M1 EOS - modèles probit et logit

Hugo Harari-Kermadec

M1 EOS - Econométrie

#Logit et Probit—

```
setwd("C:/Users/p101679/Dropbox/enseignement/Saclay M1 EOS econometrie/replication")
Sys.setlocale("LC_MESSAGES","fr_FR.UTF-8")
```

```
## Warning in Sys.setlocale("LC_MESSAGES", "fr_FR.UTF-8"): LC_MESSAGES existe sous
## Windows mais n'y est pas opérationel
```

```
## Warning in Sys.setlocale("LC_MESSAGES", "fr_FR.UTF-8"): La requête OS pour
## spécifier la localisation à "fr_FR.UTF-8" n'a pas pu être honorée
```

```
## [1] ""
```

```
library(tidyverse);library(stargazer)
etab_2019 <-read_delim("C:/Users/p101679/Dropbox/enseignement/Saclay M1 EOS econometrie/replication/db_
```

```
## Rows: 85 Columns: 23
## -- Column specification --------------------------------------------------------
## Delimiter: ";"
## chr  (3): ETABLI, Libellé, Sigle
## dbl (20): class_shanghai, effectif, hommes, femmes, bac_S, bac_L, bac_ES, ba...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#etab_2019 <- read_csv("./db_2019.csv",locale = locale(encoding = "ISO-8859-1"))
```

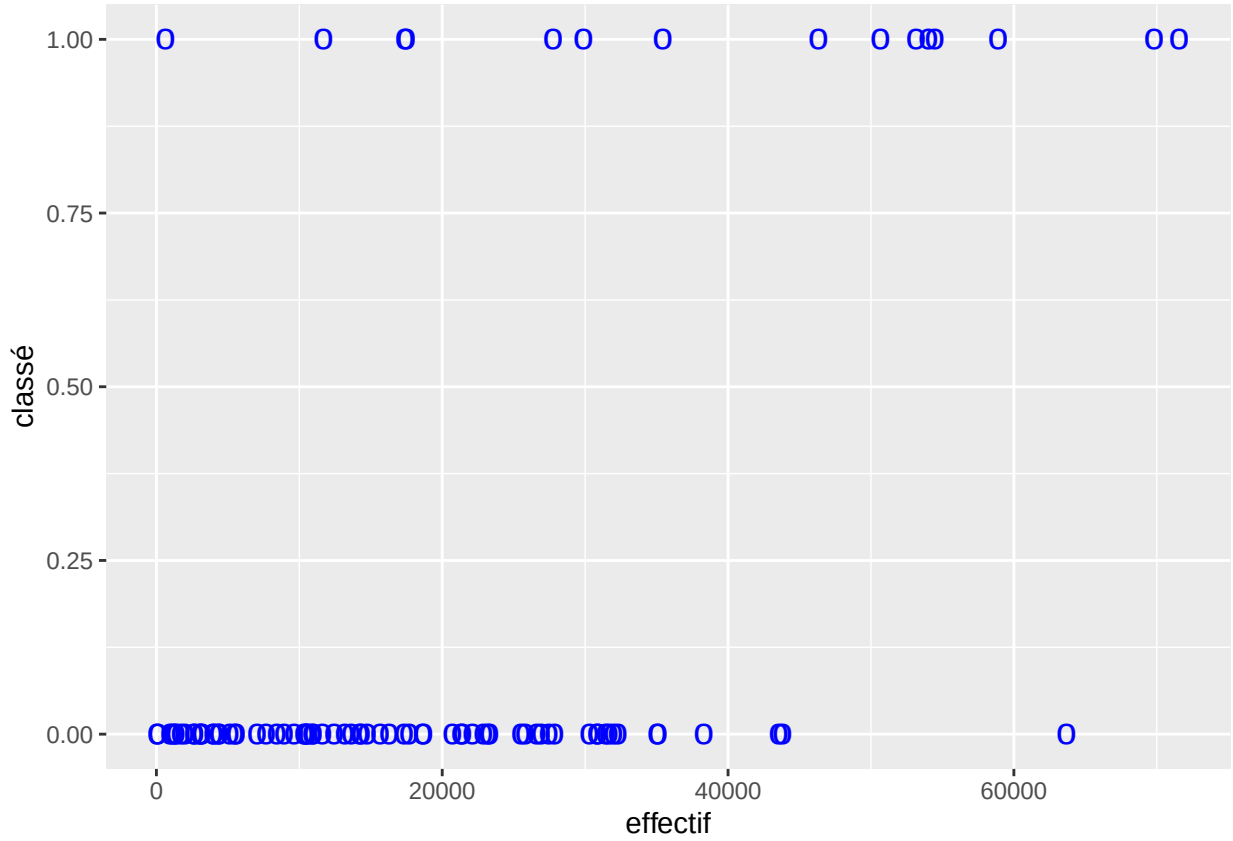
Adaptez le chemin ci-dessus pour qu'il corresponde à votre dossier

#Données qualitatives—

On créé une variable qui dit si une université est classée à Shanghai ou non.

```
db_logit<-etab_2019 %>% mutate(classé=as.numeric(!is.na(class_shanghai)))

ggplot(db_logit,aes(x=effectif,y=classé,size=2))+
  geom_point(col='blue',pch='o')+
  theme( legend.position="none")
```



On le voit, un modèle linéaire ne permettra pas de bien se rapprocher des points, puisqu'ils sont sur deux lignes parallèles !

Par contre, on peut quand, de façon grossière, savoir quelles variables ont un effet sur la probabilité d'être une fac d'élite.

