ACSC/STAT 4703, Actuarial Models II

FALL 2024
Toby Kenney

Homework Sheet 7

Model Solutions

Basic Questions

1. *The following table shows the paid losses (in thousands) on claims from one line of business of an insurance company over the past 5 years.*

Accident year	*Earned premiums*	*Development year*				
		0	*1*	*2*	*3*	*4*
2019	*40961*	*6785*	*13970*	*6914*	*3422*	*1669*
2020	*37144*	*7941*	*9810*	*6287*	*6252*	
2021	*41814*	*9894*	*13919*	*7782*		
2022	*62205*	*9454*	*15881*			
2023	*82847*	*11739*				

This table is in the file `HW7Q1.txt`

Assume that all payments on claims arising from accidents in 2019 have now been settled. Estimate the future payments arising each year from open claims arising from accidents in each calendar year using

(a) The chain-ladder method

We first compute the cumulative run-off triangle:

Accident year	Development year				
	0	1	2	3	4
2018	6785	20755	27669	31091	32760
2019	7941	17751	24038	30290	
2020	9894	23813	31595		
2021	9454	25335			
2022	11739				

Then we compute the loss development factors:

0/1	$\frac{87654}{34074} = 2.572460$
1/2	$\frac{83302}{62319} = 1.336703$
2/3	$\frac{61381}{51707} = 1.187093$
3/4	$\frac{32760}{31091} = 1.053681$

Using these values to complete the table gives the following cumulative losses:

Accident	Development year				
year	0	1	2	3	4
LDF		2.05600520979	1.35041171089	1.11404986523	1.11302023673
2020				**30290.00**	31916.00
2021			**31595.00**	37506.19	39519.57
2022		**25335.00**	33865.37	40201.34	42359.39
2023	**11739**	30198.11	40365.90	47918.07	50490.36

The future payments are the differences between consecutive years:

Accident	Development year				
year	0	1	2	3	4
2019					1626.001
2020				5911.192	2013.375
2021			8530.373	6335.963	2158.053
2022		18459.107	10167.796	7552.164	2572.296

(b) The Bornhuetter-Ferguson method with expected loss ratio 0.77.

Using the LDFs from part (a), we get the following:

Development Year	Cumulative proportion of losses paid	Proportion of losses paid
0	0.2324998	0.23249981
1	0.5980965	0.36559664
2	0.7994774	0.20138092
3	0.9490537	0.14957635
4	1.0000000	0.05094628

This gives the following reserves:

Accident	Expected Ult.	Development year				
year	Losses	0	1	2	3	4
2020	28600.88					1457.108
2021	32196.78				4815.877	1640.306
2022	47897.85			9645.713	7164.385	2440.217
2023	63792.19		23322.21	12846.530	9541.803	3249.975

(c) The Bühlmann-Straub estimate.

We first recalculate the per-premium loss development factors and $\hat{\gamma}_j$, using the same method as parts (a) and (b):

```
### Calculate per-premium losses
HW7Q1pp<-HW7Q1[,-1]/HW7Q1$Earned_premiums

HW7Q1cumpp<-t(apply(HW7Q1pp,1,cumsum)) #cumulative run-off

HW7Q1LDFpp<-colSums(HW7Q1cumpp,na.rm=TRUE)[-1]/
    (colSums(HW7Q1cumpp,na.rm=TRUE)[-5]-HW7Q1cumpp[1+4*seq_len(4)])

beta<-1/c(rev(cumprod(rev(HW7Q1LDFpp))),1)
gamma<-beta-c(0,beta[-5])
HW7Q1est_ult_pp<-c(1,cumprod(rev(HW7Q1LDFpp)))*rev(HW7Q1cumpp[1+4*seq_len(5)])
```

This gives the following values for γ_j:

j	γ_j
0	0.23344681
1	0.36272042
2	0.20107494
3	0.15181155
4	0.05094628

and for $\widehat{C_{i,J}}$:

i	$\widehat{C_{ij}}$
2019	0.7997852
2020	0.8592505
2021	0.9477775
2022	0.6831680
2023	0.6069688

