav, men her er det ikke det samme mønsteret, og ting virker litt mer tilfeldig. Igjen skyldes antakeligvis de fleste negative kravene tilbakeføringer fra 3.part fra det man kan forvente, men som bare har tatt lengre tid. En annen ting som er viktig å nevne med motor, er at motor her består av personskader og blikkskader. Blikkskader er ofte skader som har kort hale og blir gjort opp i løpet av 5-6 måneder. Personskader er det verre med, da dette ofte kan være svært langhalede krav, og til tider krav som etter hvert viser seg å bli svært kostbart for selskapet.

Det andre punktet som gjør at trekantene for motor og villa skiller seg mye fra trekantene jeg tidligere har sett på, er dette med vekst i porteføljen. Grafen under viser oversikt over tegnet premie for motor og villa for porteføljen DNS i dag sitter på.



Figur 9.1

Vi merker oss den store veksten i figur 9.1, spesielt de siste to årene etter at DNS overtok porteføljen (2009 og 2010). Stor vekst i porteføljen gjør at metoden jeg har brukt for å se etter uteliggere tidligere blir mindre egnet enn hva vi skulle ønsket. For den robuste Chain-Ladder metoden, hvor residualer fra *fitted values* blir sammenliknet, vil det bli til at de største kravene i trekanten ofte blir sett på som uteliggere selv når de ikke er det. I tillegg gjør negative krav at Pearson residualer ikke kan beregnes rett frem, og en må dermed beregne residualene på en annen måte. Dette er altså en svakhet med metoden, og gjør den mindre brukervennlig for reelle trekanter enn de enkle trekantene jeg tidligere har sett på. En metode som klarer å fange opp vekst og i tillegg ta hensyn til dette med negative krav er altså nødvendig.

Influens funksjonene som jeg beregnet, var som tidligere nevnt i første omgang for å undersøke om Chain-Ladder metoden og Bornhuetter-Ferguson metoden er robuste, noe de ikke er. Jeg fant ellers ut at influens funksjonene beregnet på trekanter kan være med å gi et innblikk i hvilke krav som gir stor influens på den totale reserven, og da hvilke krav som kan være mulige uteliggere i trekanten. For trekantene med så stor vekst som villa og motor, vil igjen kravene fra de siste skadeårene få størst innflytelse på reserven, og influens funksjoner blir da også mer verdiløs når det gjelder å oppdage uteliggere i trekantene.

Å oppdage uteliggere i slike stygge trekanter som jeg har fått tilgang til her er altså vanskelig, og andre metoder / teknikker må til. Fremdeles er nok visuell inspeksjon av trekantene noe av det beste man kan gjøre. Visuelle måter en bør se på, er for eksempel hvordan de kumulative utviklingsfaktorene for hvert skadeår utvikler seg, sammenliknet med vektet snitt av utviklingsfaktorene, eventuelt mot median og gjennomsnitt. For eksempel vil vi med sammenlikning av utviklingsfaktorene for *motor incurred* trekanten, se at kravene fra 3. og 4. kvartal i 2009 (6 495 635 og 7 944 579) som ble rapportert et kvartal etter skaden oppstod, gir utviklingsfaktorer på hhv. 1,503 og 1,535, der vektet utviklingsfaktor etter Chain-Ladder metoden er på 1,224. Disse to kravene skiller seg altså ut, og forklaringen på dette skyldes at skadebehandlerne ikke satt noe reserve på de rapporterte kravene den gangen, altså de lot kravet bare ligge som et 0-krav. For motor overtok DNS porteføljen ca. midt i året, og disse to kravene er altså rett etter overtakelse. Ulik praksis fra risikobærerene og ulik praksis for behandling av regressen ¹⁾ er årsaken til hvorfor dette skjedde. Dette var dog noe Holter var klar over på den tiden, noe som gjorde at hun la dette til som et IBNeR krav i reserven, og dermed slapp DNS en underestimert av reserven for motor. Et annet krav som skiller seg ut i trekanten for motor, er kravet fra 3. kvartal 2007, rapportert 12 kvartaler etterpå (-1 194 268). Hva et så stort negativt krav 3 år etter skade skjedde skyldes, er ikke klart, men dette er et krav jeg absolutt ville undersøkt nærmere.

For villa gir tilsvarende sammenlikning også et par krav jeg stiller spørsmålstegn til. For eksempel gir kravet fra 2.kvartal 2009, rapportert etter 5 kvartaler (-1 655 359) en utviklingsfaktor på 0,793, mot Chain-Ladder faktorene på 1,02 i dette datasettet. Dette kravet viste seg å være utfall av en gunstig jury avgjørelse etter en tvist med et annet forsikringselskap, og dermed ble estimatet på skadeomfanget nullet ut med et negativt krav.

På de to neste sidene ligger trekantene for villa og motor. Jeg har markert alle de negative inkrementelle kravene for å gjøre det lettere å se hvor stor andel negative krav det faktisk er snakk om. Kravene som er uthevet er de kravene jeg har nevnt over som mulige uteliggere. I tillegg vil man på DNS sine *ekte* trekanter (oppgitte trekanter som sagt anonymisert) lettere se at det er flere krav som er mistenkelige, og som burde vært undersøkt nærmere.

Occurrence	Development quarter	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
19.991	4 777	-4 777	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19.992	38 009	-2 645	855	0	25 478	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19.993	33 901	27 938	-2 939	3 589	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19.994	54 675	14 996	4 777	-6 188	2 134	3 930	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.001	58 119	72 831	-4 522	14 189	20 725	-10 751	14 333	0	0	0	0	0	592	-7 307	0	14 953	0	0	0	0	0	0	0	0	0	0
20.002	35 430	57 438	7 688	5 631	-15 797	-6 369	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.003	143 250	355 597	39 914	-24 366	-3 085	-10 118	470	640	-15 924	-15 924	382	10 489	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.004	144 883	112 077	-18 696	14 751	-11 257	0	502	6 389	12 739	6 308	-4 289	3 603	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.005	67 534	-18 439	51 361	-18 439	-8 130	414	-5 189	7 962	0	0	0	0	0	0	-7 962	0	0	0	0	0	0	0	0	0	0	0
20.006	81 969	173 916	54 800	-25 805	-80 204	3 838	22 675	888	22 675	0	14 200	2 660	-23 322	0	836	4 777	84	0	0	0	0	-4 777	0	0	0	0
20.007	334 902	5 163	6 904	1 760	8 118	153 903	648	892	3 414	6 058	0	5 974	0	15 962	-12 793	0	0	0	0	0	0	0	0	0	0	0
20.008	217 128	36 487	79 583	77 315	-18 980	-12 744	12 739	-15 924	-15 924	27 421	0	6 279	-15 900	854	0	0	0	0	0	0	0	0	0	0	0	0
20.009	386 375	128 057	23 225	21 040	47 542	1 840	2 038	-7 544	1 433	9 339	0	15 924	-7 276	-3 474	0	0	0	0	0	0	0	0	0	0	0	0
20.010	343 609	58 205	39 883	3 169	-10 120	-34 373	9 682	7 803	0	-252	0	6 768	-27 893	0	609	0	0	0	0	0	0	0	0	0	0	0
20.011	479 430	105 078	272 338	-96 087	-8 083	17 581	23 460	1 349	14 681	-23 075	31 590	0	0	0	0	0	0	2 484	0	16 094	0	0	0	0	0	0
20.012	502 173	216 784	42 901	-5 365	15 047	29 882	8 030	-74 147	5 105	-14 641	15 444	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.013	442 972	168 028	103 004	123 542	27 269	-23 930	4 577	9 554	4 577	9 554	0	553	2 318	0	0	0	0	0	0	2 773	9 554	0	0	0	0	0
20.014	2 297 557	180 563	-191 888	-193 736	-12 715	5 273	9 964	7 612	0	0	-1 363	2 626	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.015	601 526	149 701	72 800	39 689	-5 587	31 582	9 296	-10 466	22 320	-3 943	17 837	3 984	-3 646	-2 217	0	0	0	-6 524	2 212	0	0	0	0	0	0	0
20.016	887 140	261 101	93 979	-68 657	-1 854	-19 004	864	8 039	3 426	-4 682	-1 779	5 038	49 815	28 599	7 962	-27 881	0	0	0	0	199 045	0	0	0	0	0
20.017	5 204 111	330 639	-15 195	27 494	-35 889	51 444	56 053	12 675	-60 305	1 726	13 656	-9 567	-1 927	-40 204	0	0	0	0	0	0	0	0	0	0	0	0
20.018	2 108 985	1 047 069	324 331	-41 366	47 887	228 257	168 834	-73 712	-218 811	-23 715	5 701	-1 616	-59 641	14 461	0	0	0	0	0	0	0	0	0	0	0	0
20.019	2 123 333	372 578	-148 723	-10 518	15 487	153 188	-68 650	-3 243	94 054	318	-1 703	6 369	15 935	31 847	-8 317	0	0	0	1 592	5 894	0	0	0	0	0	0
20.020	4 678 103	910 640	244 463	556 649	9 323	25 454	43 261	25 783	99 662	79 618	-20 470	15 464	-16 691	64 234	0	0	0	0	0	0	0	0	0	0	0	0
20.021	1 533 737	639 843	344 407	38 519	79 471	28 216	23 177	42	-4 141	-2 118	-367 980	0	0	3 513	1 482	25 400	11 839	0	0	0	0	0	0	0	0	0
20.022	2 268 930	1 351 063	303 161	34 909	13 536	-284 030	61 580	-19 141	-22 859	11 137	-36 024	-14 289	35 829	0	2 530	1 592	0	0	0	0	0	0	0	0	0	0
20.023	6 180 724	676 951	550 929	170 496	107 313	7 169	24 314	-184 925	-334 539	-889 766	0	0	0	3 425	12 739	3 185	0	-17 499	0	0	0	0	0	0	0	0
20.024	3 523 379	514 823	47 863	88 544	84 531	53 474	115 583	-134 113	281 591	-53 594	80 007	168 666	84 700	9 954	9 192	50 140	31 889	-78 502	5 894	0	-48 882	0	0	0	0	0
20.025	2 780 829	1 423 719	536 337	143 831	117 800	831 005	123 591	270 983	-479 492	13 533	53 018	20 349	-15 924	-266 386	-17 469	-28 527	-45 560	-89 000	0	0	0	0	0	0	0	0
20.026	1 953 284	1 628 325	255 915	338 470	12 148	217 452	41 887	6 048	3 012	-796	-92 088	13 055	6 519	8 149	-3 623	-578	0	0	0	0	0	0	0	0	0	0
20.027	1 721 743	415 063	120 057	214 260	239 728	36 646	302 964	-97 883	-1 702	13 294	16 246	-37 427	-40 804	15 820	2 613	25 478	0	0	0	0	0	0	0	0	0	0
20.028	1 902 411	555 611	335 969	411 128	92 860	-134 547	-40 304	17 396	-13 692	-29 560	-31 283	-34 502	34 389	0	0	0	0	0	0	0	0	0	0	0	0	0
20.029	2 670 862	592 885	623 725	236 982	-77 451	74 556	18 733	-45 872	-55 081	7 064	-9 858	-34 502	37 461	0	0	0	0	0	0	0	0	0	0	0	0	0
20.030	3 776 760	1 983 595	719 329	-64 382	74 633	753 961	324 994	-169 629	109 168	-106 051	4 528	-146 953	12 246	0	0	0	0	0	0	0	0	0	0	0	0	0
20.031	3 021 993	2 268 410	1 231 374	149 514	166 480	105 592	337 039	-242 317	362 485	13 838	19 331	8 332	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.032	3 380 686	1 068 442	1 001 412	331 857	628 355	1 093 508	-127 325	-230 429	-93 310	-21 729	87 716	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.033	3 803 481	2 601 577	1 002 368	896 893	1 056 444	-101 868	791 664	-78 368	167 989	-34 326	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.034	3 104 627	2 700 030	1 624 756	1 160 394	186 968	-347 307	42 070	-22 400	59 150	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.035	5 478 836	1 248 352	995 006	749 098	237 082	79 345	96 082	99 766	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.036	5 524 208	1 028 628	1 458 673	30 589	-1 655 359	100 478	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.037	9 055 404	443 523	1 977	615 206	-15 079	1 267 921	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.038	9 690 134	1 457 057	475 912	355 706	-176 070	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.039	2 016 355	2 389 635	901 565	540 606	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.040	16 109 822	1 817 310	943 497	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.041	18 552 293	305 050	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabell 9.1 Villa incurred

