The informational content of geographical indications

Jean-Sauveur AY[*]

INRA UMR CESAER

Reproducibility Material Version 0.1 : June 7, 2019

Abstract

This file contents the R codes associated with the paper "The informational content of geographical indications" AAWE Working Paper No XXX. Data, code and results are under the copyleft licence GNU GPL V3 (licence notices must be preserved). Data are available from the INRA dataverse website: https://data.inra.fr/geoInd. Some R functions are reported in the Appendix to preserve the visibility of codes. The most recent version of this document and a Shiny application are available from the online repository: https://github.com/jsay/geoInd.

Contents

[*] jsay@inra.fr, UMR CESAER, AgroSup, INRA, Univ. Bourgogne Franche-Comté, 26 bd Dr Petitjean, 21000 Dijon (FR).

1 Descriptive Statistics

1.1 Data shaping

The details of data construction are presented (in French) in the data paper: https://github.com/jsay/geoInd/blob/master/DataPaper.pdf which also contains the metadata. The file that results from these preliminary treatments can be downloaded from the INRA dataverse website: https://data.inra.fr/geoInd/GeoRas.Rda. The following code allows to load the file once located in the /Inter/ folder at the root of the working directory of the R session. It loads a SpatialPolygonDataFrame object from the sp package that contains the characteristics of the vineyard plots under consideration (session information is reported at Section XX).

The following code also reshapes some variables of particular interest:

- It reorders the *commune* levels along the North-South gradient
- It computes the distance of each parcel from the *commune* centers
- It standardizes the variable about solar radiation
- It recodes the variable about exposition in 8 quadrants
- It projects the geographical coordinates inside the WGS84 system
- It selects the parcels with GIs and drop omitted values

```
library(sp) ; load("Inter/GeoRas.Rda")
tmp <- unique(Geo.Ras$LIBCOM[order(Geo.Ras$YCHF, decreasing= TRUE)])
Geo.Ras$LIBCOM <- factor(Geo.Ras$LIBCOM, levels= tmp)
Geo.Ras$DISTCHF <- sqrt((Geo.Ras$XL93- Geo.Ras$XCHF* 100)^2
                        + (Geo.Ras$YL93- Geo.Ras$YCHF* 100)^2)
Geo.Ras$RAYAT <- (Geo.Ras$SOLAR- mean(Geo.Ras$SOLAR, na.rm= TRUE))/
    sd(Geo.Ras$SOLAR, na.rm= TRUE)
Geo.Ras$EXPO <- cut(Geo.Ras$ASPECT,
                    breaks= c(-2, 45, 90, 135, 180, 225, 270, 315, 360))
GR84 <- spTransform(Geo.Ras, CRS("+proj=longlat +ellps=WGS84"))
dd <- coordinates(GR84) ; Geo.Ras$X= dd[, 1] ; Geo.Ras$Y= dd[, 2]
dim(Reg.Ras <- subset(Geo.Ras, !is.na(AOClb) & !is.na(DEM) & !is.na(DESCR)
                      & !is.na(RUD) & !is.na(AOC36lab) & !is.na(REGION)))
```

```
[1] 59113    73
```

The resulting object contains 59 113 vineyard plots with 73 variables without omitted values. All the information that is needed to reproduce the econometric results are in the SpatialPolygonDataFrame.

1.2 Geology and pedology

The control of sub-soil and soil characteristics is made with fixed effects that is the more general way knowing the raw spatial resolution of these data. A too small number of observation can be a problem for the estimation, we choose to include a fixed effects only for polygons with more than 1 000 vineyard plots. This recodage is made

```
Reg.Ras$NOTATION <- factor(Reg.Ras$NOTATION)
tmp <- table(Reg.Ras$NOTATION)< 1000
Geo.Ras$GEOL <- factor(
    ifelse(Geo.Ras$NOTATION %in% names(tmp[ tmp]), "0AREF",
           as.character(Geo.Ras$NOTATION)))
Reg.Ras$NOUC <- factor(Reg.Ras$NOUC)
tmp <- table(Reg.Ras$NOUC)< 1000
Geo.Ras$PEDO <- factor(
    ifelse(Geo.Ras$NOUC %in% names(tmp[tmp]), "0AREF",
           as.character(Geo.Ras$NOUC)))
Reg.Ras <- subset(Geo.Ras, !is.na(AOClb) & !is.na(DEM) & !is.na(DESCR)
                  & !is.na(RUD) & !is.na(AOC36lab) & !is.na(REGION))


library(maptools) ; library(rgeos) ; library(RColorBrewer)
tmp_geol <- gBuffer(Geo.Ras, byid= TRUE, width= 0)
geol <- unionSpatialPolygons(tmp_geol, Geo.Ras$GEOL)
pedo <- unionSpatialPolygons(tmp_geol, Geo.Ras$PEDO)

qual_col_pals = brewer.pal.info[brewer.pal.info$category == 'qual',]
col_vector = unlist(mapply(brewer.pal, qual_col_pals$maxcolors,
                           rownames(qual_col_pals)))
mycol <- sample(col_vector, length(unique(Reg.Ras$GEOL))+
                            length(unique(Reg.Ras$PEDO)))

par(mar = c(0, 0, 0, 0))
plot(pi, col= mycol)
par(fig = c(0, 1, 0, 1), oma = c(0, 0, 0, 0), mar = c(0, 0, 0, 0))
plot(0, 0, type = "n", bty = "n", xaxt = "n", yaxt = "n")
legend("bottom", cex= 1.3, legend= substr(unique(pi$DESCR), 1, 60), fill= mycol)

plot(geol, col= mycol)
plot(pedo, col= mycol)
```

1.3 Crossing GIs dimensions

The GIs on the area of interest contains both an horizontal (*communes*) and a vertical (*ranking*) dimension. The balance of the distribution can be assessed with the following Figure, which is Figure 3 (p.36) in the working paper.

```
library(lattice) ; library(RColorBrewer)
fig.dat <- aggregate(model.matrix(~0+ factor(Reg.Ras$AOC))*
                     Reg.Ras$AREA/ 1000, by= list(Reg.Ras$LIBCOM), sum)
names(fig.dat) <- c("LIBCOM", "BGOR", "BOUR", "VILL", "PCRU", "GCRU")
fig.dat$LIBCOM <- factor(fig.dat$LIBCOM, lev= rev(levels(fig.dat$LIBCOM)))
fig.crd <- t(apply(fig.dat[, -1], 1, function(t) cumsum(t)- t/2))
fig.lab <- round(t(apply(fig.dat[, -1], 1, function(t) t/ sum(t)))* 100)
my.pal <- brewer.pal(n= 9, name = "BuPu")[ 2: 8]
barchart(LIBCOM~ BGOR+ BOUR+ VILL+ PCRU+ GCRU, xlim= c(-100, 10200),
         xlab="Vineyards delineated as Geographical Indications (hectare)",
         data= fig.dat, horiz= T, stack= T, col= my.pal, border= "black",
         par.settings= list(superpose.polygon= list(col= my.pal)),
         auto.key= list(space= "top", points= F, rectangles= T, columns= 5,
                        text=c("Coteaux b.", "Bourgogne",
                               "Village", "Premier cru", "Grand cru")),
         panel=function(x, y, ...) {
             panel.grid(h= 0, v = -11, col= "grey60")
             panel.barchart(x, y, ...)
             ltext(fig.crd, y, lab= ifelse(fig.lab> 0, fig.lab, ""))})
```