We use the formulae:

$$\hat{v} = \frac{1}{I}\sum_{i=0}^{I-1}\frac{1}{I-i}\sum_{j=0}^{I-i}\hat{\gamma}_j\left(\frac{X_{ij}}{\hat{\gamma}_j} - \hat{C_{i,J}}\right)^2 \qquad \hat{a} = \frac{\sum_{i=0}^{I}\hat{\beta}_{I-i}\left(\hat{C}_{i,J} - \overline{C}\right)^2 - I\hat{v}}{\sum_{i=0}^{I}\hat{\beta}_{I-i} - \frac{1}{\sum_{i=0}^{I}\hat{\beta}_{I-i}}\sum_{i=0}^{I}\hat{\beta}_{I-i}^2}$$

```
gamma.matrix<-(rep(1,5)%*%t(gamma))

hatv<-mean(
     rowSums(gamma.matrix*(HW7Q1pp/gamma.matrix-HW7Q1est_ult_pp)^2,na.rm=TRUE)/
     (4:0),na.rm=TRUE)

### weighted average of antidiagonal elements
Cbar<-sum(HW7Q1cumpp[1+4*seq_len(5)])/sum(beta)


hata<-(sum(rev(beta)*(HW7Q1est_ult_pp-Cbar)^2)-5*hatv)/
     (sum(beta)-sum(beta^2)/sum(beta))

### Calculate credibilities
Z<-rev(beta)/(rev(beta)+hatv/hata)

### Credibility-weighted average
muhat<-sum(Z*HW7Q1est_ult_pp)/sum(Z)

### Buhlmann-Straub per-premium estimates
Chat_BS<-Z*HW7Q1est_ult_pp+(1-Z)*muhat

### per-premium reserves
Chat_BS%*%t(gamma)

### Multiply by earned premiums
Est_Future<-HW7Q1ep*Chat_BS%*%t(gamma)

### Set values in the past to zero
Est_Future[rep(seq_len(5),5)+rep(seq_len(5),each=5)<7]<-NA

### Total reserves
sum(Est_Future,na.rm=TRUE)
```

This gives $\overline{C} = 0.8165322$, $\hat{v} = 0.002829697$, $\hat{a} = 0.008158523$ and the following per-premium payments:

Accident	Development year				
year	0	1	2	3	4
2020					0.04294513
2021				0.1370062	0.04597777
2022			0.1458860	0.1101439	0.03696306
2023		0.2616502	0.1450464	0.1095100	0.03675035

which correspond to the following reserves:

Accident	Development year				
year	0	1	2	3	4
2020					1595.148
2021				5728.765	1922.510
2022			9074.844	6851.506	2299.290
2023		21676.87	12016.628	9072.552	3044.648

This gives the total outstanding claims reserves as $73,282.77.

2. *The file* `HW7_Q2.txt` *contains a run-off triangle. An actuary is planning to use the chain-ladder method to estimate future reserves. Test whether losses in different years are correlated, and whether there are any calendar year effects.*

We first calculate correlation between vectors of development factors in each year.

```
HW7Q2<-read.table("HW7_Q2.txt")
HW7Q2cum<-t(apply(HW7Q2,1,cumsum))
HW7Q2LDF<-HW7Q2cum[,-1]/HW7Q2cum[,-10]

### pairwise correlation matrix
cor(HW7Q2LDF, use="pairwise.complete")

### extract correlations between adjacent development years
r<-cor(HW7Q2LDF, use="pairwise.complete")[2+12*(seq_len(8)-1)]
### This trick is based on matrix indexing in R.  Since the
### correlation matrix is 11x11, increasing the index by 12 gives the
### entries below the diagonal.

### First pair have 10 development factors available. This number
### decreases by 1 for each pair.
tstat<-r*sqrt((8:1)/(1-r^2))

### calculate p-values
pt(-abs(tstat),df=8:1)*2 # multiply by 2 for 2-sided test.
```

This suggests that none of the development factors are correlated.