Source quarter	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
19.991	3.185																								
19.992		1645	-2.528																						
19.993		1021	3.068	35																					
19.994		5.705																							
19.995		87.262	-394	-12.991	-387																				
19.996		115.503	154.867	4.280	28.600	-61.083	-1.994																		
20.001		312.525	190.896	-32.258	-2.214	3.646	-27.166	2.106	2.822																
20.002		392.634	111.978	7.463	8.498	-58.516	-6.324	5.197	183.595	10.064	12.733	-40.351	0	5.350	5.012	-180.064	2.420	0	15.501	0	0	0	0	0	65.032
20.003		270.883	116.904	7.85	9.423	-1.821	0	271	0	0	0	1.592	0	-2.392	0	0	0	0	0	0	0	0	0	0	0
20.004		270.883	401.454	-66.464	-16.974	-1.569	6.960	12.463	7.844	3.423	3.775	1.106	0	0	0	0	-18.831	0	0	0	0	0	0	0	0
20.005		719.094	821.140	-174.445	-258.887	102.340	1.107	-15.884	-323	0	0	659	0	0	0	0	0	0	0	0	0	0	0	0	0
20.006		298.410	237.690	-32.647	-1.892	-17.760	223.777	318	0	21.990	-35.889	20.326	0	4.288	0	0	0	0	0	0	0	0	0	0	0
20.007		490.396	341.912	9.980	-24.338	3.163	-24.249	-1.592	0	-58.180	27	0	0	0	0	0	0	0	119.291	0	0	0	0	0	0
20.008		657.164	738.862	63.320	-42.849	-75.352	-170.165	7.27	964	494	478	0	0	23.314	0	0	0	0	0	0	0	0	0	0	0
20.009		880.619	343.431	159.854	-4.836	-10.271	-69.168	15.860	-4.203	14.071	1.274	2.484	0	4.221	-70.533	0	510	573	0	0	0	0	0	0	538
20.010		914.947	391.814	-115.475	-7.944	108.394	17.245	44.183	168	1.811	6.997	0	0	0	0	0	14.33	0	633	0	38.217	-1.266	33.599	0	0
20.011		111.726	578.979	33.458	23.533	-81.420	155	-7.283	-6.880	204	0	-1.784	0	0	0	0	0	2.070	0	0	0	0	0	0	0
20.012		1069.573	629.598	59.458	-91.000	-104.390	11.953	-2.079	21.300	-10.169	0	-4.178	1.323	0	0	0	538	0	0	0	0	0	0	0	0
20.013		1977.237	22.112	-105.875	9.106	19.508	11.355	6.917	14.331	14.33	0	-4.459	-63.233	0	0	0	0	0	0	0	1.534	0	0	0	0
20.014		1172.961	352.410	-4.926	-7.083	-594	6.639	-88.337	283	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.015		1962.315	778.951	-86.260	-49.981	-36.896	-36.875	-8.249	1.491	-24.570	0	12.569	0	0	0	0	0	0	0	0	0	0	0	0	0
20.016		2.231.676	1.827.331	40.477	-16.692	96.894	109.930	19.624	207	106.479	139.764	23.300	5.040	-1.592	0	0	0	0	0	0	0	0	0	0	0
20.017		3.209.955	952.535	-76.105	-60.821	20.228	-26.143	10.777	42.471	-55.180	-16.196	0	26.997	63.694	0	0	0	61.530	0	363.490	-290.979	-89.602	0	0	0
20.018		4.379.260	273.179	63.509	-85.408	-27.278	-8.790	13.878	80.969	285	64.191	101.295	-32.629	-6.277	0	0	0	2.418	0	222.930	12.741	0	0	0	0
20.019		5.820.745	2.513.175	-623.528	-327.764	-43.569	11.426	-136.280	-8.680	8.683	5.949	62.506	15.924	0	15.924	-42.444	0	637	9.554	0	0	0	0	0	0
20.020		6.604.439	1.999.895	503.769	317.441	52.670	-19.069	313	15.542	-35.108	36.015	61.779	15.924	398	0	0	0	0	0	7.962	13.547	0	0	0	0
20.021		5.653.063	623.945	-430.767	-336.321	33.405	57.234	9.572	8.176	36.722	19.468	-2.094	50.420	0	1.408	8.280	27	0	0	0	0	0	0	0	0
20.022		5.09.690	1.408.575	-367.146	-21.505	-3.064	-29.605	-10.988	-186.246	969	29.332	11.204	31.847	7.006	0	0	0	-3.060	0	0	0	0	0	0	0
20.023		6.673.926	2.036.106	-195.124	-57.536	-103.759	13.201	19.492	51.350	32.325	-11.785	-5.575	45.111	25.478	6.363	688	77.011	0	0	0	0	0	0	0	0
20.024		8.972.284	1.682.848	-625.009	-376.578	-226.107	-2.735	37.697	-12.974	10.493	28.191	-13.002	0	1.274	0	0	0	0	0	0	0	0	0	0	0
20.025		6.914.543	1.588.999	-456.006	-469.720	-17.874	-119.349	-43.290	3.631	-7.962	35.590	17.458	7.532	18.471	-48.485	2.861	1.648	0	0	0	0	0	0	0	0
20.026		5.831.988	3.16.702	-1.777.224	-409.678	-79.344	-94.869	16.405	-36.508	-92.220	10.489	9.809	27.737	0	25.208	2.170	187	0	0	0	0	0	0	0	0
20.027		7.138.722	1.614.991	-1.738.903	-246.032	-3.373	-21.308	30.767	61.151	71.911	5.036	15.754	17.867	-72.004	-40.471	215.365	5.795	-28.196	0	0	0	0	0	0	0
20.028		12.150.417	638.920	-1.168.704	-83.433	-35.929	-334.366	-145.654	-176.772	15.928	65.476	-13.293	-25.960	-27.462	4.777	-1.342	0	0	0	0	0	0	0	0	0
20.029		12.022.584	708.401	-1.384.526	-799.685	-763.018	-398.168	-56.164	5.941	10.491	-16.472	-63.796	-143.912	69.032	391.720	0	0	0	0	0	0	0	0	0	0
20.030		11.881.216	1.422.230	-1.95.887	-1.230.583	-454.790	30.812	-689	-22.799	245.573	-1.423	5.811	-1.84.289	131.598	0	0	0	0	0	0	0	0	0	0	0
20.031		16.721.102	766.194	-2.816.109	-1.078.645	-233.168	-14.462	-14.566	-10.273	33.064	-34.073	156.684	15.266	144.503	0	0	0	0	0	0	0	0	0	0	0
20.032		15.151.413	-352.666	-4.973.933	-677.975	-101.963	700.902	4.094	-76.693	-52.503	3.184	457.312	1.954	0	0	0	0	0	0	0	0	0	0	0	0
20.033		11.664.194	1.077.744	-799.734	-77.896	27.717	2.370	129.642	598.261	21.893	74.743	112.742	0	0	0	0	0	0	0	0	0	0	0	0	0
20.034		13.931.028	2.600.611	-462.616	-244.772	80.451	64.930	-29.743	44.087	20.754	-62.698	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.035		17.645.945	3.203.625	-1.615.484	-1.061.828	-208.754	-283.175	-32.657	522.954	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.036		14.945.792	7.944.579	6.435.639	449.226	-628.504	83.712	417.947	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.037		26.960.757	4.359.193	5.644	290.195	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.038		18.657.293	3.211.646	-147.706	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.039		19.378.927	3.992.103	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20.040		24.234.100	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabell 9.2 Motor incurred

Det at DNS overtok porteføljen i 2009 gjør at ulik oppgjørspraksis fra de ulike risikobærerne i tidligere år også bidrar til økt usikkerhet og problemer ved estimering av reserven. Som nevnt over, har det vært ulik praksis på hvordan blant annet regressen ¹⁾ har blitt håndtert tidligere, noe som har gitt sitt bidrag til at DNS ikke bruker de klassiske metodene jeg har sett på når IBNR reserven skal estimeres.