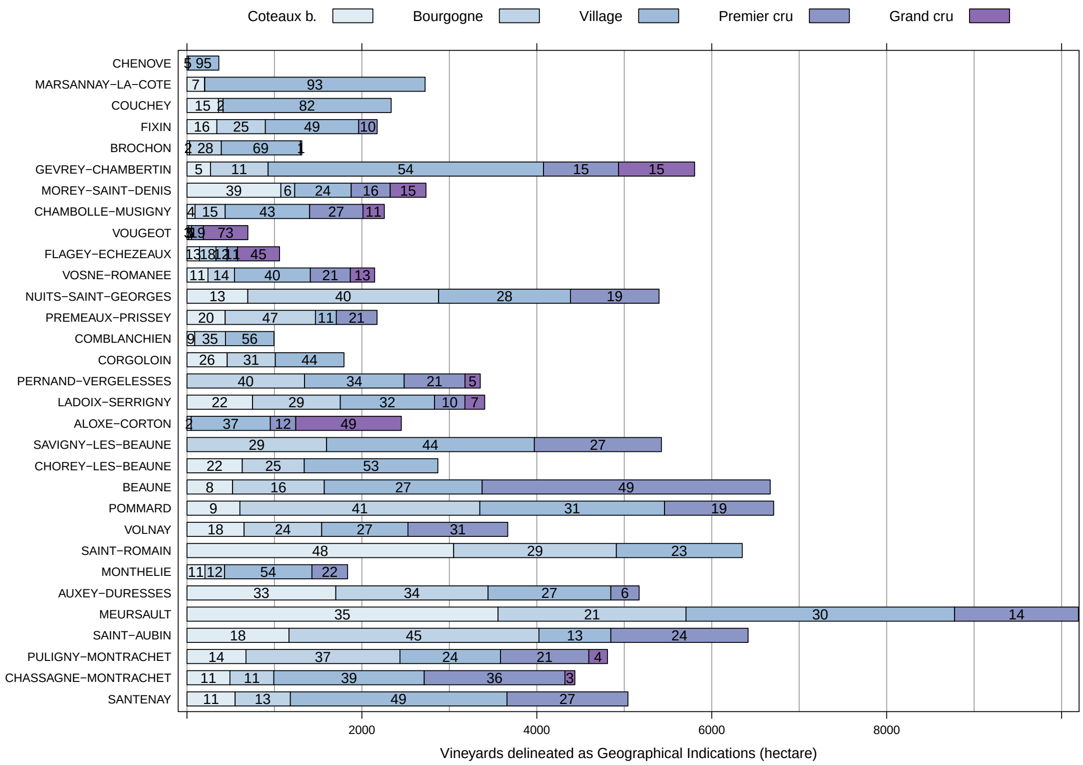


Figure 1: **Cross distribution of GI levels among communes**

2 Models of GI designation

2.1 Parametric ordered logit

We first estimate the benchmark parametric ordered logistic model `polm1` that corresponds to model (0) of Table 1 in the working paper. Model `polm1a` is the auxiliary regression used to test the presence of omitted *terroir* effect. Model `polm1b` is also auxiliary to compute the Fisher statistics associated to spatial smoothing terms in Table 1. We use the standard `polr` function from `MASS` package.

```
library(MASS)
por1 <- polr(factor(AOCc)~ 0+ LIBCOM+ EXPO
             + poly(DEM, 2)+ poly(SLOPE, 2)+ poly(RAYAT, 2)
             + poly(X, 3)* poly(Y, 3), data= Reg.Rank, Hess= TRUE)
por1a <- polr(factor(AOCc)~ 0 + EXPO
              + poly(DEM, 2)+ poly(SLOPE, 2)+ poly(RAYAT, 2)
              + poly(X, 3)* poly(Y, 3), data= Reg.Rank, Hess= TRUE)
por1b <- polr(factor(AOCc)~ 0+ LIBCOM+ EXPO
              + poly(DEM, 2)+ poly(SLOPE, 2)+ poly(RAYAT, 2)
            , data= Reg.Rank, Hess= TRUE)


library(mgcv) ; load("Inter/gamodM.Rda")
## system.time(
##
gam900 <- gam(AOC~ 0+ LIBCOM+ EXPO+ GEOL+ PEDO
##
+ s(DEM)+ s(SLOPE)+ s(RAYAT)+ s(X, Y, k= 900)
##
, data= Reg.Ras, family= ocat(R= 5))
## )
## utilisateur
système
écoulé
##
32271.43
93.78
32366.00
## save(gam900, file= "Inter/gam900.Rda")

qor1 <- polr(factor(AOC)~ 0+ LIBCOM+ EXPO
             + poly(DEM, 2)+ poly(SLOPE, 2)+ poly(RAYAT, 2)
             + poly(X, 3)* poly(Y, 3)+ GEOL+ PEDO, data= Reg.Ras, Hess= T)
qor1a <- polr(factor(AOC)~ 0 + EXPO
              + poly(DEM, 2)+ poly(SLOPE, 2)+ poly(RAYAT, 2)
              + poly(X, 3)* poly(Y, 3)+ GEOL+ PEDO, data= Reg.Ras, Hess= T)
qor1b <- polr(factor(AOC)~ 0+ LIBCOM+ EXPO
              + poly(DEM, 2)+ poly(SLOPE, 2)+ poly(RAYAT, 2)+ GEOL+ PEDO
            , data= Reg.Ras, Hess= TRUE)
```

```
Warning messages:
1: In polr(factor(AOCc) ~ 0 + LIBCOM + EXPO + poly(DEM, 2) + poly(SLOPE,  :
  une coordonnée à l'origine est nécessaire et assumée
2: In polr(factor(AOCc) ~ 0 + LIBCOM + EXPO + poly(DEM, 2) + poly(SLOPE,  :
  le plan ne semble pas de rang plein, des coefs seront ignorés
```

The warning messages are due to the lack of intercept that we force to compute the ordinal superiority measures for each *communes* below. This has no impact on the quality of the ML estimators.

2.2 Ordered generalized additive

The following code presents 2 loops that allow to estimate the OGAM models of GIs designations. Models (I) to (V) reported in Table XX are only a subset of all models estimated here. The `gamod` object contents the full models, the gammod object contents the auxiliary regression to test the omitted *terroir* effects. Because of the complexity of the models, each loop needs about 2 days to run (Dell Precision 7520, 64Go of RAM). I advice the reader to not run the loop entirely but pick some value of listk for the maximum degree of freedom and run the models individually. The objects gamod.Rda and gammod.Rda are available from the git repo mentioned in the first page.

```
library(mgcv)
listk <- c(50, 100, 200, 300, 400, 500, 600, 700, 800, 900)
gamod <- vector("list", length(listk))
system.time(
for (i in 1: length(listk)){
    gamod[[ i]] <- gam(AOCc~ 0+ LIBCOM+ EXPO+ s(DEM)+ s(SLOPE)+ s(RAYAT)
                       + s(X, Y, k= listk[ i])
                     , data= Reg.Rank, family= ocat(R= 5))
})
names(gamod) <- paste0("gam", listk)
save(gamod, file= "Inter/gamod.Rda")

gammod <- vector("list", length(listk))
system.time(
for (i in 1: length(listk)){
    gammod[[ i]] <- gam(AOCc~ 0+ EXPO+ s(DEM)+ s(SLOPE)+ s(RAYAT)
                        + s(X, Y, k= listk[ i])
                      , data= Reg.Rank, family= ocat(R= 5))
})
names(gammod) <- paste0("gam", listk)
save(gammod, file= "Inter/gammod.Rda")

library(mgcv)
listk <- c(50, 100, 200, 300, 400, 500, 600, 700)
gamodM <- vector("list", length(listk))
system.time(
for (i in 1: length(listk)){
    gamodM[[ i]] <- gam(AOC~ 0+ LIBCOM+ EXPO+ s(DEM)+ s(SLOPE)+ s(RAYAT)
                       + s(X, Y, k= listk[ i])+ GEOL+ PEDO
                     , data= Reg.Ras, family= ocat(R= 5))
})
names(gamodM) <- paste0("gam", listk)
save(gamodM, file= "Inter/gamodM.Rda")
## Timing stopped at: 2.645e+05 1109 2.656e+05

gammodM <- vector("list", length(listk))
system.time(
    for (i in 1: length(listk)){
        print(i)
        gammodM[[ i]] <- gam(AOC~ 0+ EXPO+ s(DEM)+ s(SLOPE)+ s(RAYAT)
                             + s(X, Y, k= listk[ i])+ GEOL+ PEDO
                           , data= Reg.Ras, family= ocat(R= 5))
    }
)
names(gammodM) <- paste0("gam", listk)
save(gammodM, file= "Inter/gammodM.Rda")

## utilisateur     système      écoulé
##     47749.1       235.7     47988.9
```

```
utilisateur     système      écoulé
    56177.4       384.9       56565
utilisateur     système      écoulé
    42413.2       262.8     42679.6
```

3 Diagnostics

3.1 Significance

We first reports the Chi-square statistics for the joint significance of the model (0) of Table XX in the working paper.

```
library(car)
res1a <- anova(por1, por1b)
(res1 <- Anova(por1))

qes1a <- anova(qor1, qor1b)
(qes1 <- Anova(qor1))
```

```
Analysis of Deviance Table (Type II tests)

Response: factor(AOCc)
                      LR Chisq Df Pr(>Chisq)
LIBCOM                   14609 31     <2e-16 ***
EXPO                      1209  7     <2e-16 ***
poly(DEM, 2)              5308  2     <2e-16 ***
poly(SLOPE, 2)             400  2     <2e-16 ***
poly(RAYAT, 2)            1934  2     <2e-16 ***
poly(X, 3)                2484  3     <2e-16 ***
poly(Y, 3)                 647  3     <2e-16 ***
poly(X, 3):poly(Y, 3)     9526  9     <2e-16 ***
---
codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Then, we compute the same statistics for the OGAMs, also reported in Table XX in the main paper.