To check for Calendar Year effects, we rank the development factor estimates.

```
### Rank each column of development factors, and divide by number of
### factors in column.
rank.df<-apply(HW7Q2LDF,2,rank)/rep(11:1,each=12)
### By default NA values are ranked at the end, so NA values get
### turned into larger ranks. We turn them back to NA.
rank.df[rank.df>1]<-NA

### take values in calendar year.
### matrices in R are listed by columns, so we can achieve this by
### changing the dimensions from 12x11 to 11x12
cy.dy<-matrix(rank.df,11,12)

###Count number of times above median, and number of times not equal to median
over.median<-rowSums(cy.dy>0.5,na.rm=TRUE)
not.median<-rowSums(cy.dy!=0.5,na.rm=TRUE)
cbind(over.median,not.median)

###Calculate p-values
pmin(
    pbinom(over.median-0.5,not.median,0.5,lower.tail=FALSE)*2,
    ## if more than half over median
    pbinom(over.median,not.median,0.5)*2
    ## if fewer than half over median
)

###
rowMeans(matrix(rank.df,9,10),na.rm=TRUE)

### standardise columns of rank
rank.sd<-sqrt(colMeans((matrix(rank.df,9,10)-0.5)^2,na.rm=TRUE))
((matrix(rank.df,9,10)-0.5)/(rep(1,9)%*%t(rank.sd)))

rowMeans(((matrix(rank.df,9,10)-0.5)/(rep(1,9)%*%t(rank.sd))),na.rm=TRUE)

### Standardise row means
rowMeans((matrix(rank.df,9,10)-0.5)/(rep(1,9)%*%t(rank.sd)),na.rm=TRUE)*
    sqrt(seq_len(9))
```

Mack's test, gives the following numbers of development factors above the median.

Calendar Year	Number of observations above median	Number of observations not equal to median	p-value
2	1	1	1.0000000
3	2	2	0.5000000
4	2	3	1.0000000
5	0	4	0.1250000
6	2	5	1.0000000
7	3	6	1.3125000
8	6	6	0.0312500
9	4	7	1.0000000
10	7	9	0.1796875
11	4	9	1.0000000
12	6	10	0.7539063

As we can see, none of the calendar years are significant.

The standardised ranks are also not significant.

3. *For the run-off table in Question 1, use Mack's model to estimate the MSE of the estimated outstanding losses.*

We first compute the estimates $\hat{\sigma}_j^2$.

```
beta<-1/c(rev(cumprod(rev(HW7Q1LDF))),1)
gamma<-beta-c(0,beta[-5])

fij<-HW7Q1cum[,-1]/HW7Q1cum[,-5]
sigmahat<-rowMeans(HW7Q1cum[,-5]*(fij-rep(1,5)%*%t(HW7Q1.LDF))^2,na.rm=TRUE)*
    (5-seq_len(5))/(4-seq_len(5))

### Use Mack's suggestion
sigmahat[4]<-min(sigmahat[2],sigmahat[3],sigmahat[3]^2/sigmahat[2])
```

j	$\hat{\sigma}_j^2$
0	572.4625
1	517.9426
2	273.8203
3	144.7603

Next we estimate process variance:

```
cumsum(rev(sigmahat[-5]/beta[-5]/HW7Q1LDF^2))*HW7Q1est_ult
```

This gives the following values.

j	$\widehat{\mathrm{Var}}(C_{i,J}\|C_{i,J-i})$
1	4384790
2	15034519
3	36644938
4	62465070

We then compute the variance of our estimates:

```
S<-colSums(HW7Q1cum, na.rm=TRUE)-HW7Q1cum[1+4*seq_len(5)]

HW7Q1est_ult^2*cumsum(rev(sigmahat[-5]/(HW7Q1LDF^2*S[-5])))
```

This gives the following values.

j	$\mathbb{E}\left(\widehat{C}_{i,J}-\mathbb{E}\left(\widehat{C}_{i,J}\middle\|D\right)\right)$
1	4271825
2	12418790
3	22613945
4	38600837

Finally, we compute the covariance

```
CovarianceMSE<-(HW7Q1est_ult%*%t(HW7Q1est_ult))*
    pmax(cumsum(rev(sigmahat[-5]/(HW7Q1LDF^2*S[-5])))%*%t(rep(1,4)),
         rep(1,4)%*%t(cumsum(rev(sigmahat[-5]/(HW7Q1LDF^2*S[-5])))))

### Total estimation error.
sum(CovarianceMSE)+
    sum(HW7Q1est_ult^2*cumsum(rev(sigmahat[-5]/(HW7Q1LDF^2*S[-5]))))
```

This gives the total

426139468

4. *The files* `HW7Q4_reported.txt`, `HW7Q4_settled.txt` *and* `HW7Q4_aggregate.txt` *give numbers of claims reported and settled, and aggregate claim amounts for each accident year and development year. By using the chain-ladder method to project number of reported claims, propotions of settled claims and average aggregate losses per claim, estimate the reserves needed.*