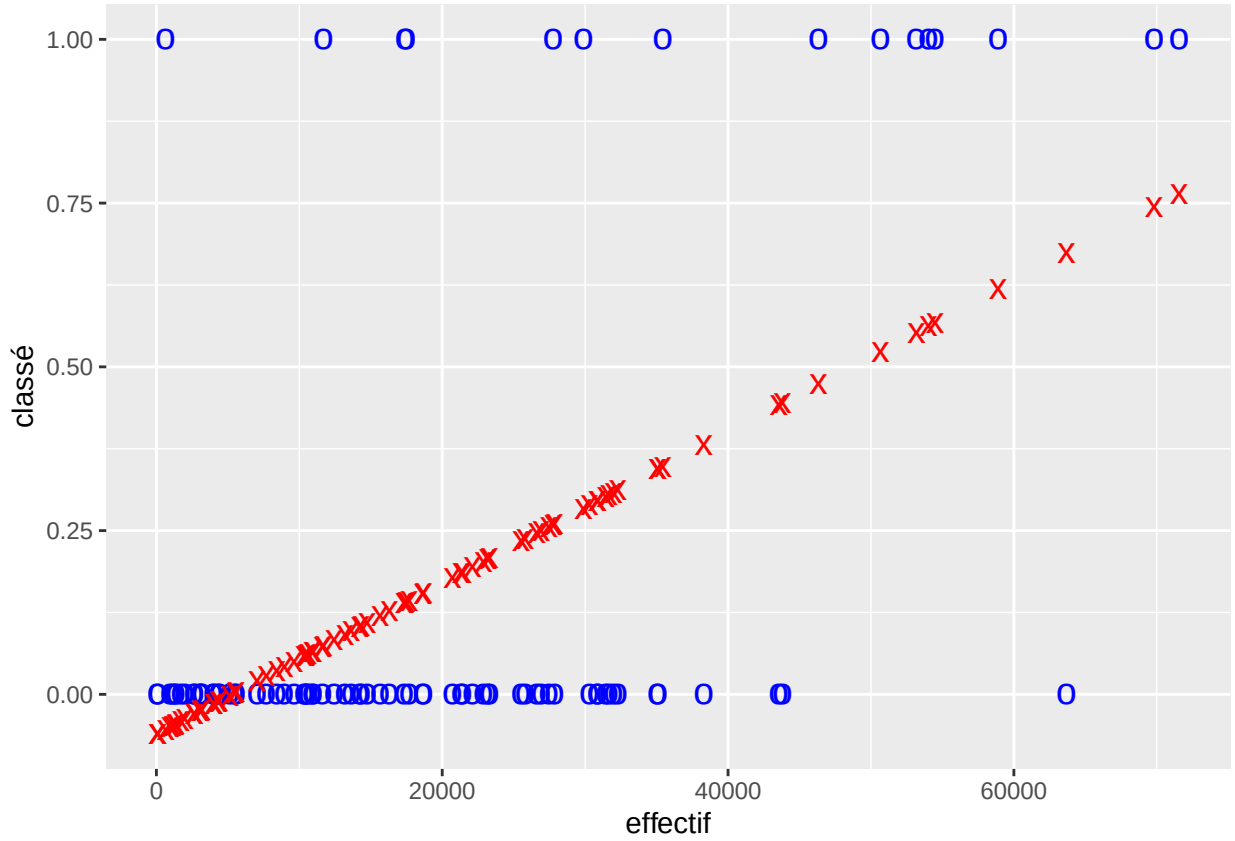
```
lm1<-lm(data=db_logit,classé ~ effectif)
summary(lm1)
```

```
##
## Call:
## lm(formula = classé ~ effectif, data = db_logit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.67311 -0.20239 -0.05961  0.04178  1.05406
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.155e-02  5.583e-02  -1.103    0.273
## effectif     1.154e-05  2.079e-06   5.550 3.34e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3295 on 83 degrees of freedom
## Multiple R-squared:  0.2706, Adjusted R-squared:  0.2618
## F-statistic:  30.8 on 1 and 83 DF,  p-value: 3.344e-07
```

```
stargazer(lm1,type="text")
```

```
##
## ================================================
##                          Dependent variable:
##                      ---------------------------
##                                 classé
## ------------------------------------------------
## effectif                      0.00001***
##                                (0.00000)
##
## Constant                        -0.062
##                                 (0.056)
##
## ------------------------------------------------
## Observations                      85
## R2                               0.271
## Adjusted R2                      0.262
## Residual Std. Error         0.329 (df = 83)
## F Statistic             30.798*** (df = 1; 83)
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