```
load("Inter/gamod.Rda") ; load("Inter/gamodM.Rda")
resume <- function(mod){
    tmp <- anova(mod)
    res <- c(as.vector(rbind(tmp$s.table[, 3], tmp$s.table[, 1])),
             as.vector(rbind(tmp$pTerms.tab[, 2], tmp$pTerms.tab[, 1])))
    names(res) <- c(as.vector(rbind(rownames(tmp$s.table), rep("", 4))),
                    as.vector(rbind(rownames(tmp$pTerms.tab), rep("", 2))))
    round(res, 1)
}
sapply(gamod[ 1: 5* 2], resume)
sapply(gamodM[ c(1, 1: 4* 2)], resume)
```

```
          gam100  gam300  gam500   gam700   gam900
s(DEM)     5020.2  2385.4  1677.7   1692.6   1766.8
              9.0     8.9     8.8      8.8      8.8
s(SLOPE)   1281.1   458.2   266.1    225.3    243.6
              8.5     8.5     8.5      8.4      8.4
s(RAYAT)   2491.6  1196.5   667.3    554.7    557.9
              8.3     8.2     7.7      7.6      7.5
s(X,Y)    41458.2 73705.5 94094.8 103941.0 107522.8
             98.7   295.2   483.1    666.7    844.7
LIBCOM     6793.2  6079.7  4594.7   3555.0   2894.5
             31.0    31.0    31.0     31.0     31.0
EXPO        110.3   123.2   222.3    153.5    160.8
              7.0     7.0     7.0      7.0      7.0
```

3.2 Goodness of fit

Here are the goodness-of-fit measures for model (0) also reported in Table XX: McFadden R^2, Akaike information criteria, and percent of good predictions.

```
psR2 <- function(x) 1- (logLik(x)/ logLik(update(x, . ~ + 1)))
round(c(McFaddenR2= psR2(por1), AIC= AIC(por1)/ 1000,
        Pcgp= sum(diag(table(predict(por1), Reg.Rank$AOCc)))/nrow(Reg.Rank)), 2)

round(c(McFaddenR2= psR2(qor1), AIC= AIC(qor1)/ 1000,
        Pcgp= sum(diag(table(predict(qor1), Reg.Ras$AOC)))/nrow(Reg.Ras)), 2)
```

```
McFaddenR2        AIC       Pcgp
      0.29     119.40       0.59
```

The same goodness of fit measures for OGAMs.

```
pcgp <- function(x){
    sum(diag(table(cut(x$line, c(-Inf, x$family$getTheta(TRUE), Inf)),
                  x$model[, 1])))/ nrow(x$model)* 100
}
rbind(Pcgp= sapply(gamod[ 1: 5* 2], pcgp), AIC= sapply(gamod[ 1: 5* 2], AIC))
sapply(gamod, psR2)

rbind(Pcgp= sapply(gamodM[ 1: 4* 2], pcgp), AIC= sapply(gamodM[ 1: 4* 2], AIC))
sapply(gamodM, psR2)
```

```
       gam100   gam300   gam500   gam700   gam900
Pcgp     73.89    79.94    84.23    86.94    89.15
AIC   82412.10 64710.89 54941.54 48291.33 43535.14
```

3.3 Omitted variable

Bootstrapped statistics for the Fisher about omitted *terroir* variables, with 100 replications for parametric ordered logistic. The absence of correlated effects is strongly rejected. We use the `sure` package for surrogate residual.

```
library(lmtest) ; library(sandwich) ; library(sure)
wal1 <- rep(NA, times= nsim <- 100)
for (i in 1: nsim){
    tmp <- surrogate(por1a)- por1a$lp
    wal1[ i] <- waldtest(lm(tmp~ Reg.Rank$LIBCOM), . ~ 1, vcov= vcovHC)$F[ 2]
}
quantile(wal1, c(.05, .5, .95))

xal1 <- rep(NA, times= nsim)
for (i in 1: nsim){
    tmp <- surrogate(qor1a)- qor1a$lp
    xal1[ i] <- waldtest(lm(tmp~ Reg.Ras$LIBCOM), . ~ 1, vcov= vcovHC)$F[ 2]
}
quantile(xal1, c(.05, .5, .95))
```

```
   5%    50%    95% 
268.0 274.2 279.6 
```

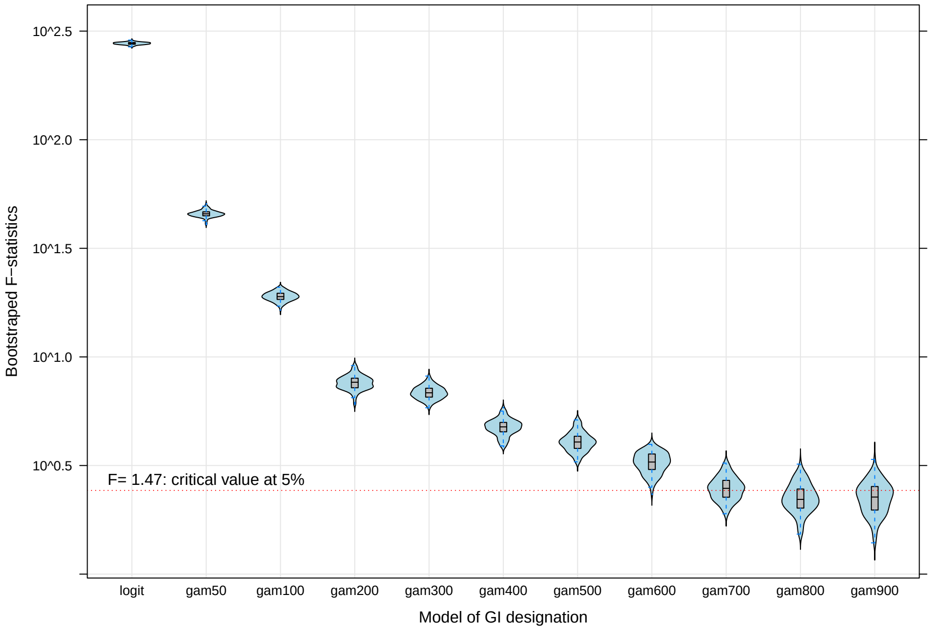
A passer en Reg.Rank, introduire la fonction sur les surrogate residuals des modèles gams en in the Appendix. Not exactly the same because of bootstrap.

```
load("Inter/gammod.Rda") ; load("Inter/gammodM.Rda") ; source("myFcts.R")
omitVar <- function(mod, var, nsim= 100){
    usq <- rep(NA, nsim)
    for(i in 1: nsim) {
        RES <- sureOGAM(mod)
        tmp <- lm(I(RES- mod$linear.pred)~ factor(var))
        usq[ i] <- waldtest(tmp, . ~ 1, vcov= vcovHC)$F[ 2]
    }
    usq
}
wal2 <- sapply(gammod, function(x) omitVar(x, RRank$LIBCOM, nsim= 100))
apply(wal2[, 1: 5* 2], 2, function(x) quantile(x, c(.05, .5, .95)))

xal2 <- sapply(gammodM, function(x) omitVar(x, Reg.Ras$LIBCOM, 100))
apply(xal2[, 1: 4* 2], 2, function(x) quantile(x, c(.05, .5, .95)))
```

```
    gam100 gam300 gam500 gam700 gam900
5%   17.38  6.060  3.377  2.004  1.704
50%  18.94  6.806  4.130  2.525  2.181
95%  20.15  7.746  4.864  3.060  2.760
```

The following plot resumes the specification diagnostics and shows the relevance of OGAMs to control for omitted spatial effects. It corresponds to Figure XX in the working paper, the bootstrapped nature of the statistics individual values change.

