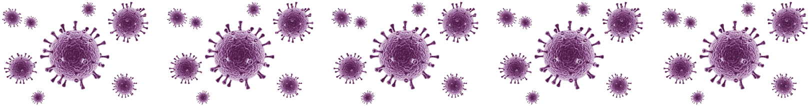
Modèles Linéaires Appliqués / Régression
GLM & Résultats Non-Asymptotiques

Arthur Charpentier

UQAM

Hiver 2020 - COVID-19 # 19



With ChainLadder package, cumulated payments

```
> library(ChainLadder)
> data(GenIns)
> round(GenIns/1000)

      dev
orig    1    2    3    4    5    6    7    8    9   10
   1  358 1125 1735 2218 2746 3320 3466 3606 3834 3901
   2  352 1236 2170 3353 3799 4120 4648 4914 5339   NA
   3  291 1292 2219 3235 3986 4133 4629 4909   NA   NA
   4  311 1419 2195 3757 4030 4382 4588   NA   NA   NA
   5  443 1136 2128 2898 3403 3873   NA   NA   NA   NA
   6  396 1333 2181 2986 3692   NA   NA   NA   NA   NA
   7  441 1288 2420 3483   NA   NA   NA   NA   NA   NA
   8  359 1421 2864   NA   NA   NA   NA   NA   NA   NA
   9  377 1363   NA   NA   NA   NA   NA   NA   NA   NA
  10  344   NA   NA   NA   NA   NA   NA   NA   NA   NA
```

Or incremental payments (row = accident year)

```
> Y=cum2incr(GenIns)

      dev
origin   1    2    3    4   5   6   7   8   9  10
     1 358  767  611  483 527 574 146 140 227  68
     2 352  884  934 1183 446 321 528 266 425  NA
     3 291 1002  926 1017 751 147 496 280  NA  NA
     4 311 1108  776 1562 272 352 206  NA  NA  NA
     5 443  693  992  769 505 471  NA  NA  NA  NA
     6 396  937  847  805 706  NA  NA  NA  NA  NA
     7 441  848 1131 1063  NA  NA  NA  NA  NA  NA
     8 359 1062 1443   NA  NA  NA  NA  NA  NA  NA
     9 377  987   NA   NA  NA  NA  NA  NA  NA  NA
    10 344   NA   NA   NA  NA  NA  NA  NA  NA  NA
```

GLM à Distance Finie

Classical model, $Y_{i,j} \sim \mathcal{P}(L_i C_j)$, multiplicatif ligne-colonne

```
> base = data.frame(Y=as.vector(Y),origin=1:10,dev=rep
    (1:10,each=10))
> reg = glm(Y ~ as.factor(origin)+as.factor(dev),data=
    base,family=poisson)
> matrix(predict(reg,newdata=base,type="response")
    ,10,10)
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] ,10]
[1,]   270  673  704  753  417  293  268  182  273   68
[2,]   376  937  981 1049  581  407  374  254  380   95
[3,]   372  927  971 1039  575  403  370  251  376   94
[4,]   367  913  957 1023  567  397  364  247  370   92
[5,]   336  838  877  938  520  364  334  227  339   85
[6,]   354  881  923  987  547  383  352  238  357   89
[7,]   392  976 1022 1093  606  425  389  264  396   99
[8,]   470 1170 1225 1310  726  509  467  317  474  118
[9,]   391  973 1019 1090  604  423  388  263  394   98
10,]   344  857  897  960  532  373  342  232  347   87
```

GLM à Distance Finie

```
> passe = which(!is.na(base$Y))
> futur = which(is.na(base$Y))
> sum(predict(reg,newdata=base,"response")[futur])
[1] 18680
```

On peut récupérer les résidus (de Pearson) sur la partie observée

```
> epsilon = residuals(reg,type="pearson")
```

Pour rappel

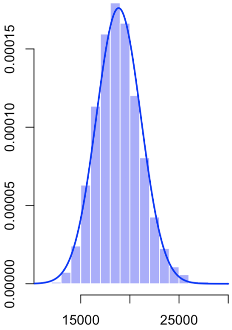
$$\widehat{\varepsilon}_{i,j} = \frac{y_{i,j} - \widehat{\mu}_{i,j}}{\sqrt{\widehat{\mu}_{i,j}}} \text{ ou } y_{i,j} = \widehat{\mu}_{i,j} + \sqrt{\widehat{\mu}_{i,j}}\, \widehat{\varepsilon}_{i,j}$$

On peut alors créer des pseudo-observations par boostrap

$$y_{i,j}^{(b)} = \widehat{\mu}_{i,j} + \sqrt{\widehat{\mu}_{i,j}}\, \widehat{\varepsilon}_{i,j}^{(b)}$$

GLM à Distance Finie

```
> R=rep(NA,9999)
> for(b in 1:9999){
L = predict(reg,type="response")
n = length(L)
bases = base
epsilon = residuals(reg,type="pearson")
bases$Ys = NA
bases$Ys[passe] = L + sqrt(L)*epsilon[sample(1:n,size=n,replace=TRUE)]
bases$Ys[bases$Ys<0]=0
regs = glm(Ys ~ as.factor(origin)+as.factor(dev),data=bases,family=poisson)
R[b] = sum(predict(regs,newdata=base,"response")[futur]) }
```



On visualise la distribution de

$$\widehat{R} = \sum_{(i,j)\text{ futur}} \widehat{\mu}_{i,j}$$