We first calculate the LDFs and project the number of reported claims:

```
Run.Off.Reported<-read.table("HW7Q4_reported.txt")

Cum.Reported<-t(apply(Run.Off.Reported,1,cumsum))
### Take latest cumulative reported claims for each year
Latest.Cum.Reported<-Cum.Reported[rev(1+9*seq_len(10))]

### Calculate development factors
Reported.DF<-colSums(Cum.Reported,na.rm=TRUE)[-1]/
    ((colSums(Cum.Reported,na.rm=TRUE)-Latest.Cum.Reported)[-10])

### Estimate the ultimate reported claims
Est.Ultimate.Reported<-c(1,cumprod(rev(Reported.DF)))*Latest.Cum.Reported
### And reported claims each year by projecting backwards.
Est.Cum.Reported<-Est.Ultimate.Reported%*%t(rev(1/c(1,cumprod(rev(Reported.DF)))))
```

This projects the number of cumulative reported claims given in the following table:

j	Expected cumulative number of reported claims through Development Year									
	0	1	2	3	4	5	6	7	8	9
2										613.8622
3									550.5834	680.0450
4								571.1241	692.6244	855.4847
5							553.5046	561.4917	680.9428	841.0564
6						559.5541	564.1454	572.2860	694.0334	857.2251
7					580.7589	580.2965	585.0579	593.5003	719.7609	889.0020
8				599.3122	600.0964	599.6186	604.5385	613.2620	743.7267	918.6030
9			606.8212	585.6286	586.3949	585.9280	590.7356	599.2599	726.7458	897.6293
10		639.4449	666.7160	643.4316	644.2735	643.7606	649.0427	658.4084	798.4775	986.2276

We then calculate LDFs for settled claims, and use them to project the number of settled claims.

```
Run.Off.Settled<-read.table("HW7Q4_settled.txt")
Cum.Settled<-t(apply(Run.Off.Settled,1,cumsum))

Settled.DF<-colSums(Cum.Settled[,-1],na.rm=TRUE)/
    (colSums(Cum.Settled[,-10],na.rm=TRUE)-Cum.Settled[1+9*seq_len(9)])

beta<-c(rev(cumprod(rev(1/Settled.DF))),1)
gamma<-beta-c(0,beta[-10])

Projected.Settled<-Est.Ultimate.Reported%*%t(gamma)

### Cover past values
Projected.Settled[seq_len(10)%*%t(rep(1,10))+rep(1,10)%*%t(seq_len(10))<12]<-NA

### Merge Observed and Projected
Observed.Projected.Settled<-
    matrix(rowMeans(cbind(as.vector(as.matrix(Run.Off.Settled)),
                          as.vector(as.matrix(Projected.Settled))),
             na.rm=TRUE),10,10)
```

j	Expected or observed number of settled claims in Development Year									
	0	1	2	3	4	5	6	7	8	9
1	**117**	**177**	**53**	**12**	**41**	**11**	**28**	**6**	**3**	**9**
2	**140**	**184**	**49**	**20**	**42**	**21**	**12**	**18**	**3**	12.08919
3	**117**	**164**	**53**	**28**	**38**	**13**	**25**	**7**	4.268852	13.39257
4	**148**	**214**	**74**	**16**	**26**	**35**	**33**	18.77273	5.370141	16.84762
5	**162**	**192**	**51**	**26**	**45**	**29**	41.49879	18.45611	5.279570	16.56347
6	**150**	**205**	**63**	**37**	**52**	36.70590	42.29657	18.81092	5.381066	16.88189
7	**160**	**212**	**68**	**42**	69.18834	38.06657	43.86448	19.50823	5.580540	17.50770
8	**172**	**231**	**52**	44.32075	71.49210	39.33407	45.32503	20.15779	5.766354	18.09065
9	**161**	**263**	95.07155	43.30882	69.85978	38.43598	44.29017	19.69755	5.634696	17.67760
10	**168**	359.9643	104.45535	47.58351	76.75512	42.22971	48.66172	21.64175	6.190855	19.42242

