Universiteit van Amsterdam
Amsterdam School of Economics

Computer class NLIST—Assignment 4

Solving IBNR problems using R

First read the last subsection of Ch. 10.3, and the slides on De Vijlder's method.

In this computer exercise, we elaborate on the paper De Vylder (1978), in which the three ways that time operates were first recognized. You'll find out about the following:

- problems when there are 'impossible' data and how to cope
- how to handle IBNR problems when the data are not triangular
- pre-computer methods: successive substitution in action
- adding a fixed inflation to a chain ladder model has no effect (multicollinearity, 'dummy-trap')
- gamma or Hoerl curves (Section 2), see e.g. here, being a parsimonious intermediate form between geometric and arbitrary parameters claimed to be often 'close to reality'
- a model where the reporting delay is geometric, except for the first period
- generating an IBNR triangle from a given stochastic model
- having R help doing analysis of *scaled* deviance
- getting estimates of $\mu\alpha_i$ directly by leaving out the intercept in a model with `i` as a factor.
- in the arithmetic separation method (see Sec. 10.3.2), to estimate the parameters is easy using `glm`,
- but to compute forecasts is much more complicated than it is with chain ladder
- a three-way model involving all directions in which time operates.

1 De Vijlder's least squares method

To read the data of De Vylder (1978) and to construct the covariates, do:

```
rm(list=ls(all=TRUE))  ## Discard old garbage
Xij <- scan(n=60)
     0      0      0     0     0  4627
     0      0      0     0 15140 13343
     0      0      0 43465 19018 12476
     0      0 116531 42390 23505 14371
     0 346807 118035 43784 12750 12284
```

```
308580 407117 132247 37086 27744      0
358211 426329 157415 68219     0      0
327996 436744 147154     0     0      0
377369 561699      0     0     0      0
333827      0      0     0     0      0
```

Note that as opposed to the theory slides and Table 10.5, zeros are included for development years 1–6 and calendar years prior to 6 and after 10. An advantage of this is that it is somewhat easier, simply invoking the `fitted()` function, to compute fitted values for the (i, j) combinations that are either not recorded (top-left) or as yet unobserved (bottom-right). Also, this gives an easy pattern for the covariates `i` and `j`. A disadvantage is that we have to eliminate these observations from the fitting process. We'll want to weight the zero observations out, attaching weight 0 to non-valid data.

Q_1 Fill in the dots below.

```
i <- ....       ## the row nrs are (1,1,1,1,1,1,2,2,2,2,2,2,...)
j <- ....       ## the col nrs are (1,2,3,4,5,6,1,2,3,4,5,6,...)
k <- ....       ## the calendar year of the payments
future <- .... ## TRUE for obs with calendar year after now
valid <- ....  ## 1 for the non-zero obs, 0 for zero obs
```

☐

For convenience, we make factors corresponding to all three covariates, letting `i` and `fi` store the row numbers, `j` and `fj` the column numbers and `k` and `fk` the calendar years of the payments in `Xij`:

```
fi <- as.factor(i); fj <- as.factor(j); fk <- as.factor(k)
```

To see if row and column numbers are right, we produce a table with the `Xij` values:

```
xtabs(Xij~i+j)
```

De Vijlder's least squares method requires solving a gaussian GLM with a log-link. See MART (10.12) and (10.13).

Suppose we simply do (using row and column number as covariates):

```
gg <- glm(Xij~fi+fj,gaussian(link=log),weights=valid)
```

Then `R` responds that it cannot find valid starting values, and asks you to please specify some. This problem is caused by the fact that the `glm()` function takes the observed values as starting values. The way the data are read in here, the missing values equal 0, and since `R` applies the link function to the observations to get starting values in the iteration, it produces an error message even if the corresponding weight equals zero.

One might change the zero observations to `NA` by doing `Xij[!valid] <- NA`, or add an argument `subset=Xij>0` in the `glm()` call to include positive observations only. The right parameter estimates result, but since the zero observations are no longer considered to be actual observations, to reconstruct the corresponding fitted values is not as easy.