Når aktuar Holter skal estimere IBNR reserven for DNS, gjør hun dette med bakgrunn i forventet frekvens og gjennomsnittlig skadestørrelse, sett bort ifra de største storskadene. Dette gjøres på bakgrunn av dagens situasjon, slik at veksten i porteføljen blir hensyntatt, og at man i mindre grad drar med seg problemer fra ulik oppgjørspraksis tidligere år. Hvordan selskapene håndterer storskader varierer, men DNS velger å gjøre det på denne måten, der de tar bort de 8 % største skadene når IBNR reserven skal estimeres. Dette er med bakgrunn i at det forventes at de største skadene oppdages og rapporteres med en gang. DNS unngår dermed i stor grad problemet med uteliggere i trekanten. Jeg sier i stor grad, ettersom datasettene over allerede er bearbeidet, men likevel er det krav i trekanten som er mulige uteliggere. Når det gjelder IBNeR håndterer DNS dette slik at de korrigerer IBNR reserven ved å legge til estimat for IBNeR manuelt etterpå.

Det finnes andre selskaper som har annen praksis på håndtering av slike storskader, for eksempel er det noen som velger å ta ut de største skadene slik som DNS, men videre å simulere frem estimerer på IBNeR etter egne modeller som selskapene har laget. Dette er for så vidt en god praksis, men kanskje mest egnet for selskaper som har porteføljer med bransjer hvor det ofte er snakk om høyere forsikringssummer, hvor IBNeR utgjør en større andel av den totale IBNR reserven.

I tabell 9.3 og 9.4 har jeg tatt med reserven for villa og motor for DNS (aktuar), samt estimat på reserven etter mine beregninger for sammenlikning.

	Villa			
	Aktuar	CL	RoB CL	BF
IBNR	8 339 727	8 728 619	9 863 828	4 268 535
RBNS	44 959 321	44 959 321	44 959 321	44 959 321
SUM	53 299 048	53 687 940	54 823 149	49 227 856
<i>avvik fra aktuar</i>		388 893	1 524 102	-4 071 191

Tabell 9.3

For alle tre beregningsmetodene har jeg kun sett på skadeårene fra 2000 til 2010, og har altså valgt å se bort ifra 1999, ettersom 1999 er et lite og ubetydelig år av trekanten. Reserven som er oppgitt er kun reserven for skadeårene 2009 og 2010.

1) Regress – krav om å få dekket et beløp man har betalt for en annen

Vi ser at Chain-Ladder metoden er nærmest DNS sine avsetninger, og for villa gir metoden egentlig tilfredsstillende resultater.

	Motor			
	Aktuar	CL	RoB CL	BF
IBNR	12 301 704	9 076 167	9 097 931	8 685 173
RBNS	19 093 606	19 093 606	19 093 606	19 093 606
SUM	31 395 311	28 169 773	28 191 537	27 778 779
avvik fra aktuar		-3 225 537	-3 203 774	-3 616 532

Tabell 9.4

For motor er avviket enda større, og noe av forklaringen bak dette ligger nok i behandlingen av reserven for personskader. Siden trekanten for motor inneholder blikk- og personskader, setter DNS av en ekstra reserve til å dekke alle kravene for personskader. Dette har jeg altså ikke tatt med i mine estimater over, og forklarer en del av avviket.

For tallene som er oppgitt under den robuste metoden, har jeg kun brukt steg 4 fra den robuste metoden til å finne uteliggere. Jeg har altså antatt at kolonne $\hat{m}_{i,j}$ i trekantene er fri for uteliggere, og laget utviklingsfaktorer etter 5.2. Residualene er videre laget ved å dividere med kvadratroten av absolutt verdien til *fitted values* (9.1), for å unngå problemet med negative krav i Pearson residualer.

$$(9.1) \quad r_{i,j} = \frac{X_{i,j} - m_{i,j}}{\sqrt{|m_{i,j}|}}$$

Tabell 9.5 (villa) under viser hvilke krav som blir fanget opp som uteliggere av den robuste metode jeg har brukt her. Vi ser at kravet i posisjon (20091,5) som jeg påpekte etter visuell inspeksjon som mulig uteligger, også blir fanget opp av metoden. I tillegg ser vi at to krav rett over i trekanten også er mulige uteliggere, og kanskje burde undersøkes nærmere. Når det gjelder kravene fra kvartal 7, så viser dette igjen en svakhet med metoden. Her blir de fleste kravene sett på som uteliggere, selv om de ikke er det, noe som skyldes måten vi bruker medianen av utviklingsfaktorene fra hvert år til å lage *fitted values* av.

Occurrence r	Development quarter													
	0	1	2	3	4	5	6	7	8	9	10	11	12	13
19 991	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19 992	4 777	-4 777	0	356	0	0	25 478	0	0	0	0	0	0	0
19 993	39 809	-2 645	855	796	0	-5 096	0	0	-5 775	0	0	0	0	0
19 994	33 901	27 938	-2 939	3 589	0	0	-2 514	0	0	0	0	0	0	0
20 001	54 675	14 996	4 777	-14 351	-6 188	2 134	3 930	0	15 836	0	0	0	-7 307	0
20 002	58 119	72 831	-4 522	14 189	946	20 725	-10 751	1 433	0	0	0	592	0	0
20 003	35 430	57 438	7 688	5 831	-15 797	-6 369	0	0	0	-9 554	0	0	0	0
20 004	143 250	355 597	39 914	-24 366	-3 085	-10 119	470	640	-15 924	382	10 489	0	0	0
20 011	144 883	112 077	-18 695	14 751	-11 267	0	502	6 369	12 739	6 308	-4 299	3 603	0	0
20 012	67 534	-19 439	51 361	15 264	-8 130	414	-5 159	7 962	0	0	-2 339	0	0	-7 962
20 013	81 969	173 916	54 800	-25 805	-80 204	30 133	898	22 675	0	1 420	2 660	-23 322	0	0
20 014	334 902	5 163	6 904	1 760	8 118	153 903	648	892	3 414	6 058	0	5 974	0	1 596
20 021	217 128	36 847	78 583	77 315	-18 990	-1 274	12 739	-15 924	-15 924	27 421	0	6 279	-15 900	854
20 022	386 375	128 057	23 225	21 040	47 542	1 840	2 038	-7 544	1 433	9 339	0	15 924	-7 276	-3 474
20 023	343 609	56 205	39 863	3 169	-10 120	-34 373	9 682	7 803	0	-262	0	0	0	-3 185
20 024	479 430	105 078	272 338	-96 087	-8 083	17 581	23 460	1 349	14 681	-23 075	31 590	6 768	-27 893	0
20 031	502 173	216 784	42 901	-9 365	15 047	29 882	8 030	-74 147	5 105	-14 641	15 444	0	0	0
20 032	442 372	168 028	103 004	129 542	27 269	-22 330	4 577	9 554	0	0	-1 353	553	2 318	0
20 033	2 297 557	180 563	-191 888	-193 798	-12 715	5 273	9 964	7 612	0	9 546	0	2 626	0	15 924
20 034	601 526	149 701	72 800	39 689	-6 587	31 582	9 296	-10 466	22 320	-3 943	17 837	3 984	-3 646	-2 217
20 041	887 040	261 101	93 979	-68 657	-1 854	-13 004	864	8 039	3 426	-4 682	-1 779	5 038	49 815	28 599
20 042	557 180	125 616	32 093	25 913	23 547	-42 231	-8 853	-1 984	12 739	10 685	0	-237	0	0
20 043	2 204 111	330 639	-15 135	27 494	-35 899	51 444	56 053	12 675	-60 305	1 726	13 656	-9 567	-1 927	-40 204
20 044	2 108 985	1 047 069	324 331	-41 366	47 897	228 257	166 834	-73 712	-218 811	-29 715	5 701	-1 616	-59 641	14 461
20 051	2 123 333	372 578	-148 723	-10 618	15 487	153 188	-68 650	3 243	94 054	318	-1 703	6 369	15 935	31 847
20 052	4 678 103	910 640	244 463	556 649	9 323	25 454	49 261	25 783	99 662	79 618	-20 470	15 464	-16 691	64 234
20 053	1 533 737	639 843	344 407	38 519	79 471	26 216	23 177	42	-4 141	-2 118	-367 980	0	3 513	1 482
20 054	2 268 930	1 351 063	303 161	34 909	13 536	-284 030	61 580	-19 141	-22 859	11 137	-36 024	-14 289	35 829	0
20 061	6 180 724	678 851	550 929	170 496	107 313	7 159	24 314	-194 925	-334 539	-889 766	0	0	3 425	12 739
20 062	3 523 379	514 823	47 863	88 544	84 631	53 474	115 583	-134 113	281 591	-53 594	80 007	158 666	84 700	9 554
20 063	2 780 829	1 423 719	536 337	143 831	117 800	831 005	123 591	270 983	-479 492	13 533	53 018	20 349	-15 924	-266 386
20 064	1 953 284	1 628 325	255 815	338 470	12 148	217 452	41 887	6 048	3 012	-796	-92 088	13 055	6 519	8 149
20 071	1 721 743	415 063	120 057	214 260	239 728	36 646	302 964	-97 883	-1 702	13 294	16 246	-37 427	-40 804	15 820
20 072	1 902 411	655 611	335 969	411 128	92 860	-134 547	-40 304	17 396	-13 692	-29 560	-31 283	-34 502	34 389	0
20 073	2 670 862	592 885	629 725	238 992	-77 451	74 558	18 733	-45 872	-55 081	7 064	-9 658	-15 924	37 461	0
20 074	3 776 760	1 983 595	719 329	-64 382	74 633	753 961	324 994	-159 629	108 168	-106 051	4 528	-146 553	12 246	0
20 081	3 021 993	2 268 410	1 231 374	149 614	166 480	105 592	337 039	-242 317	362 485	13 838	19 331	8 332	0	0
20 082	3 380 686	1 068 142	1 001 412	331 857	628 355	1 093 508	-127 325	-230 429	-93 310	-21 729	87 716	0	0	0
20 083	3 803 481	2 601 577	1 102 368	896 893	1 056 444	-101 868	791 664	-78 368	157 989	-34 326	0	0	0	0
20 084	3 104 627	2 700 030	1 624 758	1 160 394	186 968	-347 307	42 070	-22 400	59 150	0	0	0	0	0
20 091	4 378 836	1 248 352	995 006	749 098	237 082	79 345	96 082	99 766	0	0	0	0	0	0
20 092	5 524 208	1 028 628	1 458 673	-31 556	30 589	-1 655 359	100 478	0	0	0	0	0	0	0
20 093	9 055 404	443 523	1 977	615 260	-15 079	1 267 821	0	0	0	0	0	0	0	0
20 094	9 690 134	1 457 057	475 812	356 706	-176 070	0	0	0	0	0	0	0	0	0
20 101	20 216 355	2 389 635	901 565	540 606	0	0	0	0	0	0	0	0	0	0
20 102	16 109 822	1 817 310	943 497	0	0	0	0	0	0	0	0	0	0	0
20 103	11 788 240	305 050	0	0	0	0	0	0	0	0	0	0	0	0
20 104	18 552 293	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabell 9.5

Tabell 9.6 under viser uteliggere etter den robuste metoden for motor. Vi ser at kravene jeg tidligere nevnte som mulige uteliggere etter visuell inspeksjon, (6 495 635 og 7 944 579), blir ikke fanget opp som uteliggere av metoden her. Residualene fra disse to kravene blir rett og slett ikke sett på som uteliggere av metoden, noe som kan sees på som en svakhet. Ellers merker vi oss hvor mange krav metoden mener er uteliggere for motor trekanten.