Finally, we calculate the average claim severity for each development year.

```
Aggregate.Payments<-read.table("HW7Q4_aggregate.txt")


Ave.Settlement.Amount<-colMeans(Aggregate.Payments/Run.Off.Settled,na.rm=TRUE)
Exp.Agg.Payments<-Projected.Settled*(rep(1,10)%*%t(Ave.Settlement.Amount))

### Estimate total reserves
sum(Exp.Agg.Payments,na.rm=TRUE)
```

This gives the following expected aggregate payments:

j	Expected aggregate claims in Development Year									
	0	1	2	3	4	5	6	7	8	9
2										1207.576
3									4825.226	1337.769
4								2026.461	6070.050	1682.890
5							8147.406	1992.284	5967.674	1654.507
6						21564.43	8304.034	2030.584	6082.399	1686.314
7					3945.742	22363.81	8611.861	2105.857	6307.870	1748.824
8				6322.304	4077.123	23108.46	8898.609	2175.975	6517.902	1807.055
9			3753.698	6177.952	3984.033	22580.84	8695.434	2126.293	6369.085	1765.796
10		10312.89	4124.198	6787.732	4377.267	24809.63	9553.696	2336.164	6997.729	1940.084

The total outstanding claims reserve from these predictions is 299264.

Standard Questions

5. *An actuary is using a Poisson model to analyse the run-off triangle from Question 1.*

   (a) *Show that the following values of γ_j are within a 0.05 likelihood interval under the Poisson model. (That is, show that the likelihood for these parameter values is at least 0.05 times the maximum likelihood for the data.)*

j	γ_j
2019	*0.235*
2020	*0.365*
2021	*0.20*
2022	*0.15*
2023	*0.05*

The log-likelihood is

$$l(\mu,\gamma) = \sum_{i+j\leqslant I} X_{ij}\log(\mu_i\gamma_j) - \mu_i\gamma_j$$

Setting the derivative with respect to μ_i to zero for all μ_i, we get

$$\frac{\sum_{j\leqslant I-i} X_{ij}}{\mu_i} = \sum_{j\leqslant I-i}\gamma_j$$

for all i. This gives us

$$\mu_i = \frac{\sum_{j\leqslant I-i} X_{ij}}{\beta_{I-i}}$$

Substituting the values gives

$$\mu_1 = \frac{32760}{1} = 32760$$
$$\mu_2 = \frac{30290}{0.950} = 31884.2105263$$
$$\mu_3 = \frac{31595}{0.800} = 39493.75$$
$$\mu_4 = \frac{25335}{0.600} = 42225$$
$$\mu_5 = \frac{11739}{0.235} = 49953.1914894$$

This gives the following Poisson means for the observed data:

Accident	Earned	Development year			
year	0	1	2	3	4
2019	7698.600	11957.40	6552.000	4914.000	1638
2020	7492.789	11637.74	6376.842	4782.632	
2021	9281.031	14415.22	7898.750		
2022	9922.875	15412.12			
2023	11739.000				

This gives the log likelihood:

$$l(\mu, \gamma) = 1076757$$

For the MLE estimates obtained using the chain ladder method, the Poisson means are:

Accident	Earned	Development year			
year	0	1	2	3	4
2019	7616.694	11976.95	6597.239	4900.121	1669
2020	7420.464	11668.38	6427.274	4773.879	
2021	9188.292	14448.22	7958.487		
2022	9848.550	15486.45			
2023	11739.000				

so the log-likelihood is

$$l(\mu, \gamma) = 1076760$$

Thus, the log-likelihood ratio between the given values of μ_i and the MLEs is $1076760 - 1076757 = 2.703546$, so the relative likelihood is $e^{-2.703546} = 0.0669676240175$ so the MLEs given are in a 5% relative maximum likelihood interval.

(b) For the values in the above table, what is the probability that outstanding claims exceed \$65,000,000?

Under the Poisson model, the outstanding claims follow a Poisson distribution, with mean

$$\sum_{i=1}^{I} \mu_i - \sum_{i+j \leqslant I} X_{ij} = 196316.2 - 131719 = 64597.15$$

We can approximate this by a normal distribution with mean 64597.15 and variance 64597.15. The probability of this exceeding 70000 is

$$1 - \Phi\left(\frac{65000 - 64597.15}{\sqrt{64597.15}}\right) = 1 - \Phi\left(1.58502708997\right) = 0.05648009$$