Several other workarounds to solve this problem of invalid observations exist:

1. The value of the zero-weight observations being irrelevant anyway, change them into any positive number, for example `start <- Xij; start[Xij==0] <- 0.01`, or `start <- pmax(Xij,0.01)`.
2. Find starting values by another GLM that does allow zero observations: `start <- fitted.values(glm(Xij~fi+fj,poisson,weights=valid))`
3. Simply do: `start <- rep(mean(Xij[Xij>0]), length(Xij))`, or `start <- Xij+0.5`.

Now we can fit a loglinear model by least squares as follows:

```
gg <- glm(Xij~fi+fj,gaussian(link=log),weights=valid,mustart=start)
```

Just as in the theory slides, we extract the corresponding `alpha` and `beta` estimates from the vector of parameter estimates; the fitted values are the outer product $\vec{\alpha}\otimes\vec{\beta} = \vec{\alpha}\vec{\beta}' = ((\alpha_i\beta_j))$. We ensure that $\sum \beta_j = 1$.

```
cc <- exp(coef(gg)); round(cc, 3)
alpha <- cc[1] * c(1,cc[2:10]); names(alpha)[1] <- "fi1"
beta <- c(1,cc[11:15]); names(beta)[1] <- "fj1"
alpha <- alpha * sum(beta); beta <- beta / sum(beta)
round(alpha); round(beta, 3)
```

Q_2 Using the starting values `Xij+.5`, check if we have precisely reproduced the values of x_i and p_i of De Vylder (1978), Table 3, p. 253. □

Q_3 As said, there are many possibilities fo find feasible starting values. Compare the number of iterations or the error messages with various possible choices for the starting value:

```
start <- rep(1,length(Xij))
glm(Xij~fi+fj,gaussian(link=log),weights=valid,mustart=start)$iter
start <- rep(10000,length(Xij))
glm(Xij~fi+fj,gaussian(link=log),weights=valid,mustart=start)$iter
start <- rep(100000,length(Xij))
glm(Xij~fi+fj,gaussian(link=log),weights=valid,mustart=start)$iter
start <- rep(mean(Xij),length(Xij))
glm(Xij~fi+fj,gaussian(link=log),weights=valid,mustart=start)$iter
start <- rep(mean(Xij[Xij>0]), length(Xij))
glm(Xij~fi+fj,gaussian(link=log),weights=valid,mustart=start)$iter
start <- fitted.values(glm(Xij~fi+fj,poisson,weights=valid))
glm(Xij~fi+fj,gaussian(link=log),weights=valid,mustart=start)$iter
start <- Xij+0.5
glm(Xij~fi+fj,gaussian(link=log),weights=valid,mustart=start)$iter
start <- Xij; start[Xij==0] <- 0.01
glm(Xij~fi+fj,gaussian(link=log),weights=valid,mustart=start)$iter
start <- pmax(Xij, 0.01)
glm(Xij~fi+fj,gaussian(link=log),weights=valid,mustart=start)$iter
```
□

Now add the calendar year as a *variate* to the formula in the `glm` call above, and inspect the resulting coefficients. This leads to a model with fixed inflation $\alpha_i\beta_j\gamma^k$ rather than just $\alpha_i\beta_j$. In the theory slides it is demonstrated why this is it not an improvement.

```
gg <- glm(Xij~fi+fj,gaussian(link=log),weights=valid,mustart=start)
ggg <- glm(Xij~fi+fj+k,gaussian(link=log),weights=valid,mustart=fitted(gg))
round(exp(coef(gg)),3); round(exp(coef(ggg)),3)
gg$iter; ggg$iter
(gg$deviance - ggg$deviance)/ggg$deviance
```

Q_4 Compare the coefficients estimated in gg and in ggg. Also comment on the number of iterations done and the deviances reached. □

Q_5 Using R and (10.12), verify if the results in De Vylder (1978) were computed and printed correctly. Compare his Table 2 with the future fitted values found by:

```
xtabs(round(fitted(gg))*future~i+j)[6:10,2:6]
```
□