```
db_logit<-db_logit %>% mutate(fit_lm1=predict(lm1))
ggplot(db_logit,aes(x=effectif,y=classé,size=2))+
  geom_point(col='blue',pch='o')+
  geom_point(aes(x=effectif,y=fit_lm1),col='red',pch='x')+
  theme( legend.position="none")
```



On voit que l'effect a un effet fort et positif, mais que l'adéquation du modèle aux données ne va pas du tout.

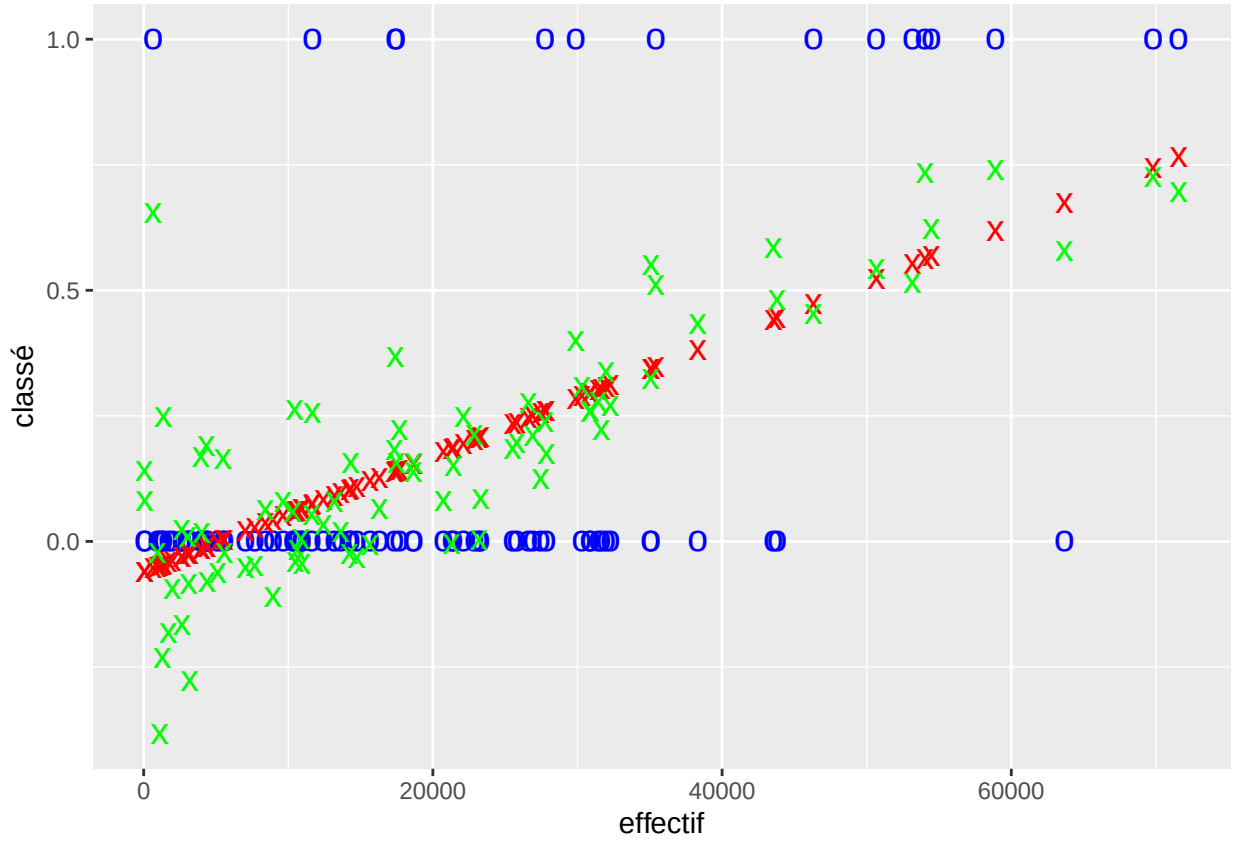
```
lm2<-lm(data=db_logit,classé ~ effectif+bac_S+masters+prof_sup)

stargazer(lm1,lm2,type="text")
```

```
##
## ===============================================================
##                                 Dependent variable:
##                     ------------------------------------------
##                                        classé
##                               (1)                   (2)
## ---------------------------------------------------------------
## effectif                   0.00001***            0.00001***
##                            (0.00000)             (0.00000)
##
## bac_S                                             0.519**
##                                                   (0.251)
##
## masters                                          -0.603***
##                                                   (0.223)
##
## prof_sup                                          1.514***
##                                                   (0.441)
##
## Constant                    -0.062                -0.482***
##                            (0.056)                (0.117)
```

```
##
## ------------------------------------------------------------------
## Observations                    85                      85
## R2                             0.271                   0.401
## Adjusted R2                    0.262                   0.371
## Residual Std. Error     0.329 (df = 83)         0.304 (df = 80)
## F Statistic          30.798*** (df = 1; 83) 13.385*** (df = 4; 80)
## ==================================================================
## Note:                                  *p<0.1; **p<0.05; ***p<0.01
```