```
library(lattice)
pltdat <- stack(data.frame(logit= wal1, wal2))
pltdat <- stack(data.frame(logit= xal1, xal2))

bwplot(values~ ind, data= pltdat, type=c("l","g"), horizontal= FALSE,
       xlab='Model of GI designation', ylab='Bootstraped F-statistics',
       par.settings = list(box.rectangle=list(col='black'),
                           plot.symbol = list(pch='.', cex = 0.1)),
       scales=list(y= list(log= TRUE)),
       panel = function(..., box.ratio) {
           panel.grid(h= -1, v = -11)
           panel.violin(..., col = "lightblue",
                        varwidth = FALSE, box.ratio = box.ratio)
           panel.bwplot(..., col='black',
                        cex=0.8, pch='|', fill='gray', box.ratio = .1)
           panel.abline(h= log(1.47), col= "red", lty= 3)
           panel.text(2, log(1.55), "F= 1.47: critical value at 5%")})
```



3.4 Specification

Surrogate residuals can also be used to test specification, results not reported.

```
library(sure) ; library(ggplot2) ; library(gridExtra)
var <- c("DEM", "SLOPE", "RAYAT", "EXPO", "LIBCOM", "X", "Y")
plots <- lapply(var, function(.x)
    autoplot(por1, what= "covariate", x= Reg.Rank@data[, .x], xlab= .x))
do.call(grid.arrange, c(list(autoplot(por1, what= "qq")), plots))

plots <- lapply(var, function(.x)
    autoplot(qor1, what= "covariate", x= Reg.Ras@data[, .x], xlab= .x))
```

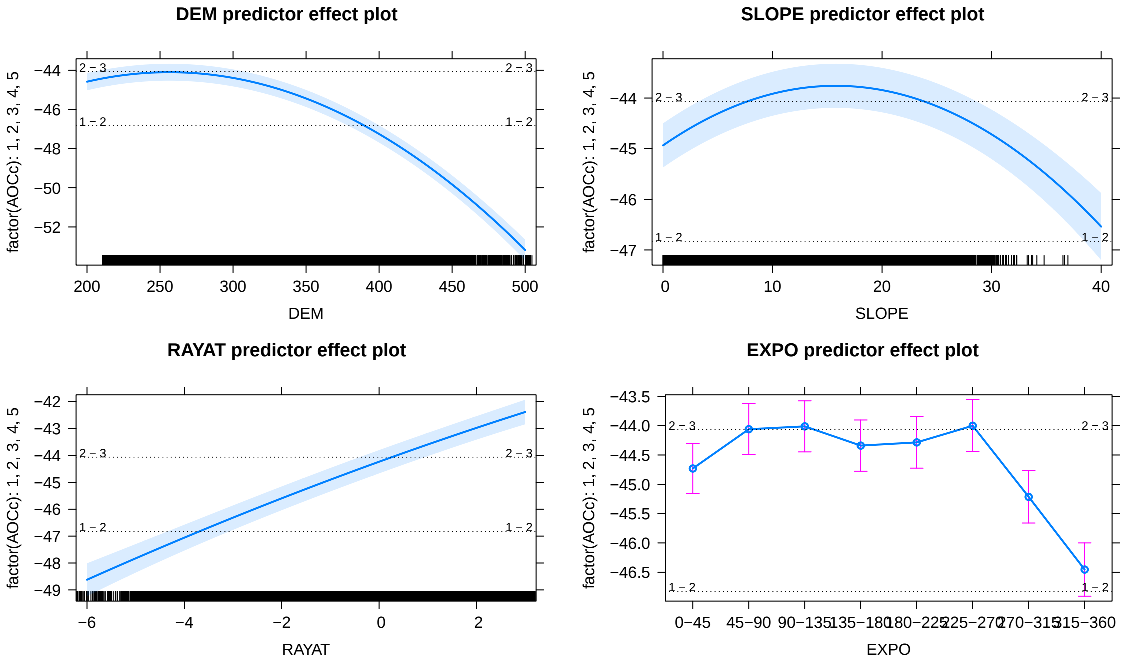
```
do.call(grid.arrange, c(list(autoplot(qor1, what= "qq")), plots))


restmp <- sureOGAM(gamod$gam900)- gamod$gam900$line
plot(qlogis(1: nrow(RRank)/ nrow(RRank), scale= 1), sort(restmp))
abline(0, 1)
pltSURE <- function(resid, xvar, lab){
    plot(xvar, resid, xlab= lab, main= paste("Surrogate Analysis", lab))
    abline(h= 0, col= "red", lty= 3, lwd= 2)
    lines(smooth.spline(resid ~ xvar), lwd= 3, col= "blue")
}
par(mfrow= c(3, 3)) ; for (i in var) pltSURE(restmp, RRank@data[, i], i)

restmp <- sureOGAM(gamodM$gam700)- gamodM$gam700$line
plot(qlogis(1: nrow(Reg.Ras)/ nrow(Reg.Ras), scale= 1), sort(restmp))
abline(0, 1)
par(mfrow= c(3, 3)) ; for (i in var) pltSURE(restmp, Reg.Ras@data[, i], i)
```

4 Marginal effects

4.1 Parametric ordered logit

Marginal effects from parametric models, corresponds to the dotted lines in Figure XX of the working paper.

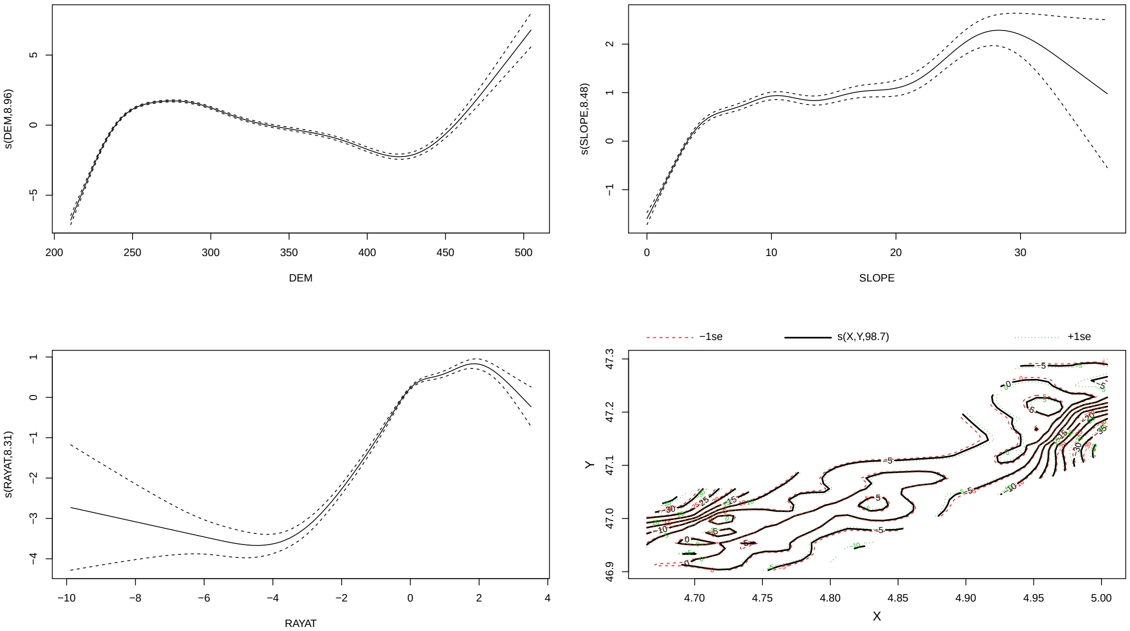
```
library(effects)
plot(predictorEffects(por1, ~ DEM+ SLOPE+ RAYAT+ EXPO, latent= TRUE,
                      xlevels=list(DEM= 200: 500,
                                   SLOPE= 0: 400/ 10, RAYAT= -60: 30/ 10)))