Q_6 De Vijlder, in the pre-PC era, in fact used the method of successive substitution to find estimates of the coefficients, see Section 9.3. Verify that setting zero the derivatives of (10.13) (without γ's) leads to:

$$\alpha_i = \sum_j w_{ij} x_{ij} \beta_j \Big/ \sum_j w_{ij} \beta_j^2; \qquad \beta_j = \text{similar.}$$

To implement successive substitution for the case, fill in the dots in the following script, like in Sec. 9.3. The result should be the same as that from the previous question.

```
beta <- rep(1, 6)
repeat
{ beta.old <- beta
  alpha <- tapply(....,i,sum)/tapply(....,i,sum)
  beta <- tapply(....,j,sum)/tapply(....,j,sum)
  if (sum(abs((beta.old-beta)/beta)) < 1e-7) break  ## out of the loop
  cat(beta,"\n")  ## to monitor the iteration process
}
round(xtabs(alpha[i]*beta[j]*future~i+j)[6:10,2:6])
```
□

2 Using Hoerl growth curves

One way to reduce the number of parameters to be estimated for an IBNR problem is to replace, e.g., the arbitrary developments factors $\beta_1, \dots, \beta_{\mathrm{TT}}$, with $\beta_1 = 1$, by a geometric series $\beta^0, \dots, \beta^{\mathrm{TT}-1}$. This way only one parameter β has to be estimated.

Adding one parameter for the first coefficient separately may sometimes help to get a much better fit. Often, the first development year is different from later years, notably when claims are either filed directly or after a delay with a distribution resembling an exponential one. This leads to parameters $\beta_1, \beta_2 = \beta^1, \beta_3 = \beta^2, \dots$ for two real parameters β_1 and β.

Quite often, the parameters have a growth pattern resembling a so-called Hoerl curve, or gamma curve, which has $\beta_j = \exp(\gamma j + \delta \log(j))$ for some real parameters $\gamma \leq 0$ and δ. These can be used for all rows in common, or for each row separately (interaction). This pattern is also known as 'exponential decay'. The Hoerl-curve $\beta(x) = \exp(\gamma x + \delta \log(x)) = x^\delta \, \mathrm{e}^{-|\gamma| x}$, $x \geq 0$ is proportional to a gamma-density. For negative δ, it is infinite at $x \downarrow 0$ and decreasing, for positive δ it increases to a maximum and decreases after that.

Q_7 At which x does the Hoerl-curve function $\beta(x) = \exp(\gamma x + \delta \log(x))$ have a maximum?

For given d, for which γ and δ does $\beta(x+1)/\beta(x) \to d$ hold in the limit for $x \to \infty$? The limit d is called the (asymptotic) decay factor. □

Now we are going to generate a run-off triangle of payments by a (quasi-)Poisson(μ, ϕ) model with fixed ϕ and means $\mu \alpha_i \beta_j$.

Q_8 Fill in the dots in the following script to fill `beta[1:TT]` with the β's following a Hoerl pattern in which the $\beta(x)$ function has its top at $x = 2$ and decay $d = .5$. Normalize the coefficients to ensure that `beta[1]==1` holds.

```
rm(list=ls(all=TRUE)) ## Discard old garbage
TT <- 10; x.top <- 2; d <- .5
gamma <- ....; delta <- ....
beta <- ....
```

□

We will generate a dataset following a Hoerl pattern, but to make things more interesting, we allow `beta` to deviate by at most 4%.

```
beta <- beta * runif(TT,.96,1.04)
plot(beta)
```

The `alpha` coefficients exhibit a 3% growth with some fixed deviations:

```
alpha <- 1.03^(1:TT) * c(1,1,1,1.05,.95,1.05,.95,1.05,.95,1)
```

To ensure that the mean of the top-left cell equals 1000, we do

```
alpha <- 1000 * alpha / alpha[1] / beta[1]
```

We compute row and column numbers as follows:

```
i <- rep(1:TT,TT:1); j <- sequence(TT:1); fi <- as.factor(i); fj <- as.factor(j)
```

Then the theoretical means for each cell are given in

```
mu.ij <- alpha[i] * beta[j]
```