Occurrence c	Development quarter													
	0	1	2	3	4	5	6	7	8	9	10	11	12	13
19 991	3 185	0	1645	-2 525	0	0	0	0	0	0	0	0	0	0
19 992	693	5 705	1 021	3 008	35	159	0	0	0	0	0	0	0	0
19 993	136 768	87 262	32 523	-394	-12 991	-397	0	0	-2 516	0	0	271	0	0
19 994	115 503	154 867	25 819	-32 258	-68 657	4 280	28 600	-61 089	-1 994	0	0	0	0	0
20 001	312 525	190 696	-93 710	3 366	-2 214	3 646	-27 166	2 106	2 922	0	0	0	0	0
20 002	392 634	111 878	4 785	7 453	8 498	-58 516	-6 324	5 137	193 595	10 064	12 739	-40 351	0	0
20 003	270 883	116 904	16 184	-35 511	9 423	-1 821	0	0	0	271	0	1 592	0	-2 392
20 004	315 738	401 454	-66 543	-16 974	78 611	-1 559	6 950	12 463	7 844	3 423	3 775	1 106	0	0
20 011	713 094	821 410	-174 446	-256 667	102 340	-20 160	1 107	-15 884	-323	0	0	0	659	0
20 012	298 410	237 690	-32 647	-1 852	-17 760	229 777	318	0	21 990	5 803	-95 859	20 326	0	4 268
20 013	490 356	341 512	9 380	-24 338	-38 437	3 163	-24 249	-1 592	0	-55 180	27	0	0	0
20 014	657 164	738 862	63 320	-42 649	-75 352	-170 145	727	0	964	494	478	0	0	23 314
20 021	880 619	343 431	159 654	-4 836	10 271	-69 158	15 860	-4 203	14 071	1 274	2 484	0	0	4 221
20 022	914 947	391 814	-115 475	-7 944	108 394	17 245	441 183	158	1 811	6 997	0	1 433	0	0
20 023	1 111 726	578 979	33 458	23 533	-81 420	155	-7 283	-6 880	204	0	-1 784	0	0	0
20 024	1 069 573	629 586	59 680	-31 130	-104 390	11 953	-2 079	21 300	-10 169	0	-4 178	1 323	0	0
20 031	1 977 337	721 179	-22 112	-105 875	9 106	18 508	11 395	2 617	14 331	1 433	0	-4 459	-63 233	0
20 032	1 172 561	362 410	155 491	1 082	-4 926	-7 083	-554	6 639	-88 337	269	-8 787	0	0	0
20 033	1 962 315	778 551	27 048	-86 260	-49 981	-36 896	-35 875	-8 249	1 491	-24 570	0	12 569	0	0
20 034	2 231 676	1 827 331	40 477	-16 692	95 894	109 930	18 624	207	106 479	139 764	23 300	5 040	-1 592	0
20 041	3 907 583	952 635	-76 105	-60 821	20 228	-26 143	10 777	42 471	-55 180	134 438	-15 196	0	26 997	63 694
20 042	3 209 955	1 123 614	-156 538	75 499	114 819	2 796	-30 295	11 791	16 686	-9 151	6 352	-20	17 649	11 275
20 043	4 379 260	273 179	63 509	-85 408	-27 278	-8 790	13 878	80 989	285	64 191	101 295	-32 629	-6 277	0
20 044	5 820 745	2 513 176	-523 828	-327 764	-43 559	11 426	-136 280	-9 690	8 683	-6 821	5 949	62 506	15 924	0
20 051	6 604 439	1 199 895	-503 769	317 441	52 670	-13 059	313	15 543	-35 109	36 015	61 779	15 924	398	4 870
20 052	5 653 063	623 545	-490 767	-336 321	39 405	57 234	9 572	8 176	36 722	13 468	-2 034	50 420	0	-1 408
20 053	5 109 690	1 408 575	-367 146	-21 505	-3 064	-29 605	-10 988	-156 246	959	29 332	11 204	31 847	7 006	0
20 054	6 673 326	2 036 106	-155 124	-57 536	-103 759	13 201	19 492	51 350	32 325	-11 755	-5 575	45 111	25 478	6 369
20 061	8 972 284	1 682 848	-625 009	-375 678	-226 107	-2 735	37 697	-12 974	10 493	28 191	-13 002	0	1 274	0
20 062	6 914 543	1 588 999	-456 006	-469 720	-117 874	-118 349	-43 290	3 631	-7 962	35 590	17 458	7 532	18 471	-48 485
20 063	5 831 988	3 616 702	-1 777 224	-409 678	-79 344	-94 869	16 405	-36 508	-92 220	10 489	9 809	27 737	0	25 208
20 064	7 138 722	1 614 991	-738 503	-246 032	-2 373	-21 308	30 767	61 151	71 911	5 036	15 754	17 867	-72 004	-40 471
20 071	12 150 417	638 520	-1 168 704	-83 433	-35 323	-334 366	-145 654	-176 772	15 928	65 476	-13 293	-25 960	-222	-27 482
20 072	12 022 564	708 401	-1 384 526	-799 665	-763 018	-356 158	-56 134	5 941	10 491	-15 472	-63 756	-143 512	69 032	15 307
20 073	11 881 216	1 422 230	-195 587	-1 230 583	-454 790	30 812	23 541	-699	-22 799	245 573	-1 423	5 811	-1 194 268	131 598
20 074	12 088 011	3 219 934	-2 816 109	-1 079 645	-233 158	-14 462	-14 566	-10 273	33 064	-34 073	156 684	15 266	144 509	0
20 081	16 721 102	766 194	-2 497 933	-677 975	-101 953	700 902	4 094	-76 693	-52 503	3 184	457 312	1 954	0	0
20 082	15 151 413	-352 666	-1 371 915	-85 509	27 717	2 370	129 642	598 261	21 183	74 743	112 742	0	0	0
20 083	11 664 194	1 077 744	-799 734	-73 886	90 451	64 930	-28 743	44 087	20 754	-62 638	0	0	0	0
20 084	13 931 028	2 600 611	-482 616	-244 772	-93 402	-34 739	-94 895	-1 370	-52 287	0	0	0	0	0
20 091	17 645 945	3 203 625	-1 515 484	-1 051 828	-208 754	-283 175	-92 527	522 954	0	0	0	0	0	0
20 092	14 937 675	3 959 695	-626 058	168 165	-206 845	24 688	15 018	0	0	0	0	0	0	0
20 093	12 912 771	6 495 635	449 226	-528 504	83 712	417 947	0	0	0	0	0	0	0	0
20 094	14 848 792	7 944 579	-1 274 106	-150 869	109 880	0	0	0	0	0	0	0	0	0
20 101	26 360 757	4 359 135	5 844	290 195	0	0	0	0	0	0	0	0	0	0
20 102	18 657 293	3 211 646	-147 706	0	0	0	0	0	0	0	0	0	0	0
20 103	19 378 927	3 992 003	0	0	0	0	0	0	0	0	0	0	0	0
20 104	24 234 100	0	0	0	0	0	0	0	0	0	0	0	0	0

Tabell 9.6

For den klassiske Chain-Ladder metoden har jeg også gjort bootstrap simuleringer for å komme frem til en bootstrap reserve med konfidensintervall. Jeg har gjort 1000 simuleringer, og har kommet frem til en bootstrap reserve for villa på 10 079 275, og et 95 % konfidensintervall på [-841 387, 20 604 447]. Som vi ser gir konfidensintervallet oss meget lite ettersom det er såpass stort, men det sier oss dog noe om hvor stor usikkerheten knyttet til en slik trekant er, hvor det tidligere har vært så stor forekomst av negative inkrementelle krav. Prediksjonsfeilen i denne simuleringen kom på 46 % av bootstrap reserven, noe som igjen betyr stor usikkerhet.

For motor får jeg en bootstrap reserve på 11 215 206 med 95 % konfidensintervall [-31 031 671, 68 995 017], som altså gir enda styggere tall enn for villa, noe som skyldes større andel av negative krav i trekanten. Prediksjonsfeilen blir ellers over 100 %.

Det kan ellers stilles spørsmålsteget ved bootstrap metoden brukt på disse trekantene, pga den store forekomsten av negative inkrementelle krav i trekantene. Så stor andel negative og positive inkrementelle krav øker usikkerheten mye, da det for noen bootstrap samples vil trekkes flere negative residualer enn for andre samples, noe som gir store variasjoner i

simuleringene. Dette gjør nok at bootstrap metoden gir en del større usikkerhet enn hva som egentlig er tilfellet, og er en svakhet med denne metoden på slike trekanter. I tillegg kan den store veksten i porteføljen gi økt usikkerhet pga resampling av residualer (store mot små og vica versa), men det er i størst grad de negative kravene som ødelegger for denne metoden. En modifisert bootstrap metode hvor residualer fra noen utviklingskvartaler kun resamples til de samme utviklingsårene vil i de fleste tilfeller gi et riktigere bilde av den opprinnelige usikkerheten. Spesielt for trekanter slik som motor, der det i noen utviklingskvartaler forventes negative krav slik som vi har sett over. Det kan for eksempel være aktuelt for motor trekanter som over å resample residualer fra 2-5 utviklingskvartal for seg, og resten for seg.