plot(predictorEffects(qor1, ~ DEM+ SLOPE+ RAYAT+ EXPO, latent= TRUE,
                      xlevels=list(DEM= 200: 500,
                                   SLOPE= 0: 400/ 10, RAYAT= -60: 30/ 10)))
```



4.2 Ordered generalized additive

On voit bien que le lissage est le même que le papier. Can be changed by indexing the list `gamod`, below is the reported effect for a maximum effective degrees of freedom of 100. For all models of `gamod`, we obtain the grey curves of Figure XX of the working paper.

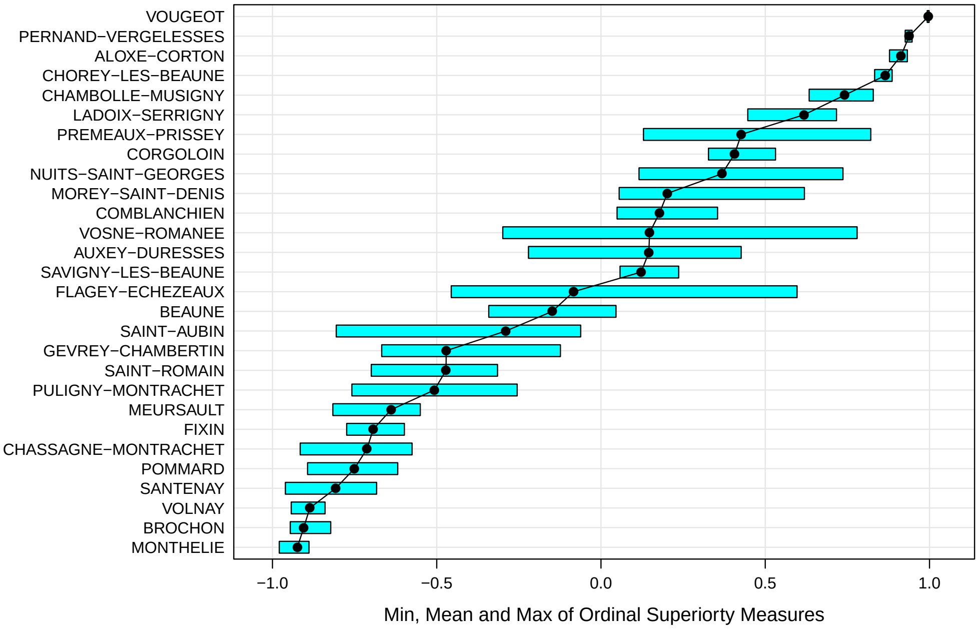
```
plot(gamod[[ 1]], pages= 1, scale= 0)

plot(gamodM[[ 1]], pages= 1, scale= 0)
```



4.3 Ordinal superiority figure

From the equation XX of the working paper, we compute ordinal superiority measures for each OGAMs relatively to the average. It produces the Figure XX of the main text. Drop Chenôve, Marsannay, Couchey, for which the method is not appropriate.

```
library(latticeExtra)
plogi <- function(x) exp(x/ sqrt(2))/ (1+ exp(x/ sqrt(2)))
xx <- data.frame(sapply(gamod, function(x)
    2* plogi(I(x$coeff[ 4: 31]- mean(x$coeff[ 4: 31])))- 1))
ww <- data.frame(xx,
                 LIBCOM= substr(names(gamod[[1]]$coef[ 4: 31]), 7, 30),
                 MIN= apply(xx[ 7: 10], 1, min),
                 MAX= apply(xx[ 7: 10], 1, max),
                 MEAN= apply(xx[ 7: 10], 1, mean))
segplot(reorder(factor(LIBCOM), MEAN)~ MIN+ MAX, length= 5, draw.bands= T,
        data= ww[order(ww$MEAN), ], center= MEAN, type= "o",
        unit = "mm", axis = axis.grid, col.symbol= "black", cex= 1,
        xlab= "Min, Mean and Max of Ordinal Superiorty Measures")


yy <- data.frame(sapply(gamodM, function(x)
    2* plogi(I(x$coeff[ 4: 31]- mean(x$coeff[ 4: 31])))- 1))
zz <- data.frame(yy,
                 LIBCOM= substr(names(gamodM[[1]]$coef[ 4: 31]), 7, 30),
                 MIN= apply(yy[ 5: 8], 1, min),
                 MAX= apply(yy[ 5: 8], 1, max),
                 MEAN= apply(yy[ 5: 8], 1, mean))
segplot(reorder(factor(LIBCOM), MEAN)~ MIN+ MAX, length= 5, draw.bands= T,
        data= zz[order(zz$MEAN), ], center= MEAN, type= "o",
        unit = "mm", axis = axis.grid, col.symbol= "black", cex= 1,
          xlab= "Min, Mean and Max of Ordinal Superiorty Measures")
```

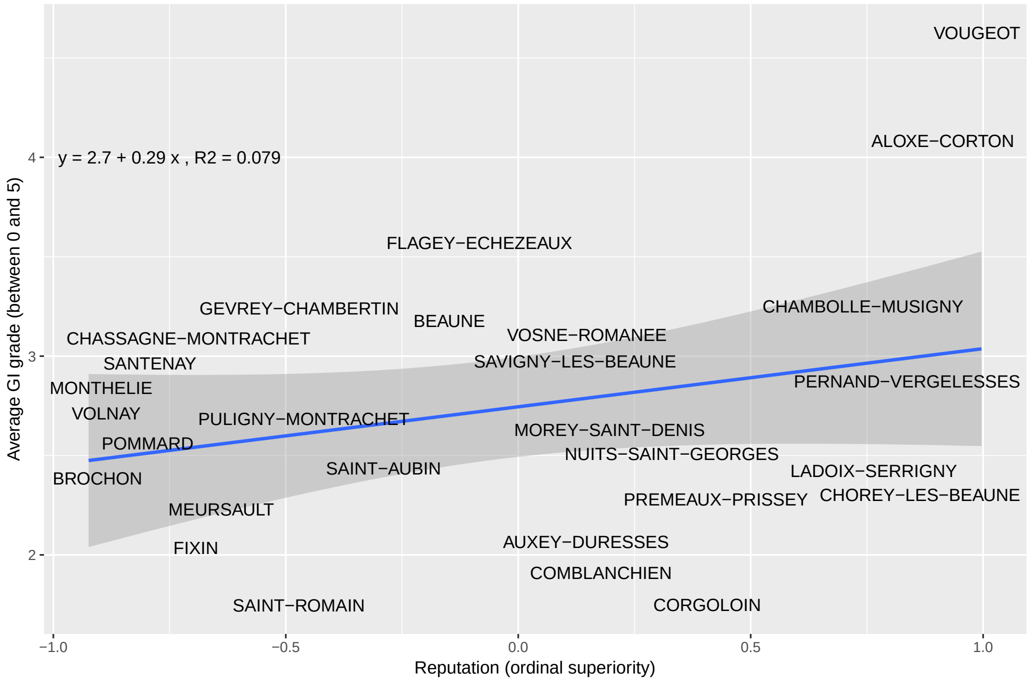


4.4 Correlation between *Communes*

Below an unreported Figure to illustrate the claim that "*commune* with higher GIs do not have a preferential treatment" (p.XX) of the working paper. It correlates the average vertical GI score with the ordinal superiority measures from OGAM with XX maximum effective degrees of freedom.

```
library(plyr) ; library(ggrepel)
yy <- ddply(RRank@data, .(LIBCOM),
            function(x) weighted.mean(x$AOCc, x$Area))
zz <- merge(ww, yy, by= "LIBCOM")
m <- lm(V1~ MEAN, data= zz)
a <- signif(coef(m)[1], digits = 2)
b <- signif(coef(m)[2], digits = 2)
c <- signif(summary(m)$r.sq, digits = 2)
textlab <- paste("y = ", a, " + ", b, " x ", ", R2 = ", c, sep= "")
ggplot(zz, aes(MEAN, V1, label= LIBCOM)) +
    geom_smooth(method= lm, aes(MEAN, V1))+
    geom_text_repel(point.padding = NA) +
    annotate("text", x= -.75, y= 4, label= textlab, size= 4, parse= F)+
    xlab("Reputation (ordinal superiority)") +
    ylab("Average GI grade (between 0 and 5)")

aa <- ddply(Reg.Ras@data, .(LIBCOM),
            function(x) weighted.mean(x$AOC, x$AREA))
bb <- merge(zz, aa, by= "LIBCOM")

m <- lm(V1~ MEAN, data= bb)
a <- signif(coef(m)[1], digits = 2)
```

```
b <- signif(coef(m)[2], digits = 2)
c <- signif(summary(m)$r.sq, digits = 2)
textlab <- paste("y = ", a, " + ", b, " x ", ", R2 = ", c, sep= "")
ggplot(bb, aes(MEAN, V1, label= LIBCOM)) +
    geom_smooth(method= lm, aes(MEAN, V1))+
    geom_text_repel(point.padding = NA) +
    annotate("text", x= -.75, y= 4, label= textlab, size= 4, parse= F)+
    xlab("Reputation (ordinal superiority)") +
    ylab("Average GI grade (between 0 and 5)")
```



5 Informational content

5.1 Decomposition table

see appendix for the detailed presentation of the R code to implement the decomposition decompositions. The following code for all OGAMs some computation times, allow the reader to compute the models individually.