```
db_logit<-db_logit %>% mutate(fit_lm2=predict(lm2))
ggplot(db_logit,aes(x=effectif,y=classé,size=2))+
  geom_point(col='blue',pch='o')+
  geom_point(aes(x=effectif,y=fit_lm1),col='red',pch='x')+
    geom_point(aes(x=effectif,y=fit_lm2),col='green',pch='x')+
  theme( legend.position="none")
```



On va utiliser un modèle qui intègre une variable latente : le logit. Au lieu de modèliser Y directement, on modèlise la probabilité que Y soit égale à 1.

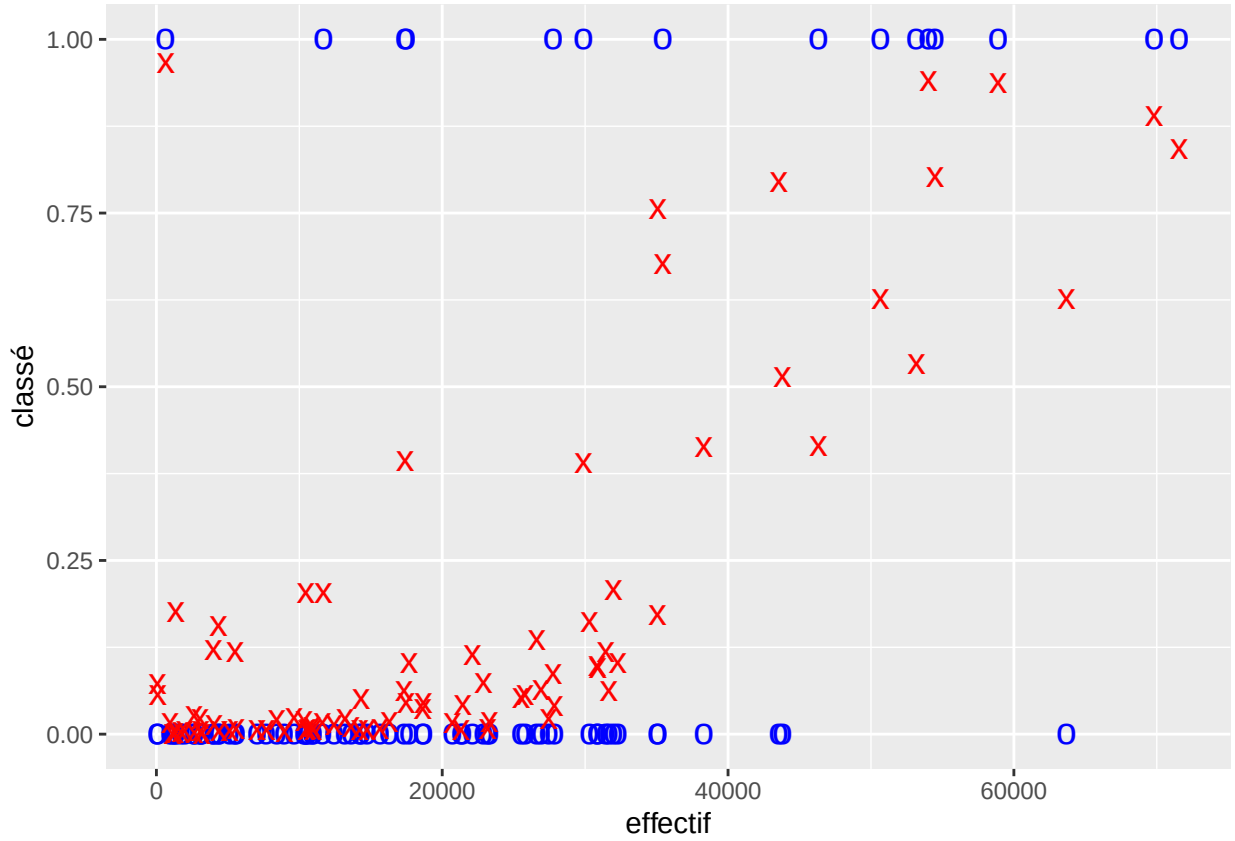
```
logit1<-glm(data=db_logit,classé ~effectif+bac_S+masters+
              prof_sup , family=binomial(link="logit"))
stargazer(logit1,type="text")
```

```
##
## ================================================
##                         Dependent variable:
##                     ---------------------------
```

```
##                              classé
## ---------------------------------------------
## effectif                    0.0001***
##                             (0.00002)
##
## bac_S                        6.276*
##                              (3.585)
##
## masters                      -6.575*
##                              (3.557)
##
## prof_sup                    18.167***
##                              (6.865)
##
## Constant                   -10.262***
##                              (2.644)
##
## ---------------------------------------------
## Observations                   85
## Log Likelihood               -20.638
## Akaike Inf. Crit.            51.276
## =============================================
## Note:             *p<0.1; **p<0.05; ***p<0.01
```

```
db_logit<-db_logit %>% mutate(fit_logit1=logit1$fitted.values)

ggplot(db_logit,aes(x=effectif,y=classé,size=2))+
  geom_point(col='blue',pch='o')+
  geom_point(aes(x=effectif,y=fit_logit1),col='red',pch='x')+
  theme( legend.position="none")
```



On peut aussi modifier la loi probabilité qui sert à modéliser la variable binaire Y, en remplaçant la loi logistique par la gaussienne. C'est alors un Probit.

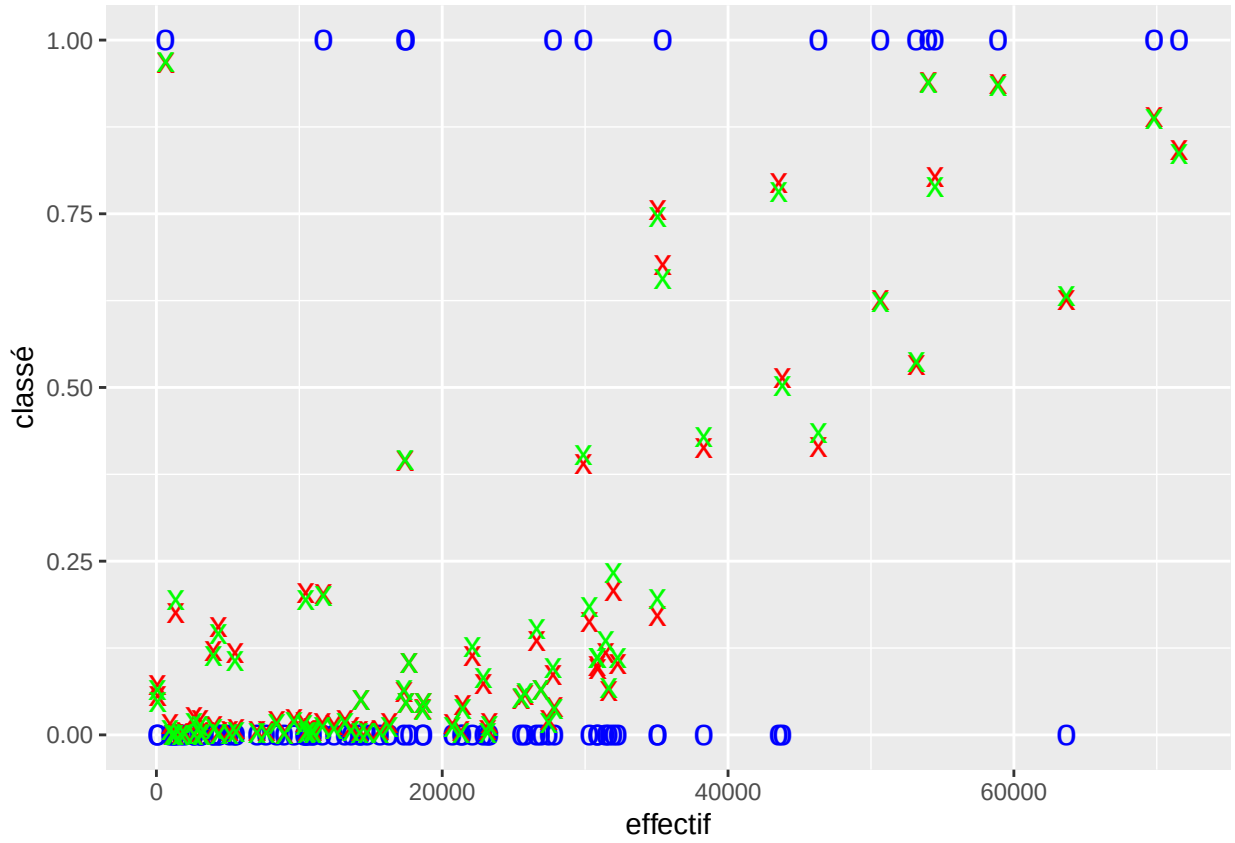
```
probit1<-glm(data=db_logit,classé ~effectif+bac_S+masters+
             prof_sup , family=binomial(link="probit"))
stargazer(logit1,probit1,type="text")
```