To draw from (quasi-)Poisson($\mu_t, \phi = 2$) r.v. with μ_t given in `mu.ij`, do

```
phi <- 2; Xij <- phi * rpois(length(mu.ij), mu.ij/phi)
```

To display the results, do:

```
xtabs(round(mu.ij)~i+j)
round(xtabs(Xij~i+j))
```

Q_9 Why are all `Xij` values even numbers? □

Now we have a triangle `Xij` that is suitable for our IBNR methods. First we estimate a chain ladder model, as follows:

```
CL <- glm(Xij~fi+fj-1, quasipoisson)
exp(coef(CL))
```

Note that by dropping the constant from the model (enforced by including "`-1`" in the formula), we get a direct estimate for each $\mu\alpha_1, \dots, \mu\alpha_{\mathrm{TT}}$ instead of the exponent of an intercept and coefficients that are actually equal to $\alpha_2/\alpha_1, \dots, \alpha_{\mathrm{TT}}/\alpha_1$.

Requiring the β_j parameters to follow a Hoerl pattern means that $\beta_h = \exp(\gamma h + \delta \log(h)), h = 1, 2, \dots$ with parameters γ and δ. To estimate those parameters, we only need to include a term `j` as well as `log(j)` in the linear predictor. Here `j` stores the row numbers as a variate, `fj` being the factor. So the Hoerl model can be estimated by

```
Hoerl <- glm(Xij~fi+I(j-1)+log(j)-1, quasipoisson)
```

Note that in order to be able to compare the coefficients with those of CL, we dropped the constant again, adding a term `-1`. Also, for the second covariate, instead of `j` we prefer to use `j-1` as a covariate, consisting of numbers `0:7` rather than `1:8`, so the first column has contribution 0 in the linear predictor. We cannot just use `j-1` as a term in the model formula, because the minus sign there means removing a term, not subtraction. This problem is solved by using the 'as is' function `I()` (see `?I`). This also makes it possible to use, for example, a covariate `X1*X2+X3^2-1` without actually constructing it first.

Now compare the coefficients in both models:

```
round(coef(CL),3); round(coef(Hoerl),3)
beta.CL <- exp(c(0,coef(CL)[(TT+1):(2*TT-1)]))
beta.Hoerl <- exp(coef(Hoerl)[TT+1]*(0:(TT-1))) * (1:TT)^coef(Hoerl)[TT+2]
round(rbind(beta.CL, beta.Hoerl), 4)
plot(beta.CL); points(beta.Hoerl, col="red")
```

Note how closely the Hoerl `beta`'s approximate the arbitrary (CL) ones. Whether the restriction to a Hoerl pattern presents a significant loss in scaled deviance is answered by the following. First we use the fullest available model to estimate the scale parameter ϕ. With it, compute differences in scaled deviance and corresponding df.

```
scale <- CL$deviance/CL$df.residual
Delta.Dev.Sc <- (Hoerl$deviance - CL$deviance)/scale
Delta.df <- Hoerl$df.residual - CL$df.residual
reject <- Delta.Dev.Sc > qchisq(0.95, Delta.df)
```

```
cat("The Hoerl model", ifelse(reject, "is", "is not"), "rejected",
    "since the scaled deviance gained by CL is", round(Delta.Dev.Sc,1),
    "\nwith", Delta.df, "extra parameters.\n")
```

Q_{10} Even though the coefficients look alike, the difference in scaled deviance may well be significant. To see this, run the lines from the one with "`rpois()`" above to the end of the "`cat()`" call a few times. □

Q_{11} Verify that in this way, the scaled deviance is correctly analyzed. □

Q_{12} Consider the incremental data of exer. 10.2.1, to be read as:

```
Xij <- c(232,106,35,16,2, 258,115,56,27, 221,82,4, 359,71, 349)
  i <- c(  1,  1, 1, 1,1,   2,  2, 2, 2,   3, 3,3,   4, 4,   5)
  j <- c(  1,  2, 3, 4,5,   1,  2, 3, 4,   1, 2,3,   1, 2,   1)
```

Doing an analysis of deviance, find out if the Hoerl curve is a good fit for the β's, compared to CL. Does a Hoerl curve describe the portfolio growth (with `i`)? □

3 Separation methods; calendar year

In this section, we see that in the arithmetic separation method (see Sec. 10.3.2), to estimate the parameters is easy using `glm`, but to compute forecasts is much more complicated than it is with chain ladder. We also study a three-way model involving all directions in which time operates.