Etter å ha sett på to reelle trekanter fra DnB NOR Skadeforsikring, så har jeg kommet frem til at den robuste Chain-Ladder metoden [1], fungerer dårlig for trekanter som tilhører porteføljer med mye vekst og stor forekomst av negative inkrementelle krav. Flere krav blir sett på som uteliggere selv om de kanskje ikke er det, og metoden klarer ikke fange opp alle mulige uteliggere heller. Det er altså nødvendig med en ytterligere modifisering for at den robuste metoden skal gi tilfredsstillende resultater for slike stygge trekanter.

Inflasjonsjustering er ellers noe jeg ikke har sett på i denne oppgaven, men som er viktig å ta hensyn til. Dette vil ikke løse problemet med vekst i porteføljen, men dette er noe som må tas med når reserven estimeres for selskapene. Derimot kan en form for vekting av skadeårene mot et basisår være en mulig måte å gjøre det lettere å oppdage uteliggere ved stor vekst i porteføljen. Andre ting som kan være aktuelt å ta for seg, som jeg så vidt nevnte tidligere i oppgaven, er å se på *paid* (utbetalt) fremfor *incurred* trekanter. Dette gjør at man i stor grad slipper unna problemet med negative krav. Å basere seg på *paid* gjør at man må stille enda høyere krav til historiske data, ettersom *paid* utvikler seg langsommere enn *incurred*.

9. Oppsummering

I denne oppgaven har jeg sett på hvordan uteliggere kan påvirke IBNR reserven i skadereservering. Jeg har studert influens funksjoner for den klassiske Chain-Ladder metoden, og videre sett at uteliggere kan påvirke Chain-Ladder estimatet for den totale reserven i stor grad. Jeg har videre tatt for meg den klassiske Chain-Ladder metoden, som er mye brukt ved IBNR reservering, og sett på en modifisering av metoden, slik at metoden skal fange opp uteliggere, og gi oss et innblikk i hva IBNR reserven vil være uten disse uteliggerne i datasettet. Dette har jeg gjort for tre ulike datasett, hvor resultatene ligger i kapittel 7. Ved å bruke den robuste Chain-Ladder metoden som et hjelpeverktøy, er det naturlig å sammenligne resultatene fra den robuste metoden mot den klassiske Chain-Ladder metoden. For resultater der avviket er stort er det nødvendig å gå nærmere inn på datasettet man jobber med, og undersøke hva uteliggerne egentlig skyldes, og hvor sannsynlig det er at slike ekstreme observasjoner vil oppstå igjen.

Etter å også ha fått tilgang til to reelle og aktuelle datasett fra DnB NOR Skadeforsikring, har jeg kommet frem til at den robuste metoden ikke alltid fungerer like bra på reelle data, spesielt for trekanter som tilhører porteføljer med mye vekst og stor forekomst av negative inkrementelle krav. Her har jeg riktignok benyttet metoden på trekanter som er delt inn i kvartaler, men selv om jeg aggregerer opp til inndeling per år, får vi tilsvarende problem, men dog i mindre grad. Dette er uansett mindre aktuelt ettersom DNS, og de fleste selskaper, må estimere reserven for hvert kvartal / måned fremover, pga månedlige og kvartalsvise rapporteringer.

For slike stygge datasett som de fra DNS, er det viktig å gjøre en grundig visuell inspeksjon av trekantene. Dette er for så vidt også viktig å gjøre for andre trekanter også, men for datasett som er mindre stygge, mindre vekst og lite eller ingen negative inkrementelle krav, vil den robuste metoden være mer egnet når det gjelder å oppdage eventuelle uteliggere og krav som er sterkt innflytelsesrik.

Noe som kan være aktuelt å se på videre er hvordan den robuste metoden kan modifiseres ytterligere, slik at en kan få ut nyttige resultater selv med stor vekst og stor forekomst av negative krav i trekantene. Et naturlig startsted for ytterligere modifisering, kan være å se på metoden med å legge til en konstant i trekantene for å unngå problemet med negative krav, og artikkelen "*Claims reserving when there are negative values in the runoff triangle: Bayesian analysis using the three-parameter Log-normal distribution*" – E. de Alba, kan være en artikkel å ta utgangspunkt i. Andre ting som kan være aktuelt å se mer på, er punktene nevnt i siste avsnitt i kapittel 8.

10. Appendix

Appendix A – Loglinear Poisson GLM = Marginale summer metoden

Vi antar nå at hvert krav i celle (i,j) i trekanten vår følger en $Poisson(\alpha_i \beta_j)$.

[der $X_{ij} = \alpha_i \beta_j$.] Vi ønsker nå å vise at å estimere α_i og β_j med MLE, gir samme resultat som med marginale summer metoden.

Først må vi finne likelihoodfunksjonen.

Vi har

$$P(X_{ij} = x_{ij}) = \frac{(\alpha_i \beta_j)^{x_{ij}}}{x_{ij}!} e^{-\alpha_i \beta_j}$$

Som gir oss likelihoodfunksjonen

$$L = \prod_{i,j} \frac{(\alpha_i \beta_j)^{x_{ij}}}{x_{ij}!} e^{-\alpha_i \beta_j}$$

Og log-likelihood

$$\begin{aligned} l &= \log L = \log \prod_{i,j} \frac{(\alpha_i \beta_j)^{x_{ij}}}{x_{ij}!} e^{-\alpha_i \beta_j} \\ &= \sum_{i,j} \log \left(\frac{(\alpha_i \beta_j)^{x_{ij}}}{x_{ij}!} e^{-\alpha_i \beta_j} \right) \\ &= \sum_i \sum_j (x_{ij} \log \alpha_i + x_{ij} \log \beta_j - \log(x_{ij}!) - \alpha_i \beta_j) \end{aligned}$$

Vi finner videre MLE ved

$$\frac{d}{d\alpha_i} l = 0$$

$$\frac{d}{d\beta_j} l = 0$$

Som gir oss

$$\frac{d}{d\alpha_i} l = \frac{\sum_j x_{ij}}{\alpha_i} - \sum_j \beta_j = 0$$

$$\frac{d}{d\beta_j} l = \frac{\sum_i x_{ij}}{\beta_j} - \sum_i \alpha_i = 0$$

og da

$$\sum_{i(j)} x_{ij} = \sum_{i(j)} \alpha_i \beta_j$$

Dette kjenner vi igjen som metoden marginale summer, som var det vi ville vise.

Appendix B – Influens funksjoner for krav, skadeår i og totalt reserve

Influens funksjoner for hhv utestående krav, skadeår i og total reserve vil tilsvare influens funksjoner for funksjonalene $T_{\alpha_i\beta_j}$, $T_{\sum_{j=n-i+2}^n \alpha_i\beta_j}$ og $T_{\sum_{i=2}^n \sum_{j=n-i+2}^n \alpha_i\beta_j}$, som kan skrives på følgende måte:

$$T_{\alpha_i\beta_j} = T_{\alpha_i} T_{\beta_j}, \quad i=1, \dots, n \text{ og } j=1, \dots, n$$

$$T_{\sum_{j=n-i+2}^n \alpha_i\beta_j} = \sum_{j=n-i+2}^n T_{\alpha_i} T_{\beta_j}, \quad i=1, \dots, n$$

$$T_{\sum_{i=2}^n \sum_{j=n-i+2}^n \alpha_i\beta_j} = \sum_{i=2}^n \sum_{j=n-i+2}^n T_{\alpha_i} T_{\beta_j}$$

Vi kan videre finne influens funksjonene for disse uttrykkene ved å bruke enkle derivasjonsregler på uttrykkene for influens funksjoner vi kom frem til tidligere.

For utestående krav (i,j) får vi da ved å bruke produktregelen:

$$\begin{aligned} (4.7) \quad \text{IF}([z, p, q]; T_{\alpha_i\beta_j}, P) &= \frac{d}{d\epsilon} T_{\alpha_i\beta_j}(P_{p,q,\epsilon,z})|_{\epsilon=0} \\ &= \frac{d}{d\epsilon} [T_{\alpha_i}(P_{p,q,\epsilon,z})|_{\epsilon=0} \cdot T_{\beta_j}(P_{p,q,\epsilon,z})|_{\epsilon=0}] \\ &= T_{\alpha_i}(P_{p,q,\epsilon,z})|_{\epsilon=0} \cdot \text{IF}([z, p, q]; T_{\beta_j}, P) + T_{\beta_j}(P_{p,q,\epsilon,z})|_{\epsilon=0} \cdot \text{IF}([z, p, q]; T_{\alpha_i}, P) \\ &= \alpha_i \cdot \text{IF}([z, p, q]; T_{\beta_j}, P) + \beta_j \cdot \text{IF}([z, p, q]; T_{\alpha_i}, P) \end{aligned}$$

For skadeår i finner vi influensfunksjonen:

$$\begin{aligned} (4.8) \quad \text{IF}([z, p, q]; T_{\sum_{j=n-i+2}^n \alpha_i\beta_j}, P) &= \frac{d}{d\epsilon} T_{\sum_{j=n-i+2}^n \alpha_i\beta_j}(P_{p,q,\epsilon,z})|_{\epsilon=0} \\ &= \frac{d}{d\epsilon} \sum_{j=n-i+2}^n [T_{\alpha_i}(P_{p,q,\epsilon,z}) \cdot T_{\beta_j}(P_{p,q,\epsilon,z})]|_{\epsilon=0} \\ &= \sum_{j=n-i+2}^n \frac{d}{d\epsilon} [T_{\alpha_i}(P_{p,q,\epsilon,z}) \cdot T_{\beta_j}(P_{p,q,\epsilon,z})]|_{\epsilon=0} \\ &= \sum_{j=n-i+2}^n \text{IF}([z, p, q]; T_{\alpha_i\beta_j}, P) \end{aligned}$$

Og for den totale reserven finner vi influens funksjonen

$$\begin{aligned}
(4.9) \quad \text{IF}([z, p, q]; T_{\sum_{i=2}^n \sum_{j=n-i+2}^n \alpha_i \beta_j}, P) &= \frac{d}{d\varepsilon} T_{\sum_{i=2}^n \sum_{j=n-i+2}^n \alpha_i \beta_j} (P_{p,q,\varepsilon,z}) \Big|_{\varepsilon=0} \\
&= \frac{d}{d\varepsilon} \sum_{i=2}^n T_{\sum_{j=n-i+2}^n \alpha_i \beta_j} (P_{p,q,\varepsilon,z}) \Big|_{\varepsilon=0} \\
&= \sum_{i=2}^n \text{IF}([z, p, q]; T_{\sum_{j=n-i+2}^n \alpha_i \beta_j}, P)
\end{aligned}$$

Og vi har da kommet frem til uttrykkene (4.7), (4.8) og (4.9).