```
load("Inter/gamod.Rda") ; source("myFcts.R")
ddtt <- data.frame(AOCc= RRank$AOCc, LIBCOM= RRank$LIBCOM,
                   sapply(gamod[ 1: 5* 2], function(x)
                       rowSums(predict(x, type= 'terms')[, -1])))
dcmp <- sapply(names(ddtt[, 3: 7]), function(x)
    c("Total Signal"= var(ddtt[, x]), "Total Noise"= pi^2/ 3,
      jointSignal(ddtt, x),                     jointNoise(ddtt, x),
      vertiSignal(ddtt, x), vertiResid(ddtt, x), vertiNoise(ddtt, x),
      horizSignal(ddtt, x), horizResid(ddtt, x), horizNoise(ddtt, x)))
round(t(apply(dcmp, 1, function(x) x/ (pi^2/ 3+ dcmp[1, ])* 100)), 1)
```

```
ddtt <- data.frame(AOC= Reg.Ras$AOC, LIBCOM= Reg.Ras$LIBCOM,
                   sapply(gamodM[ 1: 4* 2], function(x)
                       rowSums(predict(x, type= 'terms')[, -1])))

dcmp <- sapply(names(ddtt[, 3: 6]), function(x)
    c("Total Signal"= var(ddtt[, x]), "Total Noise"= pi^2/ 3,
      jointSignal(ddtt, x, vt= "AOC"),
      jointNoise(ddtt, x, vt= "AOC"),
      vertiSignal(ddtt, x, vt= "AOC"),
      vertiResid(ddtt, x, vt= "AOC"),
      vertiNoise(ddtt, x, vt= "AOC"),
      horizSignal(ddtt, x, vt= "AOC"),
      horizResid(ddtt, x, vt= "AOC"),
      horizNoise(ddtt, x, vt= "AOC")))

round(t(apply(dcmp, 1, function(x) x/ (pi^2/ 3+ dcmp[1, ])* 100)), 1)
```

```
                 gam100 gam300 gam500 gam700 gam900
Signal             84.8   94.7   95.9   96.8   97.6
Noise              15.2    5.3    4.1    3.2    2.4
Joint Signal       68.9   78.5   76.0   77.9   78.7
Joint Noise        16.0   16.2   20.0   18.9   18.9
Rank Signal        55.1   40.3   56.8   61.3   57.6
Rank Residual      13.8   38.2   19.2   16.5   21.2
Rank Noise         29.7   54.4   39.1   35.4   40.0
Com Signal         21.3   37.2   24.6   27.5   29.1
Com Residual       47.6   41.3   51.4   50.4   49.7
Com Noise          63.5   57.5   71.3   69.3   68.5
```

6 Models for GIs of 1936

6.1 Descriptive statistics

I present here the detail of the analysis with past GIs, to show that *communes* influences have decreased and informational content has increased since then. It typically makes the same analysis than for actual GIs, first some descriptive statistics.

```
Reg.Old <- subset(Reg.Rank, !is.na(Reg.Rank$AOC36lvl) &
                  !Reg.Rank$LIBCOM %in%
                  c("CHENOVE", "MARSANNAY-LA-COTE", "COUCHEY",
                    "COMBLANCHIEN","CORGOLOIN", "SAINT-ROMAIN"))
Reg.Old$LIBCOM <- factor(Reg.Old$LIBCOM)
Reg.Old$AOCo <- as.numeric(ifelse(Reg.Old$AOC36lvl== "0", 1,
                           ifelse(Reg.Old$AOC36lvl== "3", 2, 3)))
table(Reg.Old$AOC36lvl, Reg.Old$AOCc)
#table(Reg.Old$LIBCOM, Reg.Old$AOCo)

Reg.Old <- subset(Reg.Ras, !is.na(Reg.Ras$AOC36lvl) &
                  !Reg.Ras$LIBCOM %in%
                  c("CHENOVE", "MARSANNAY-LA-COTE", "COUCHEY",
                    "COMBLANCHIEN","CORGOLOIN", "SAINT-ROMAIN"))
```

```
Reg.Old$LIBCOM <- factor(Reg.Old$LIBCOM)
Reg.Old$AOCo <- as.numeric(ifelse(Reg.Old$AOC36lvl== "0", 1,
                           ifelse(Reg.Old$AOC36lvl== "3", 2, 3)))
table(Reg.Old$AOC36lvl, Reg.Old$AOC)
# table(Reg.Old$LIBCOM, Reg.Old$AOCo)
```

```
        1     2     3     4     5
0    7204 12605  4120   567    39
3      15   662 15378  8017   261
5       0     1    13     3  1604
```

6.2 Estimation

The estimation of both the parametric and OGAMs, long computation times for the latter, prefer to fit models individually.

```
library(MASS)
por2 <- polr(factor(AOC)~ 0+ LIBCOM+ EXPO
             + poly(DEM, 2)+ poly(SLOPE, 2)+ poly(RAYAT, 2)
             + poly(X, 3)* poly(Y, 3), data= Reg.Old, Hess= T)
por2a <- polr(factor(AOC)~ 0+ EXPO
              + poly(DEM, 2)+ poly(SLOPE, 2)+ poly(RAYAT, 2)
              + poly(X, 3)* poly(Y, 3), data= Reg.Old, Hess= T)
por2b <- polr(factor(AOC)~ 0+ LIBCOM+ EXPO
              + poly(DEM, 2)+ poly(SLOPE, 2)+ poly(RAYAT, 2)
            , data= Reg.Old, Hess= T)

qor2 <- polr(factor(AOC)~ 0+ LIBCOM+ EXPO+ GEOL+ PEDO
             + poly(DEM, 2)+ poly(SLOPE, 2)+ poly(RAYAT, 2)
             + poly(X, 3)* poly(Y, 3), data= Reg.Old, Hess= T)
qor2a <- polr(factor(AOC)~ 0+ EXPO+ GEOL+ PEDO
              + poly(DEM, 2)+ poly(SLOPE, 2)+ poly(RAYAT, 2)
              + poly(X, 3)* poly(Y, 3), data= Reg.Old, Hess= T)
qor2b <- polr(factor(AOC)~ 0+ LIBCOM+ EXPO+ GEOL+ PEDO
              + poly(DEM, 2)+ poly(SLOPE, 2)+ poly(RAYAT, 2)
            , data= Reg.Old, Hess= T)

library(mgcv)
listk <- c(50, 75, 100, 150, 200, 250, 300)
gamoldM <- vector("list", length(listk))

system.time(
    for (i in 1: length(listk)){
        gamoldM[[ i]] <- gam(AOCo~ 0+ LIBCOM+ EXPO
                             + s(DEM)+ s(SLOPE)+ s(RAYAT)
                             + s(X, Y, k= listk[ i])+ GEOL+ PEDO
                        , data= Reg.Old, family= ocat(R= 3))
    }
)
names(gamoldM) <- paste0("gam", listk)
save(gamoldM, file= "Inter/gamoldM.Rda")
## utilisateur     système      écoulé
##     24064.0       178.8     24243.2
```

```
gammold <- vector("list", length(listk))
system.time(
for (i in 1: length(listk)){
    gammold[[ i]] <- gam(AOCo~ 0+ EXPO+ s(DEM)+ s(SLOPE)+ s(RAYAT)
                         + s(X, Y, k= listk[ i])
                       , data= Reg.Old, family= ocat(R= 3))
})
names(gammold) <- paste0("gam", listk)
save(gammold, file= "Inter/gammold.Rda")
```

```
utilisateur     système      écoulé
    12259.5       144.1     12405.5
utilisateur     système      écoulé
    9582.37       78.69     9661.62
```

6.3 Significance

Significance of all models of GIs designation, corresponds to Table XX in Appendix of the working paper.

```
load("Inter/gamold.Rda") ; load("Inter/gamoldM.Rda")
res2a <- anova(por2, por2b)
qes2a <- anova(qor2, qor2b)

res2 <- Anova(por2)
sapply(gamold[ 3: 7], resume)

qes2 <- Anova(qor2)
sapply(gamoldM[ 3: 7], resume)
```

```
            gam100  gam150  gam200  gam250  gam300
s(DEM)       499.8   647.4   702.3   541.9   344.5
               8.5     8.2     8.8     8.4     7.7
s(SLOPE)     387.3   314.0   254.4   244.3   153.0
               8.7     8.7     8.6     8.6     8.3
s(RAYAT)     242.0   160.1   127.1   122.9   105.2
               8.5     8.3     8.1     5.0     5.9
s(X,Y)     17520.5 20194.2 22301.7 23507.2 23801.4
              98.3   146.3   194.4   239.8   286.6
LIBCOM      2782.5  1843.0  1642.4  1283.0  1049.4
              25.0    25.0    25.0    25.0    25.0
EXPO         119.8    91.8    91.9    96.1    90.2
               7.0     7.0     7.0     7.0     7.0
```

6.4 Goodness of fit

Goodness of fit measures from the same Table XX in Appendix.