```
##
## ================================================
##                         Dependent variable:
##                     ---------------------------
##                                classé
##                           logistic       probit
##                             (1)            (2)
## ------------------------------------------------
## effectif                 0.0001***     0.00005***
##                          (0.00002)      (0.00001)
##
## bac_S                     6.276*         3.594*
##                          (3.585)        (1.940)
##
## masters                  -6.575*        -3.804**
##                          (3.557)        (1.862)
##
## prof_sup                 18.167***      9.950***
##                          (6.865)        (3.503)
##
```

```
## Constant             -10.262***      -5.694***
##                       (2.644)         (1.312)
##
## ------------------------------------------------
## Observations             85              85
## Log Likelihood       -20.638         -20.434
## Akaike Inf. Crit.     51.276          50.868
## ================================================
## Note:               *p<0.1; **p<0.05; ***p<0.01
```

```
db_logit<-db_logit %>% mutate(fit_probit1=probit1$fitted.values)

ggplot(db_logit,aes(x=effectif,y=classé,size=2))+
  geom_point(col='blue',pch='o')+
  geom_point(aes(x=effectif,y=fit_logit1),col='red',pch='x')+
  geom_point(aes(x=effectif,y=fit_probit1),col='green',pch='x')+
    theme( legend.position="none")
```



Tobit —

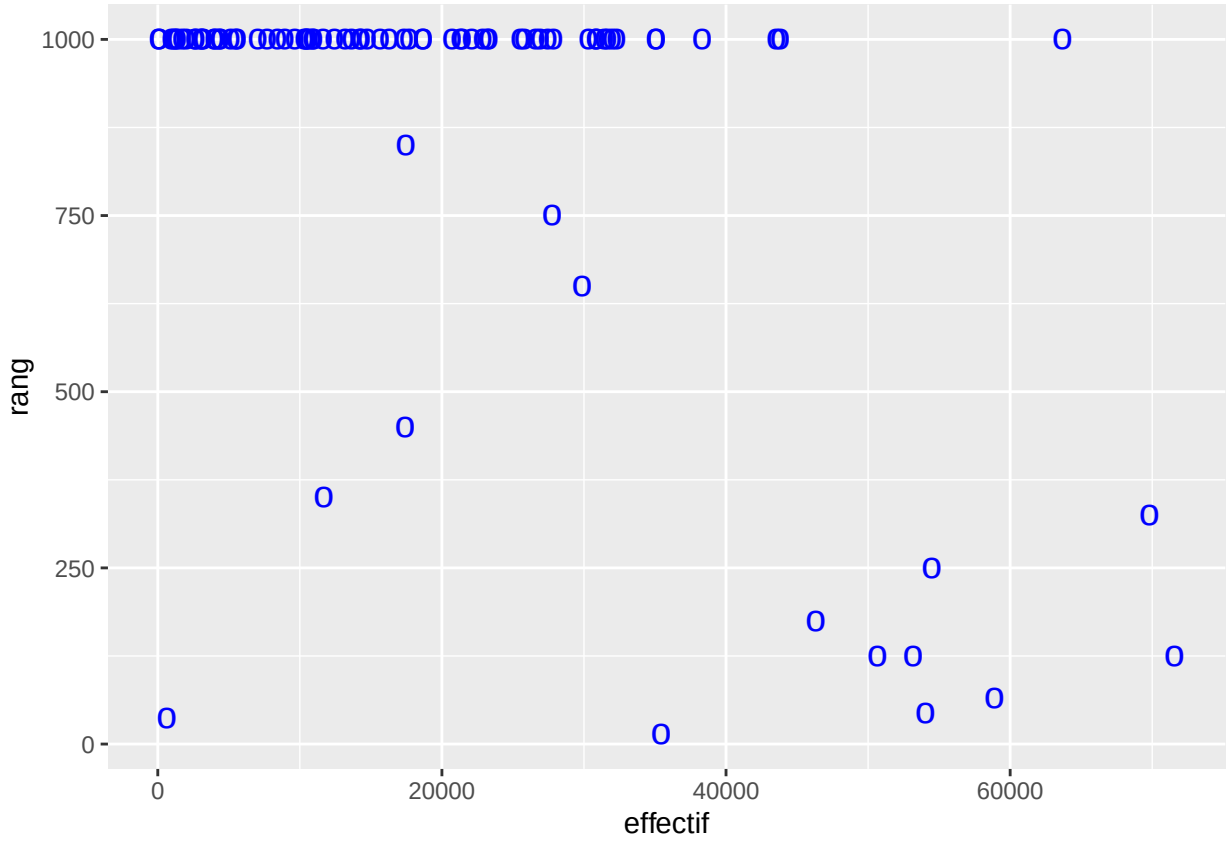
On va essayer de faire mieux, en prenant en compte le rang dans le classement