Read the data (claim numbers) and construct the covariates:

```
rm(list=ls(all=TRUE)) ## Discard old garbage
Xij <- scan(n=36)
156  37   6   5   3   2   1   0
154  42   8   5   6   3   0
178  63  14   5   3   1
198  56  13  11   2
206  49   9   5
250  85  28
252  44
221
TT <- trunc(sqrt(2*length(Xij)))
i <- rep(1:TT, TT:1); j <- sequence(TT:1); k <- i+j-1
fi <- as.factor(i); fj <- as.factor(j); fk <- as.factor(k)
```

Now we first estimate the CL model, then the 3-way model:

```
CL <- glm(Xij~fi+fj, poisson)
Threeway <- glm(Xij~fi+fj+fk, poisson)
anova(CL, Threeway)
round(qchisq(0.95, c(21,15,6)),1)
```

From the analysis of deviance (scaled equals unscaled here) we see:

1. the pure Poisson CL model has a deviance 34.158, which is larger than the critical value 32.7 at 95% level, so it is rejected
2. the 3-way model is not ($21.252 < 25.0$)
3. the CL model is rejected in favor of the 3-way model ($12.9 > 12.6$)

As we have seen before, it is easy to expand the fitted values in case of a CL GLM to a square.

Q_{13} Extract `alpha` and `beta` from the vector of coefficients of `CL`, and construct and print the square of fitted values. Explain the last column. □

Q_{14} Now using an appropriate call of `glm`, apply the Arithmetic Separation method (10.9), with Poisson errors, and column and diagonal numbers as covariates.

Store the result as follows:

```
AS <- ....
```

Why is the exponent of the intercept equal to the top left observation? □

To generate fitted values for the lower triangle of the IBNR-data is not as easy as with CL. This is because values of the inflation levels for the future calendar years are not available. We can store the coefficients corresponding to the calendar years in an array `gamma.AS`, and then call `plot(log(gamma.AS))`:

```
cc <- exp(coef(AS))
beta.AS <- c(1,cc[2:8])*cc[1]; gamma.AS <- c(1,cc[9:15])
par(mfrow=c(1,2)); plot(gamma.AS); plot(log(gamma.AS))
```

In this case there is not a clearly visible trend, neither on linear scale nor on loglinear (geometric) scale. We are going to extrapolate the coefficients in the linear predictor log-linearly anyway. First we fit the regression line through the coefficients:

```
ab <- coef(lm(log(gamma.AS)~I(1:8)))
```

`ab[1]` now is the intercept, `ab[2]` the slope of the regression line. The fitted value for the first coefficient is `exp(ab[1]+1*ab[2])`, for the second, `exp(ab[1]+2*ab[2])`, and so on until the fifteenth.

Q_{15} Construct the fitted and extrapolated values, and then generate fitted values for the full IBNR-square. Use the actual coefficients for calendar years `1:8`, not the fits.

```
gamma.extrapolated <- ....
gammas <- gamma.extrapolated; gammas[1:8] <- ....
jjj <- rep(1:8,8); kkk <- jjj + rep(1:8,each=8) - 1
mm <- ....  ## vector with predictions beta[jjj] * gamma[kkk]
round(matrix(mm,8,byrow=TRUE),3)
```

□

Q_{16} As regards AIC, which model AS, CL or Threeway fits better for these data? □