```
round(c(McFaddenR2= psR2(por2), AIC= AIC(por2)/ 1000,
        Pcgp= sum(diag(table(predict(por2), Reg.Old$AOCo)))/ nrow(Reg.Old)), 2)
rbind(Pcgp= sapply(gamold, pcgp), AIC= sapply(gamold, AIC))
# sapply(gamold, psR2)

round(c(McFaddenR2= psR2(qor2), AIC= AIC(qor2)/ 1000,
        Pcgp= sum(diag(table(predict(qor2), Reg.Old$AOCo)))/ nrow(Reg.Old)), 2)
rbind(Pcgp= sapply(gamoldM, pcgp), AIC= sapply(gamoldM, AIC))
sapply(gamoldM, psR2)
```

```
McFaddenR2        AIC       Pcgp
      0.38      51.29       0.79
        gam50    gam75   gam100   gam150   gam200  gam250   gam300
Pcgp    84.34     85.9    87.08    89.26    90.28    91.4    92.54
AIC  40789.58 36833.3 33810.36 30271.01 27574.12 24526.6 22482.20
```

6.5 Omitted variable

Bootstrapped statistics for omitted variables, not reported in the working paper, mentioned at p.XX, .

```
library(lmtest) ; library(sandwich) ; library(sure)
wal3 <- rep(NA, nsim= 100)
for (i in 1: nsim){
    tmp <- surrogate(qor2a)- qor2a$lp
    wal3[ i] <- waldtest(lm(tmp~ Reg.Old$LIBCOM), . ~ 1, vcov= vcovHC)$F[ 2]
}
quantile(wal3, c(.05, .5, .95))
load("Inter/gammold.Rda") ; load("Inter/gammoldM.Rda") ; source("myFcts.R")
wal4 <- sapply(gammold, function(x) omitVar(x, SRank$LIBCOM, nsim= 100))
wold <- data.frame(logit= wal3, wal4)
apply(wold, 2, function(x) quantile(x, c(.05, .5, .95)))
```

```
    logit gam50  gam75 gam100 gam150 gam200 gam250 gam300
5%  168.1 7.408  7.340  4.714  3.498  2.057  1.178  1.091
50% 173.6 8.553  8.843  5.894  4.310  2.709  1.832  1.488
95% 179.8 9.958 10.501  6.858  5.396  3.851  2.495  2.057
```

The same plot as for current GIs, same evidences about the relevance of spatial smoothing terms, the non significance is reach for smaller degrees of freedom (p.XX)

```
library(lattice)
poldat <- stack(wold)
bwplot(values~ ind, data= poldat, type=c("l","g"), horizontal= FALSE,
       xlab='Model of GI designation', ylab='Bootstraped F-statistics',
       par.settings = list(box.rectangle=list(col='black'),
                           plot.symbol = list(pch='.', cex = 0.1)),
       scales=list(y= list(log= TRUE)),
       panel = function(..., box.ratio) {
           panel.grid(h= -1, v = -11)
           panel.violin(..., col = "lightblue",
                        varwidth = FALSE, box.ratio = box.ratio)
           panel.bwplot(..., col='black',
                        cex=0.8, pch='|', fill='gray', box.ratio = .1)
           panel.abline(h= log(1.47), col= "red", lty= 3)
           panel.text(2, log(1.55), "F= 1.47: critical value at 5%")})
```

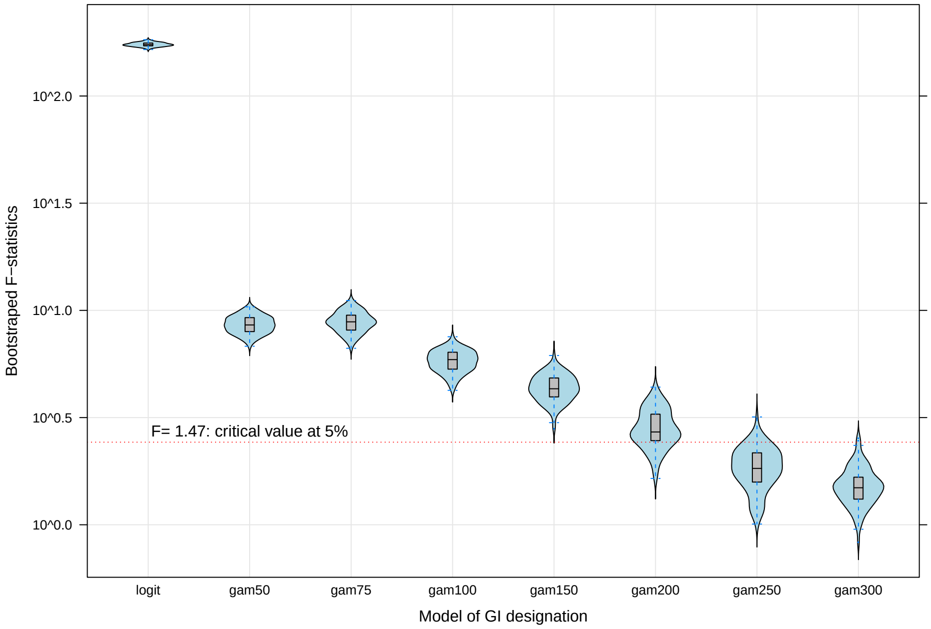


Figure 2: Effects of model XX

6.6 Specification

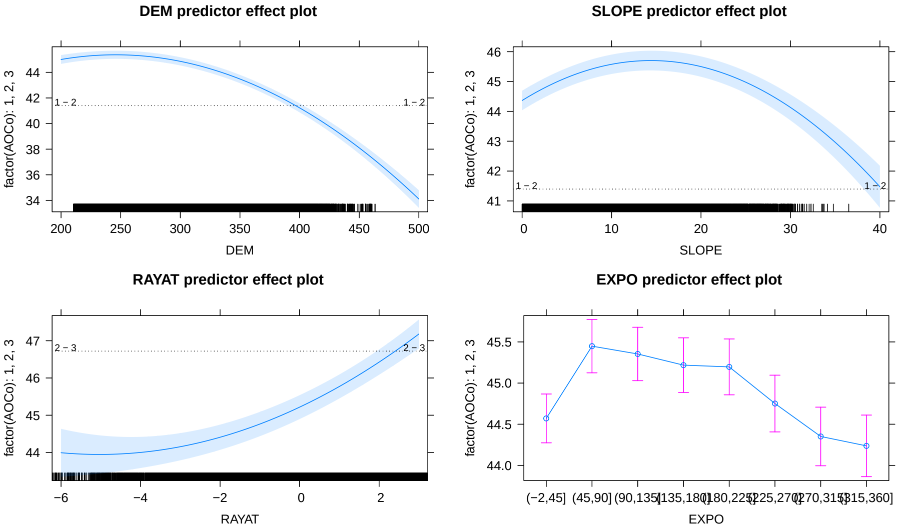
results not reported, parler de ce qu'il se passe moins bien mais qui n'est pas grave. Dans le gam 300 il y a un point qui fait n'imp, probablement un trou dans la carte de Florian.

```
library(sure) ; library(ggplot2) ; library(gridExtra)
var <- c("DEM", "SLOPE", "RAYAT", "EXPO", "LIBCOM", "X", "Y")
plots <- lapply(var, function(.x)
    autoplot(por2, what= "covariate", x= Reg.Old@data[, .x], xlab= .x))
do.call(grid.arrange, c(list(autoplot(por2, what= "qq")), plots))

restmp <- sureOGAM(gamold$gam300)- gamold$gam300$line
plot(qlogis(1: nrow(SRank)/ nrow(SRank), scale= 1), sort(restmp))
abline(0, 1)
var <- c("DEM", "SLOPE", "RAYAT", "EXPO", "LIBCOM", "X", "Y")
par(mfrow= c(3, 3)) ; for (i in var) pltSURE(restmp, SRank@data[, i], i)
```

6.7 Marginal effects

Marginal effect ca be assessed, corresponds to Figure XX in the appendix in the working paper.