```
summary(db_logit$class_shanghai)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    14.0    95.0   175.0   288.9   400.0   850.0      70
```

```
db_tobit<-db_logit %>% mutate(rang=case_when(is.na(class_shanghai)~ 1000,T~class_shanghai))

ggplot(db_tobit,aes(x=effectif,y=rang,size=2))+
  geom_point(col='blue',pch='o')+
  theme( legend.position="none")
```



Cette fois-ci, on a une partie de l'information directement, et une autre non, censurée, le classement de Shanghai n'allant pas au-delà de 1 000.

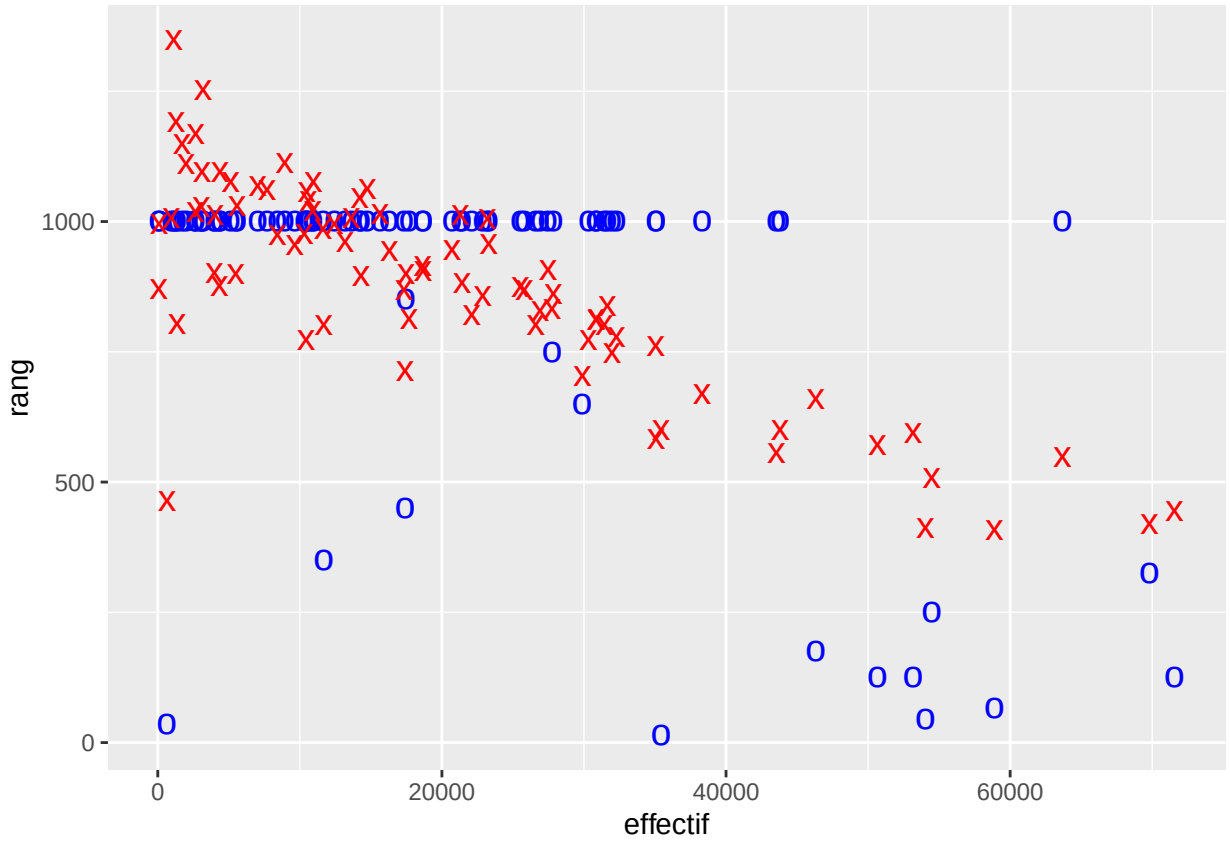
Et si on utilisait un modèle linéaire ?

```
lm3<-lm(data=db_tobit,rang ~ effectif+bac_S+masters+
              prof_sup)
stargazer(lm3,type="text")
```

```
##
## ================================================
##                         Dependent variable:
##                     ---------------------------
##                                rang
## ------------------------------------------------
## effectif                     -0.008***
##                               (0.001)
##
## bac_S                        -345.479*
##                              (181.779)
##
## masters                      504.837***
```

```
##                                (161.275)
##
## prof_sup                     -1,352.284***
##                                (319.279)
##
## Constant                     1,414.877***
##                                (84.854)
##
## ------------------------------------------------
## Observations                      85
## R2                               0.467
## Adjusted R2                      0.440
## Residual Std. Error       220.229 (df = 80)
## F Statistic            17.522*** (df = 4; 80)
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

```
db_tobit<-db_tobit %>% mutate(fit_lm3=lm3$fitted.values)
ggplot(db_tobit,aes(x=effectif,y=rang,size=2))+
  geom_point(col='blue',pch='o')+
  geom_point(aes(x=effectif,y=fit_lm3),col='red',pch='x')+
  theme( legend.position="none")
```