Appendix C – Utledning influens funksjon for Bornhuetter-Ferguson

La

$$T_{\theta}(P) = E_P \left[\sum_{j=1}^n \frac{\sum_{i=1}^{n-j+1} X_{i,j}}{\sum_{i=1}^{n-j+1} P_i} \right] = \theta \quad \text{og} \quad T_{\beta_j}(P) = E_P \left[\frac{\sum_{i=1}^{n-j+1} X_{i,j}}{\hat{\theta} \sum_{i=1}^{n-j+1} P_i} \right] = \beta_j$$

være statistiske funksjonaler for θ og β_j , som gjelder for $j=1, \dots, n$.

Finner videre influens funksjonene j.f. definisjonen tidligere i oppgaven, (4.2), og trenger da $T_{\theta}(P_{p,q,\varepsilon,z})$ og $T_{\beta_j}(P_{p,q,\varepsilon,z})$.

Som i (4.4) lar vi

$X_{ij} \sim P_{p,q,\varepsilon,z}$ hvis

$$\begin{cases} X_{ij} \sim P_{ij} - \forall (i, j) \neq (p, q) \\ X_{ij} \sim (1 - \varepsilon)P_{ij} + \varepsilon z \end{cases}$$

Og vi får da

$$\begin{aligned} T_{\theta}(P_{p,q,\varepsilon,z}) &= E_{P_{p,q,\varepsilon,z}} \left[\sum_{j=1}^n \frac{\sum_{i=1}^{n-j+1} X_{i,j}}{\sum_{i=1}^{n-j+1} P_i} \right] \\ &= E_{P_{p,q,\varepsilon,z}} \left[\sum_{j=1, j \neq q}^n \left(\frac{\sum_{i=1, i \neq p}^{n-j+1} X_{i,j}}{\sum_{i=1}^{n-j+1} P_i} \right) + \frac{\sum_{i=1, i \neq p}^{n-q+1} X_{i,q}}{\sum_{i=1}^{n-q+1} P_i} + \frac{X_{p,q}}{\sum_{i=1}^{n-q+1} P_i} \right] \\ &= \sum_{j=1, j \neq q}^n \left(\frac{\sum_{i=1, i \neq p}^{n-j+1} P_i \hat{\theta} \hat{\beta}_j}{\sum_{i=1}^{n-j+1} P_i} \right) + \frac{\sum_{i=1, i \neq p}^{n-q+1} P_i \hat{\theta} \hat{\beta}_q}{\sum_{i=1}^{n-q+1} P_i} + \frac{P_p \hat{\theta} \hat{\beta}_q (1 - \varepsilon) + \varepsilon z}{\sum_{i=1}^{n-q+1} P_i} \\ &= \sum_{j=1}^n \left(\frac{\sum_{i=1}^{n-j+1} P_i \hat{\theta} \hat{\beta}_j}{\sum_{i=1}^{n-j+1} P_i} \right) + \frac{(z - P_p \hat{\theta} \hat{\beta}_q) \varepsilon}{\sum_{i=1}^{n-q+1} P_i} \\ &= \hat{\theta} + \frac{(z - P_p \hat{\theta} \hat{\beta}_q) \varepsilon}{\sum_{i=1}^{n-q+1} P_i} \end{aligned}$$

Vi finner så influens funksjonen som

$$\begin{aligned}
 \text{IF}([z, p, q]; T_\theta, P) &= \frac{\partial}{\partial \varepsilon} T_\theta(P_{p,q,\varepsilon,z}) \Big|_{\varepsilon=0} \\
 &= \frac{\partial}{\partial \varepsilon} \left[\hat{\theta} + \frac{(z - p_p \hat{\theta} \hat{\beta}_q) \varepsilon}{\sum_{i=1}^{n-q+1} p_i} \right] \Big|_{\varepsilon=0} \\
 &= \frac{(z - p_p \hat{\theta} \hat{\beta}_q)}{\sum_{i=1}^{n-q+1} p_i}
 \end{aligned}$$

Vi gjør tilsvarende for $T_{\beta_j}(P_{p,q,\varepsilon,z})$, men ser nå på situasjonene når $j=q$ og $j \neq q$:

$j = q$:

$$\begin{aligned}
 T_{\beta_q}(P_{p,q,\varepsilon,z}) &= E_{P_{p,q,\varepsilon,z}} \left[\frac{\sum_{i=1}^{n-q+1} X_{i,q}}{T_\theta(P_{p,q,\varepsilon,z}) \cdot \sum_{i=1}^{n-q+1} p_i} \right] \\
 &= E_{P_{p,q,\varepsilon,z}} \left[\frac{\sum_{i=1, i \neq p}^{n-q+1} X_{i,q} + X_{p,q}}{T_\theta(P_{p,q,\varepsilon,z}) \cdot \sum_{i=1}^{n-q+1} p_i} \right] \\
 &= \frac{\sum_{i=1, i \neq p}^{n-q+1} p_i \hat{\theta} \hat{\beta}_q + p_p \hat{\theta} \hat{\beta}_q (1 - \varepsilon) + \varepsilon z}{T_\theta(P_{p,q,\varepsilon,z}) \cdot \sum_{i=1}^{n-q+1} p_i} \\
 &= \frac{\sum_{i=1}^{n-q+1} p_i \hat{\theta} \hat{\beta}_q + (z - p_p \hat{\theta} \hat{\beta}_q) \varepsilon}{T_\theta(P_{p,q,\varepsilon,z}) \cdot \sum_{i=1}^{n-q+1} p_i} \\
 &= \frac{\hat{\theta} \hat{\beta}_q}{T_\theta(P_{p,q,\varepsilon,z})} + \frac{(z - p_p \hat{\theta} \hat{\beta}_q) \varepsilon}{T_\theta(P_{p,q,\varepsilon,z}) \cdot \sum_{i=1}^{n-q+1} p_i}
 \end{aligned}$$

$j \neq q$:

$$\begin{aligned}
 T_{\beta_j}(P_{p,q,\varepsilon,z}) &= E_{P_{p,q,\varepsilon,z}} \left[\frac{\sum_{i=1}^{n-j+1} X_{i,j}}{T_{\theta}(P_{p,q,\varepsilon,z}) \cdot \sum_{i=1}^{n-j+1} p_i} \right] \\
 &= \frac{\sum_{i=1}^{n-j+1} p_i \hat{\theta} \hat{\beta}_j}{T_{\theta}(P_{p,q,\varepsilon,z}) \cdot \sum_{i=1}^{n-j+1} p_i} \\
 &= \frac{\hat{\theta} \hat{\beta}_j}{\underline{\underline{T_{\theta}(P_{p,q,\varepsilon,z})}}}
 \end{aligned}$$

Og influens funksjonene blir da:

$j = q$:

$$\begin{aligned}
 \text{IF}([z, p, q]; T_{\beta_q}, P) &= \frac{\partial}{\partial \varepsilon} T_{\beta_q}(P_{p,q,\varepsilon,z}) \Big|_{\varepsilon=0} \\
 &= \frac{\partial}{\partial \varepsilon} \left(\frac{\hat{\theta} \hat{\beta}_q}{T_{\theta}(P_{p,q,\varepsilon,z})} + \frac{(z - p_p \hat{\theta} \hat{\beta}_q) \varepsilon}{T_{\theta}(P_{p,q,\varepsilon,z}) \cdot \sum_{i=1}^{n-q+1} p_i} \right) \Big|_{\varepsilon=0} \\
 &= - \frac{\hat{\theta} \hat{\beta}_q \cdot \text{IF}([z, p, q]; T_{\theta}, P)}{(T_{\theta}(P_{p,q,\varepsilon,z}))^2} \Big|_{\varepsilon=0} + \frac{(z - p_p \hat{\theta} \hat{\beta}_q) \cdot T_{\theta}(P_{p,q,\varepsilon,z}) \cdot \sum_{i=1}^{n-q+1} p_i - (z - p_p \hat{\theta} \hat{\beta}_q) \varepsilon \cdot \text{IF}([z, p, q]; T_{\theta}, P) \cdot \sum_{i=1}^{n-q+1} p_i}{\left(T_{\theta}(P_{p,q,\varepsilon,z}) \cdot \sum_{i=1}^{n-q+1} p_i \right)^2} \Big|_{\varepsilon=0} \\
 &= - \frac{\hat{\theta} \hat{\beta}_q \cdot \text{IF}([z, p, q]; T_{\theta}, P)}{\hat{\theta}^2} + \frac{(z - p_p \hat{\theta} \hat{\beta}_q) \cdot \hat{\theta} \cdot \sum_{i=1}^{n-q+1} p_i}{\left(\hat{\theta} \cdot \sum_{i=1}^{n-q+1} p_i \right)^2} \\
 &= - \frac{\hat{\beta}_q \cdot \text{IF}([z, p, q]; T_{\theta}, P)}{\hat{\theta}} + \frac{(z - p_p \hat{\theta} \hat{\beta}_q)}{\hat{\theta} \cdot \sum_{i=1}^{n-q+1} p_i} \\
 &= \underline{\underline{\frac{(1 - \hat{\beta}_q)}{\hat{\theta}} \cdot \text{IF}([z, p, q]; T_{\theta}, P)}}
 \end{aligned}$$