4 Forecasting IBNR-claims

In this section, you analyze your own personal 10×10 IBNR triangle. Start by running the following script:

```
rm(list=ls(all=TRUE)) ## Discard old garbage
set.seed(birthday)  ## replace by your birthday in format yymmdd
top <- 1+1.5*runif(1); decay <- .5 + runif(1)/5
gamma <- log(decay); delta <- -top*gamma
beta <- exp(gamma*(0:(10-1)) + delta*log(1:10))
alpha <- 1.03^(1:10) * (.80+runif(10)/5)
alpha <- 100 * alpha / alpha[1] / beta[1]
i <- rep(1:10,10:1); j <- sequence(10:1)
fi <- as.factor(i); fj <- as.factor(j)
phi <- 1.1+runif(1)/3
Xij <- round(phi * rpois(55, alpha[i] * beta[j]/phi))
rm(phi,alpha,beta,gamma,delta,top,decay)
Xij <- pmax(Xij,1)
xtabs(Xij~i+j)
anova(glm(Xij ~ i+j+log(i)+log(j)+fi+fj, quasipoisson))  ## for Q16
rbind(1:10,round(qchisq(.95,1:10),1))
##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
## [1,]  1.0    2  3.0  4.0  5.0  6.0  7.0  8.0  9.0  10.0
## [2,]  3.8    6  7.8  9.5 11.1 12.6 14.1 15.5 16.9  18.3
```

Q_{17} The final model in the chain considered by the `anova()` call uses as covariates the year of origin and the development year, as factors. Therefore it is the CL method. Through an analysis of deviance, find out if replacing one or two of these factors by variates or Hoerl patterns gives a 'better' model.

Which estimate $\widehat{\phi}$ of the dispersion parameter ϕ do you use? ☐

Q_{18} Describe the three components of the GLM of your 'best' model; see Sec. 9.2. ☐

Q_{19} As in Sec. 10.6, using only the vectors `alpha, beta, i, j`, compute the total of the retrofitted values $\alpha_i\beta_j$ with $i+j$ 'past', and also the total of retrofitted values and estimated future values $\alpha_i\beta_j$ with $i+j$ 'past or future'. Argue why their difference estimates the total reserve to be kept by the CL method. ☐

5 Zero adjusted geometric delays

Suppose that in year i, the number of claims in total is Poisson(α_i) distributed. Further suppose that a claim on a contract of year i is filed in calendar year $i+T$, that is, after a random delay $T \in \{0, 1, \dots\}$. The delay random variable resembles a geometric(p) distribution as it has $\Pr[T = t] \propto (1-p)^t$ when $t > 0$, but the probability of $T = 0$ is arbitrary rather than

p, so a constant decay as of year 1. Such random variables are called 'zero adjusted' geometric random variables. Note that this adjustment may increase the probability of $T = 0$, or decrease it. A geometric decay of the delay probabilities looks plausible; it resembles the situation in Hoerl curves at the end. But in the first year, the probability may be larger when some claims are handled instantly, or smaller when in the first year, there is less time for a claim to mature.

Q_{20} A probabilistic setting in which such probabilities occur is when there is an exponential(λ) random variable U determining when the claim is filed, and $T = 0$ if $U \leq t_0$ but $T = \lceil U - t_0 \rceil$ if $U > t_0$.

Show that then, $\Pr[T = t+1]/\Pr[T = t] = \Pr[T = 2]/\Pr[T = 1]$ for all $t = 1, 2, \ldots$

Given $\Pr[T = 0] = p_0$ and $\Pr[T = 2]/\Pr[T = 1] = 1 - p$, find λ and t_0.

Show that $p_0 > p$ iff $t_0 > 1$.

Note that $\Pr[T = t]$ decreases for $t = 1, 2, \ldots$; given λ, for which t_0 is $\Pr[T = 0] = \Pr[T = 1]$?

For which t_0 is $\Pr[T = t]$ maximal when $t = 0$, for which when $t = 1$? $\square$

Q_{21} Consider a CL-model with Poisson claim numbers with expected value $\alpha_i\beta_j$ with the β_j following the model above.

Express β_j, $j = 1, 2, \ldots$ in p, p_0. Express p and p_0 in terms of β_1/β_2.

Which GLM specification would enable you to estimate α_i, p_0 and p?

Fit such a model on the triangle of the previous section.

Which model fits better as to AIC, the Hoerl one or the zero adjusted geometric model? $\square$