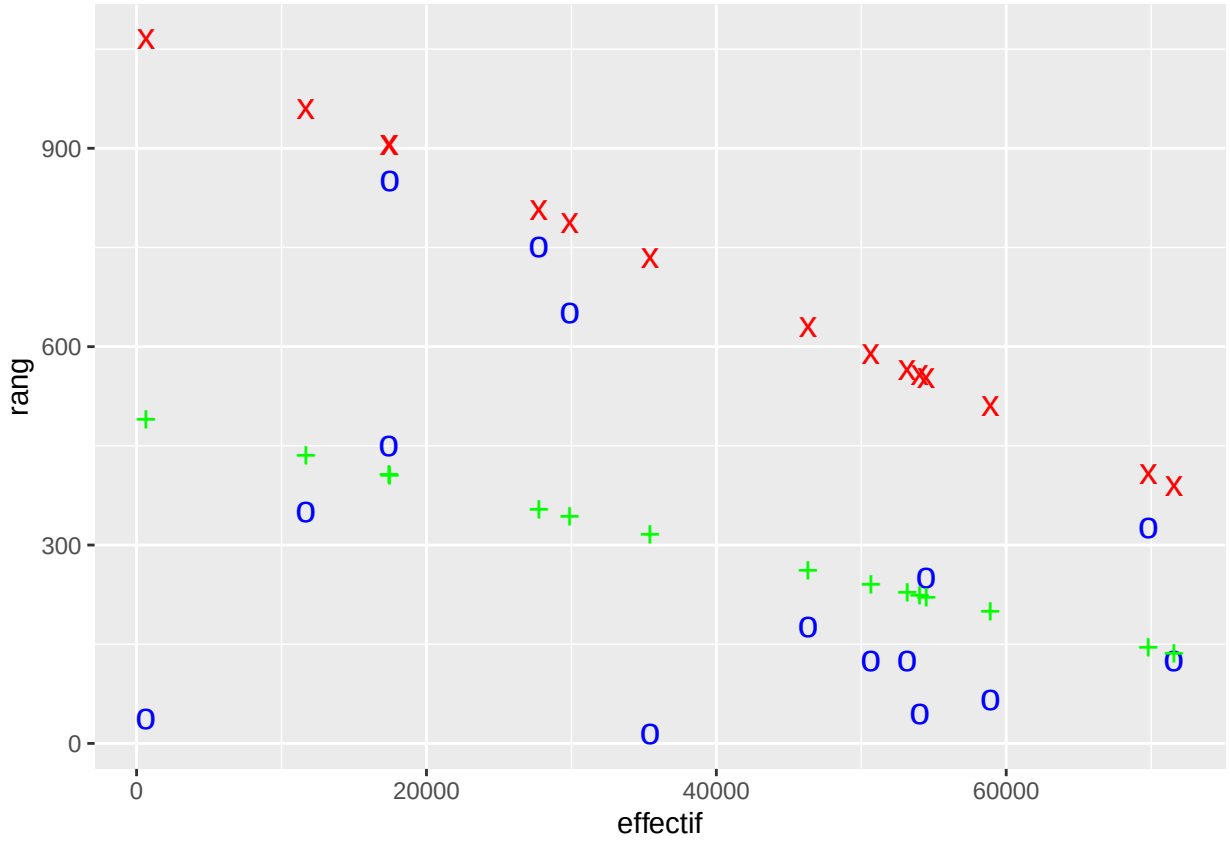
On introduit un biais car l'estimateur des moindres carrés fait comme s'il n'y avait pas de censure. Or, en réalité, beaucoup d'universités ont un rang "latent", inobservable, supérieur à 1000. Donc on sous-observe le rang, et donc l'effet réel de l'effectif est plus fort que ce qu'on estime ici.

Et si on n'utilisait que les données non censurées ?

```
lm4<-lm(data=db_tobit[db_tobit$rang<1000,],rang ~ effectif)
stargazer(lm3,lm4,type="text")
```

```
##
## ===============================================================
##                               Dependent variable:
##                     ----------------------------------------
##                                       rang
##                             (1)                  (2)
## ---------------------------------------------------------------
## effectif                 -0.010***              -0.005
##                           (0.002)               (0.003)
##
## Constant                1,070.920***           488.480***
##                           (41.602)              (143.403)
##
## ---------------------------------------------------------------
## Observations                85                     15
## R2                         0.313                 0.160
## Adjusted R2                0.304                 0.095
## Residual Std. Error   245.519 (df = 83)    258.462 (df = 13)
## F Statistic         37.761*** (df = 1; 83) 2.472 (df = 1; 13)
## ===============================================================
## Note:                               *p<0.1; **p<0.05; ***p<0.01
```

```
db_tobit %>%  filter(rang<1000) %>% mutate(fit_lm4=predict(lm4)) %>%
ggplot(aes(x=effectif,y=rang,size=2))+
  geom_point(col='blue',pch='o')+
    geom_point(aes(x=effectif,y=fit_lm3),col='red',pch='x')+
  geom_point(aes(x=effectif,y=fit_lm4),col='green',pch='+')+
  theme( legend.position="none")
```



On a alors un biais de sélection : on estime notre modèle uniquement sur les "meilleures" universités. C'est peut-être un bon modèle pour cette sous-population particulière, mais pas pour l'ensemble de la population. Mathématiquement, rien ne dit que le lien entre les variables est le même pour les données censurées, puisque la censure (ou réciproquement la sélection) n'est pas aléatoire. Economiquement, le classement de Shanghai regroupe des universités mieux financées par étudiant, donc on sur-estime sans doute l'effet de l'effectif : il y a des très grandes universités non-classées, parce qu'elles ont peu de dépenses de recherche.

Pour ces données mixtes, partiellement censurées, on utilise donc un modèle mixte, à moitié linéaire à moitié logit.