$j \neq q$:

$$\begin{aligned}
 \text{IF}([z, p, q]; T_{\beta_j}, P) &= \frac{\partial}{\partial \epsilon} T_{\beta_j}(P_{p,q,\epsilon,z}) \Big|_{\epsilon=0} \\
 &= \frac{\partial}{\partial \epsilon} \left(\frac{\hat{\theta} \hat{\beta}_j}{T_{\theta}(P_{p,q,\epsilon,z})} \right) \Big|_{\epsilon=0} \\
 &= - \frac{\hat{\theta} \hat{\beta}_j \cdot \text{IF}([z, p, q]; T_{\theta}, P)}{(T_{\theta}(P_{p,q,\epsilon,z}))^2} \Big|_{\epsilon=0} \\
 &= - \frac{\hat{\beta}_j}{\hat{\theta}} \cdot \text{IF}([z, p, q]; T_{\theta}, P)
 \end{aligned}$$

Når det gjelder $\text{IF}([z, p, q]; T_{p_i \theta \beta_j}, P) = \frac{\partial}{\partial \epsilon} T_{p_i \theta \beta_j}(P_{p,q,\epsilon,z}) \Big|_{\epsilon=0}$, som er influensen på selve kravet i posisjon (i,j), der $T_{p_i \theta \beta_j}(P_{p,q,\epsilon,z}) = p_i T_{\theta}(P_{p,q,\epsilon,z}) T_{\beta_j}(P_{p,q,\epsilon,z})$, gir dette

$$\begin{aligned}
 \text{IF}([z, p, q]; T_{p_i \theta \beta_j}, P) &= \frac{\partial}{\partial \epsilon} T_{p_i \theta \beta_j}(P_{p,q,\epsilon,z}) \Big|_{\epsilon=0} \\
 &= \frac{\partial}{\partial \epsilon} [p_i T_{\theta}(P_{p,q,\epsilon,z}) T_{\beta_j}(P_{p,q,\epsilon,z})] \Big|_{\epsilon=0} \\
 &= p_i [T_{\theta}(P_{p,q,\epsilon,z}) \text{IF}([z, p, q]; T_{\beta_j}, P) + T_{\beta_j}(P_{p,q,\epsilon,z}) \text{IF}([z, p, q]; T_{\theta}, P)] \Big|_{\epsilon=0}
 \end{aligned}$$

og ved å sette inn uttrykkene vi har funnet over, får vi:

$$\text{IF}([z, p, q]; T_{p_i \theta \beta_j}, P) = \begin{cases} p_i [\hat{\theta} \cdot \text{IF}([z, p, q]; T_{\beta_q}, P) + \hat{\beta}_q \text{IF}([z, p, q]; T_{\theta}, P)] \\ 0, \forall j \neq q \end{cases}$$

Altså er influensen av en endring z i punkt $(p,q) = 0$ for $j \neq q$, noe som gir mening j.f. (4.10)

11. Programkode

Jeg har gjort mye programmering i R knyttet til denne oppgaven, men velger å ta med kun den mest relevante koden i denne oppgave.

R-kode brukt ved en robuste Chain-Ladder metoden, kapittel 5.1:

```
Robust_CL = function(data,r,alpha){
# input:
# data = trekanten med inkrementelle krav, NA for fremtidige krav
# r = 0/1 (linear / exponential regresjon for utviklingsfaktor i steg 5 )
# alpha = signifikansnivå ved sjekk av uteliggere i  $X_{1,n}$   $X_{1,n-1}$   $X_{2,n-1}$ 

r=r
alpha=alpha
out.data=data # lagrer input data
fal = find.utvF(data)

n.skadeAr = nrow(data)
n.utvAr = ncol(data)
n=n.skadeAr

cum.data = create.cum.data(data) # lager cumulative trekant

##### STEP 1 #####
# utviklingsfaktorer
ut.factor = find.rob.cum.utvF(data)
cum.fit.data = cum.data
# lager fitted cumulative data ved recursiv deling på utv.factor
for(i in 1:(n.utvAr-1)){
  for(j in (n.utvAr-i):1){
    cum.fit.data[i,j] = cum.fit.data[i,j+1]/ut.factor[j]
  }
}
# incremental fitted values
inc.fit.data = create.inc.data(cum.fit.data) # lager inkrementell trekant

##### STEP 2 #####
disp=1
# spiller ingen rolle om vi tar denne med, siden kun bruker pearson residuals for å sjekke
etter outlying value

# pearson residuals

p.resid = (data-inc.fit.data) / sqrt(disp*inc.fit.data)
p.resid.values = p.resid[!is.na(p.resid)]
Q1 = quantile(p.resid.values, 0.25)
Q3 = quantile(p.resid.values, 0.75)
lowL = Q1-3*IQR(p.resid.values)
upL = Q3+3*IQR(p.resid.values)

#lager matrix for outlying residuals
outliers = matrix(0, nrow=n.skadeAr, ncol=n.utvAr)
for(i in 1:n.skadeAr){
  for(j in 1:n.utvAr){
    if(!is.na(p.resid[i,j])){
      if((p.resid[i,j]>=upL) | (p.resid[i,j]<=lowL))
        outliers[i,j]=1;
    }
  }
}

## sjekker om data[n,1] er outlying, og erstatter evnt med median fra første kolonne
m1 = median(data[,1])

Q1 = quantile(data[,1], 0.25)
Q3 = quantile(data[,1], 0.75)
lowL = Q1-3*IQR(data[,1])
upL = Q3+3*IQR(data[,1])

if((data[n,1]>upL) | (data[n,1]<lowL)){
  data[n,1]=m1
}

##### STEP 3 #####
temp=0;
for(i in 1:(n.skadeAr-1)){
  temp[i]=data[i,2]/data[i,1]
}
```

```

ny.data=data;
ny.cum.data=cum.data

for(i in 1:n.skadeAr){
  if(outliers[i,1]==1){
    if(outliers[i,2]==1){
      ny.data[i,1] = median(ny.cum.data[,1])
    }
    else{
      ny.data[i,1] = ny.data[i,2] / median(temp)
    }
  }
}

##### STEP 4 #####
#nye utviklingsfactorer based on incremental claims mot første kolonne

ny.faktor = find.rob.utvF.coll(ny.data)

# lager fitted data utifra kolonne en som har ingen outliers
# og robuste utviklingsfaktorer

ny.fit.data = ny.data;

for(i in 1:(n.skadeAr-1)){
  for(j in 1:(n.utvAr-i)){
    ny.fit.data[i,j+1] = ny.data[i,1]*ny.faktor[j]
  }
}

# finner nye residualer, og sjekker for outliers i resten av data

p.resid2 = (ny.data-ny.fit.data) / sqrt(disp*ny.fit.data)
p.resid2.values = p.resid2 [!is.na(p.resid2 )]
Q1 = quantile(p.resid2.values, 0.25)
Q3 = quantile(p.resid2.values, 0.75)
lowL = Q1-3*IQR(p.resid2.values)
upL = Q3+3*IQR(p.resid2.values)

#lager matrix for hva som er outlying residuals
outliers2 = matrix(0, nrow=n.skadeAr, ncol=n.utvAr)
for(i in 1:n.skadeAr){
  for(j in 1:n.utvAr){
    if(!is.na(p.resid2[i,j])){
      if((p.resid2[i,j]>=upL) | (p.resid2[i,j]<=lowL)){
        outliers2[i,j]=1;
      }
    }
  }
}
outliers2[1,(n-1)]=0;
outliers2[1,n]=0;
outliers2[2,(n-1)]=0;

# erstatter outliers med median av residuals

for(i in 1:(n.skadeAr-1)){
  for(j in 2:n.utvAr){
    if(outliers2[i,j]==1)
      p.resid2[i,j] = median(p.resid2.values)
  }
}

# backtransform to datamatrix

rob.data = (p.resid2 * sqrt(disp*ny.fit.data)) + ny.fit.data

##### STEP 5 #####
# apply classical chain ladder method
#chain_ladder(rob.data)
#chain_ladder(org.data)

fa2 = find.utvF(rob.data)

### siste steg for å finne outliers i  $x_{1,n}$   $x_{1,n-1}$   $x_{2,n-1}$ 
### to siste i rad 1, siste i rad 2

test.data = create.cum.data(rob.data)

a = test.data[1,(n-1)] / test.data[1,(n-2)]
b = test.data[2,(n-1)] / test.data[2,(n-2)]

f = find.utvF(rob.data)

```

```

x=0;
y=0;
for(i in 2:n.utvAr){
  x[i-1] = 1/i;
  y[i-1] = exp(-i);
}

dat = data.frame(Y=f[1:7],X=x[1:7],X2=y[1:7])
mod1 = lm(Y ~ X, dat)
mod2 = lm(Y ~ X2, dat)

# linear model (runofftriangles) eller exp model (tayAshe)

p.lin = predict(mod1,data.frame(X=0.1111111), interval="prediction")
p.exp = predict(mod2,data.frame(X2=exp(-9)), interval="prediction")

if(r==1){
  fit1 = p.exp[1]
} else{
  fit1 = p.lin[1]
}

up=fit1*(1+alpha) # avvik kan maks være 5 % opp eller ned fra fitted value før vi sier vi har
outlier
low=fit1*(1-alpha)

# a1 for å lagre om X[1,n-1] er uteligger eller ikke.
a1=0
if(((b>up) | (b<low)) && ((a>up) | (a<low))){
  a=b=fit1
  a1=1
}
if((b>up) | (b<low) ){
  b=a
}
if((a>up) | (a<low) ){
  a=b
  a1=1
}
test.data[1,(n-1)] = test.data[1,(n-2)]*a
test.data[2,(n-1)] = test.data[2,(n-2)]*b

rob.data2 = create.inc.data(test.data)
rob.data2[1,n] = out.data[1,n]

#chain_ladder(rob.data2)

fa3 = find.utvF(rob.data2)

## test for outlier i X[1,10]

test.data2 = create.cum.data(rob.data2)
n2=8

c = test.data2[1,n] / test.data2[1,(n-1)]

if(a1==0){
dat2 = data.frame(Y=fa3[1:(n-2)],X=x[1:(n-2)],X2=y[1:(n-2)])
}

if(a1==1){
dat2 = data.frame(Y=c(fa3[1:(n-3)],a),X=x[1:(n-2)],X2=y[1:(n-2)])
}

mod3 = lm(Y ~ X, dat2)
mod4 = lm(Y ~ X2, dat2)
p.lin2 = predict(mod3,data.frame(X=0.1), interval="prediction")
p.exp2 = predict(mod4,data.frame(X2=exp(-10)), interval="prediction")

if(r==1){
  fit2 = p.exp2[1]
} else{
  fit2 = p.lin2[1]
}
up2=fit2*(1+alpha)
low2=fit2*(1-alpha)

#hvis outlier i X[1,9] må neste utv.f. endres, da denne ikke vil stemme.
if(a1==1){
  test.data2[1,n] = test.data2[1,(n-1)] * fit2
}

if(a1==0){
  if((c>up2)|(c<low2)){