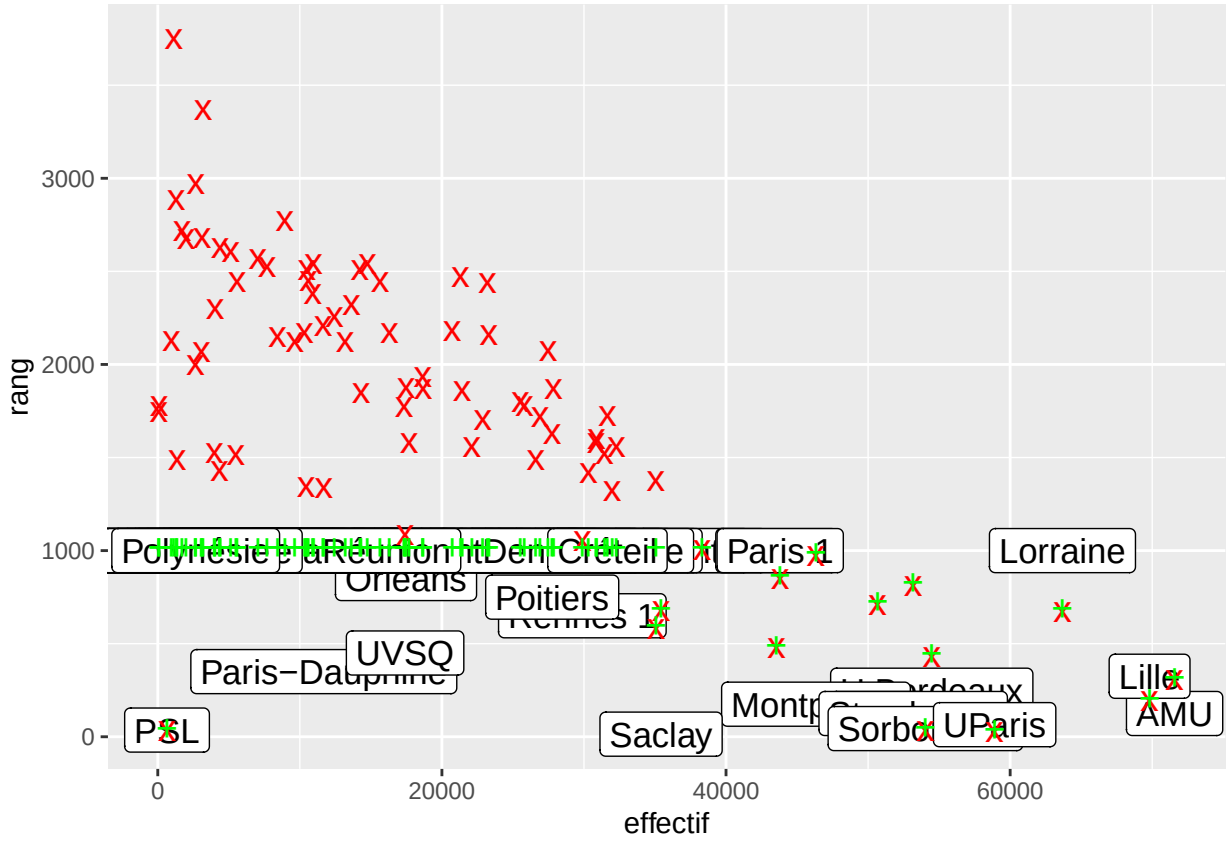
```
#install.packages("AER")
library(AER)
tobit1<-tobit(data=db_tobit,rang ~effectif+bac_S+masters+
              prof_sup, right=1000)
stargazer(lm3,lm4,tobit1,type="text")
```

```
##
## ================================================================================
##                                        Dependent variable:
##                    ------------------------------------------------------------
##                                               rang
##                                  OLS                          Tobit
##                             (1)                  (2)              (3)
## --------------------------------------------------------------------------------
## effectif                  -0.008***             -0.005          -0.029***
##                            (0.001)              (0.003)          (0.006)
##
## bac_S                     -345.479*                            -1,829.863**
```

```
##                            (181.779)                              (908.929)
##
## masters                    504.837***                            1,840.559***
##                            (161.275)                              (695.213)
##
## prof_sup                 -1,352.284***                           -5,442.672***
##                            (319.279)                              (1,392.933)
##
## Constant                  1,414.877***          488.480***        4,084.925***
##                             (84.854)            (143.403)          (704.570)
##
## ------------------------------------------------------------------------------
## Observations                   85                    15                 85
## R2                           0.467                 0.160
## Adjusted R2                  0.440                 0.095
## Log Likelihood                                                      -127.885
## Residual Std. Error   220.229 (df = 80)    258.462 (df = 13)
## F Statistic         17.522*** (df = 4; 80) 2.472 (df = 1; 13)
## Wald Test                                                      27.640*** (df = 4)
## ==============================================================================
## Note:                                              *p<0.1; **p<0.05; ***p<0.01
```

```
db_tobit<-db_tobit %>% mutate(fit_tobit1=predict(tobit1),
                              res_tobit1=residuals(tobit1),
                              cens_fit_tobit1=fit_tobit1*(fit_tobit1<1000)+
                                              1000*(fit_tobit1>=1000))

ggplot(db_tobit,aes(x=effectif,y=rang,size=2))+
  geom_point(col='blue',pch='o')+
  geom_label(aes(label=Sigle))+
  geom_point(aes(x=effectif,y=fit_tobit1),col='red',pch='x')+
  geom_point(aes(x=effectif,y=cens_fit_tobit1),col='green',pch='+')+
      theme( legend.position="none")
```


rang
3000
2000
1000
0
0
20000
40000
60000
effectif
Orleans
Poitiers
UVSQ
Paris−Dauphine
PSL
Saclay
Paris 1
Lorraine
Lille
AMU
UParis