```

```

        test.data2[1,n] = test.data2[1,(n-1)]*fit2
    }
}
rob.data3 = create.inc.data(test.data2)

#chain_ladder(rob.data3)
#chain_ladder(out.data)

# oversikt over outliers, hvor de befinner seg
t.outliers = matrix(0,n.skadeAr,n.utvAr)
for(i in 1:n.skadeAr)
{
    for(j in 1:(n.utvAr-i+1))
    {
        if(round(rob.data3[i,j])!=round(out.data[i,j]))
        {
            t.outliers[i,j]=1
        }
    }
}

fa4 = find.utvF(rob.data3)

# Tot reserve rob chain ladder og klassisk chain ladder
resCL=chain_ladder(out.data)[(nrow(data)+1),(ncol(data)+1)]
resRCL =chain_ladder(rob.data3)[(n+1),(n+1)]

# returnerer trekant med 1 på kravene som betraktes som uteliggere
return(t.outliers)
}

```

R-kode til funksjon som finner robuste utviklingsfaktorer (se Step 1 – oversikt over stegene, kapittel 5.1):

```

find.rob.cum.utvF = function(data){
data = as.matrix(data)
n.skade = nrow(data)
n.utv = ncol(data)

ut.factor=0
for(i in 1:(n.utv-1)){
    temp=0;
    for(j in 1:(n.skade-i)){
        temp[j]=cum.data[j,i+1]/cum.data[j,i]
    }
    ut.factor[i]=median(temp)
}

ut.factor
}

```

R-kode for bootstrapping av Chain-Ladder (for 10x10 trekanter):

```

cl.bootstrap = function(data, N){
n.row = nrow(data)
n.col = ncol(data)
f = find.utvF(data)

cumfit = create.cum.data(data)

for(i in 1:(n.utvAr-1)){
    for(j in (n.utvAr-i):1){
        cumfit[i,j] = cumfit[i,j+1]/f[j]
    }
}
incfit = create.inc.data(cumfit)
resid = (data - incfit) / sqrt(incfit)
resid[1,10]=resid[10,1]=0;
p=as.vector(resid)
p=na.exclude(p)
n=length(p)+2;

p=p[!as.logical(p==0)]

disp=sum(p^2)/36

resid2=resid
reserve=matrix(0,N,10)
for(k in 1:N){
    for(i in 1:n.row){
        for(j in 1:(n.col-i+1)){

```

```

        resid2[i,j] = sample(p,1,replace=T)
      }
    }
    nydata = resid2*sqrt(incfit)+incfit
    temp=chain_ladder(nydata)
    reserve[k,1] = temp[2,11]
    reserve[k,2] = temp[3,11]
    reserve[k,3] = temp[4,11]
    reserve[k,4] = temp[5,11]
    reserve[k,5] = temp[6,11]
    reserve[k,6] = temp[7,11]
    reserve[k,7] = temp[8,11]
    reserve[k,8] = temp[9,11]
    reserve[k,9] = temp[10,11]
    reserve[k,10] = temp[11,11]
  }

  org.reserve = chain_ladder(data)[(2:11),11]
  sd.reserve=0
  pred=0;
  m.reserve=0;

  for(i in 1:10){
    sd.reserve[i] = sd(reserve[,i])
    pred[i] = sqrt(dis*org.reserve[i] + (55/36)*sd.reserve[i]^2)
    m.reserve[i]=mean(reserve[,i])
  }

  r = cbind(m.reserve,pred)
  return(r)
}

```

R-kode til beregning av influens funksjon for Chain-Ladder, kaptittel 4.4:

```

if_chainladder = function(data){
  data_org=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))

  CS = tapply(claims[!is.na(claims)],dev[!is.na(claims)],sum)
  RS = tapply(claims[!is.na(claims)],origin[!is.na(claims)],sum)

  alpha = beta = numeric(n.origin)
  aa = alpha[1] = RS[1]
  bb = beta[n.origin] = CS[n.origin]/RS[1]
  for(n in 2:n.origin){
    aa=aa+(alpha[n]=RS[n]/(1-bb))
    bb=bb+(beta[n.origin-n+1]=CS[n.origin-n+1]/aa)
  }
  A=alpha
  B=beta

  XX = cbind(claims,origin,dev)
  XX = XX[!is.na(XX[,1]),]
  xij=XX[,1]
  i=XX[,2]
  j=XX[,3]
  n=length(A)
  K=length(XX[,1])

  z=2 # multipliseringsfaktor fra oppgaven

  EIF = matrix(0,nrow=K,ncol=(length(A)+length(B)))
  for(l in 1:n){
    for(k in 1:K){
      if(l==1){
        dum1=dum2=0; # pga sum under ikke fungerer når l=1 bounds)
      }
      else{
        dum1=sum(EIF[k,(n+n-1+2):(n+n)])
        dum2=sum(B[(n-1+2):n])
      }
      # erstatter xij[k] med 1
      if(l==i[k]){
        EIF[k,1]=((A[i[k]]*B[j[k]]*z)-A[1]*B[j[k]]+A[1]*dum1)/(1-dum2)
      }
    }
  }
}

```

```

else{
  EIF[k,1]=(A[1]*dum1)/(1-dum2)
}
if((n-1+1)==j[k]){
  EIF[k,n+(n-1+1)]=((A[i[k]]*B[j[k]]*z)-A[i[k]]*B[n-1+1]-B[n-
1+1]*sum(EIF[k,1:1]))/sum(A[1:1])
}
else{
  EIF[k,n+(n-1+1)]=(-B[n-1+1]*sum(EIF[k,1:1]))/sum(A[1:1])
}
}
}
EIF_ab = matrix(0, nrow=K,ncol=n*n)
for(c in 1:(n*n)){
  for(k in 1:K){
    EIF_ab[k,c]=A[origin[c]]*EIF[k,(n+as.numeric(dev[c]))]+B[dev[c]]*EIF[k,as.numeric(ori
n[c])]
  }
}
EIF_row = matrix(0,nrow=K,ncol=(n-1))
for(r in 2:n){
  for(k in 1:K){
    dum=EIF_ab[k,(origin==r)]
    EIF_row[k,(r-1)]=sum(dum[(n-r+2):n])
  }
}
EIF_R = rep(0,K)
for(k in 1:K){
  EIF_R[k]=sum(EIF_row[k,])
}
test=cbind(EIF_R,i,j)
matrix = matrix(NA,nrow=n,ncol=n)
for(p in 1:n){
  for(q in 1:(n-p+1)){
    matrix[p,q]=as.numeric(test[((test[,2]==p)&(test[,3]==q)),1])
  }
}
R = chain_ladder(data_org)[n.origin,n.origin]
round(matrix/R*100,2) # IF i hvert punkt på total reserve

```

12. Kilder

- [1] Verdonck, T., Van Wouwe, M., Dhaene, J. (2009). *A ROBUSTIFICATION OF THE CHAIN-LADDER METHOD*. North American Actuarial Journal, Volume 13 – Number 2.
- [2] England, P., Verrall, R. (1999) *Analytic and bootstrap estimates of prediction errors in claims reserving*. Insurance: Mathematics and Economics 25 (1999) 281–293
- [3] De Silva, N. *An Introduction to R: Examples for Actuaries* 2006.
<http://toolkit.pbworks.com/R-Examples-For-Actuaries>. {side 30-36}
- [4] Mack, T. (1993). *Distribution Free Calculation of the Standard Error of Chain-Ladder Reserve Estimates*. ASTIN Bulletin 23(2): 213-225.
- [5] Sagosen, M. (2010). *Chain Ladder Metoden og Mack's Modell sammenlignet med Poissonbaserte modeller*. M.Sc.Thesis, University of Bergen.
- [6] Verdonck, T. (2009). *New robust methods in multivariate statistics and actuarial science*. PhD-Defence, Antwerpen 2009.
- [7] Verdonck, T., Debruyne M. (2009). *The influence of individual claims on the chain-ladder estimates: Analysis and diagnostic tool*. Insurance: Mathematics and Economics 48 (2011) 85-98
- [8] G.G. Venter, D.R. Tampubolon (2008). *Robustifying Reserving*. Casualty Actuarial Society E-Forum, Fall 2008,
http://www.casact.org/pubs/forum/08fforum/16Venter_Tampubolon.pdf
- [9] W. Neuhaus (2006). *Lecture Notes on Estimating Outstanding Claims in General Insurance*. Unpublished lecture notes.
- [10] Casella G., Berger R.L. (2002). *Statistical Inference*. 2nd edition – Duxbury. Side 517-518.
- [11] Hess, K., Schmidt, K. (2002) *A comparison of models for the chain-ladder method*. Insurance: Mathematics and Economics 31 (2002) 351-364.

- [12] Kaas, R., Goovaerts, M., Dhaene, J., Denuit, M. (2008) *Modern Actuarial Risk Theory – Using R*. 2nd edition – Springer.
- [13] Pison, G., Van Aelst, S., (2004) *Diagnostic plots for robust multivariate methods*. Journal of Computational and Graphical Statistics 13, 310-329.
- [14] Verrall, R.J., Li, Z., (1993) *Negative incremental claims: Chain Ladder and Linear Models*. JIA (Journal of the Institute of Actuaries) 120, s171-183.
- [15] Dobson, A.J., Barnett, A.G., (2008) *An Introduction to Generalized Linear Models*, 3rd edition. Chapman & Hall / CRC