```
library(effects)
plot(predictorEffects(por2, ~ DEM+ SLOPE+ RAYAT+ EXPO, latent= TRUE,
                      xlevels=list(DEM= 200: 500,
                                   SLOPE= 0: 400/ 10, RAYAT= -60: 30/ 10)))
plot(gamold$gam300, pages= 1, scale= 0)

plot(predictorEffects(qor2, ~ DEM+ SLOPE+ RAYAT+ EXPO, latent= TRUE,
                      xlevels=list(DEM= 200: 500,
                                   SLOPE= 0: 400/ 10, RAYAT= -60: 30/ 10)))
plot(gamoldM$gam300, pages= 1, scale= 0)
```



6.8 Ordinal superiority

Ordinal superiority of *commune* from the GIs of 1936, same equation XX of the working paper and Figure XX in the appendix.

```
xxx <- data.frame(sapply(gamold, function(x)
    2* plogi(I(x$coeff[ 1: 25]- mean(x$coeff[ 1: 25]))- 1))
www <- data.frame(xxx,
                  LIBCOM= substr(names(gamold[[ 1]]$coef[ 1: 25]), 7, 30),
                  MIN= apply(xxx[ 6: 7], 1, min),
                  MAX= apply(xxx[ 6: 7], 1, max),
                  MEAN= apply(xxx[ 6: 7], 1, mean))
segplot(reorder(factor(LIBCOM), MEAN)~ MIN+ MAX, length= 5, draw.bands= T,
        data= www[order(www$MEAN), ], center= MEAN, type= "o",
        unit = "mm", axis = axis.grid, col.symbol= "black", cex= 1,
        xlab= "Min, Mean and Max of Ordinal Superiorty Measures")

xxx
summary(gamoldM[[7]])
```

```
xxx <- data.frame(sapply(gamoldM, function(x)
    2* plogi(I(x$coeff[ 1: 25]- mean(x$coeff[ 1: 25]))- 1))
www <- data.frame(xxx,
                  LIBCOM= substr(names(gamoldM[[ 1]]$coef[ 1: 25]), 7, 30),
                  MIN= apply(xxx[ 6: 7], 1, min),
                  MAX= apply(xxx[ 6: 7], 1, max),
                  MEAN= apply(xxx[ 6: 7], 1, mean))
segplot(reorder(factor(LIBCOM), MEAN)~ MIN+ MAX, length= 5, draw.bands= T,
        data= www[order(www$MEAN), ], center= MEAN, type= "o",
        unit = "mm", axis = axis.grid, col.symbol= "black", cex= 1,
        xlab= "Min, Mean and Max of Ordinal Superiorty Measures")
```

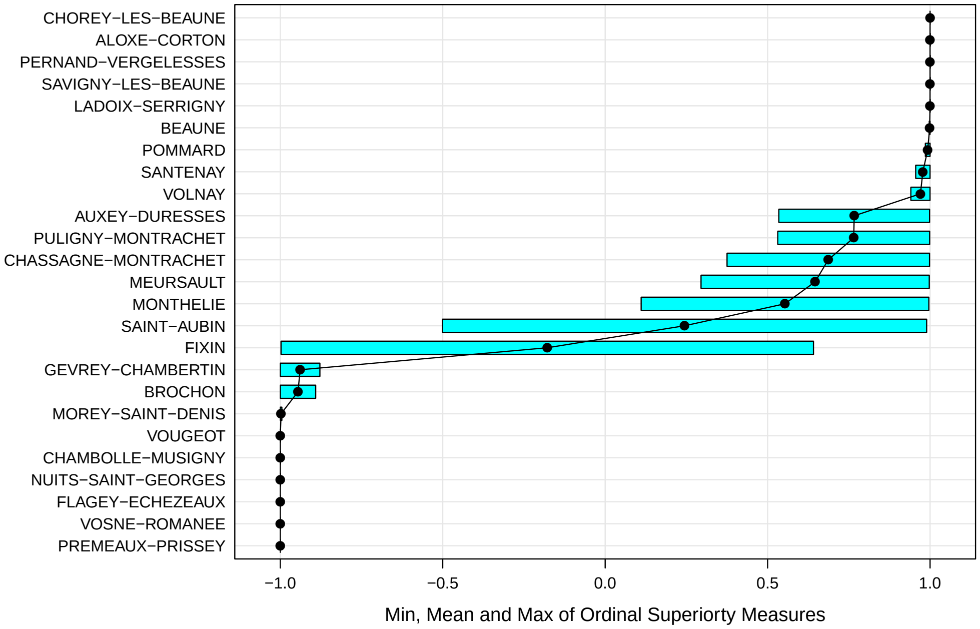
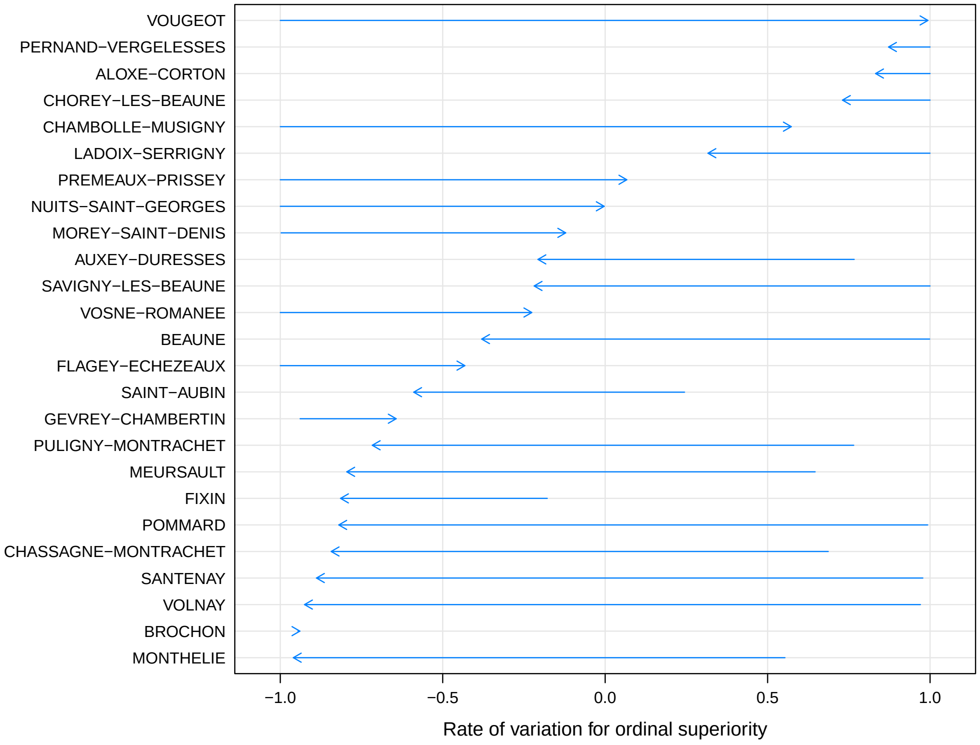


Figure 3: Effects of model XX

6.9 Correlation between models

An additional unreported Figure to show the claim that "the importance of *communes* has decreased since the 1936 scheme" (p.XX)

```
zzz <- merge(ww, www, by= "LIBCOM")
segplot(reorder(factor(LIBCOM), MEAN.x)~ MEAN.y+ MEAN.x, data= zzz,
        segments.fun = panel.arrows, length = 2, unit = "mm",
        draw.bands= F, axis = axis.grid,
        xlab= "Rate of variation for ordinal superiority")
```



6.10 Decomposition table

And then the decomposition table unreported in the main text that show the "smaller joint informational content of GIs in 1936" (p.XX).

```
load("Inter/gamold.Rda") ; source("myFcts.R")
ddoo <- data.frame(AOCavt= SRank$AOCavt, LIBCOM= SRank$LIBCOM,
                   sapply(gamold, function(x)
                       rowSums(predict(x, type= 'terms')[, -1])))
dcop <- sapply(names(ddoo[, 3: 9]), function(x)
    c("Total Signal"= var(ddoo[, x]), "Total Noise"= pi^2/ 3,
      jointSignal(ddoo, x, "AOCavt"), jointNoise(ddoo, x, "AOCavt"),
      vertiSignal(ddoo, x, "AOCavt"), vertiResid(ddoo, x, "AOCavt"), vertiNoise(ddoo, x, "AOCavt"),
      horizSignal(ddoo, x, "AOCavt"), horizResid(ddoo, x, "AOCavt"), horizNoise(ddoo, x, "AOCavt")))
round(t(apply(dcop, 1, function(x) x/ (pi^2/ 3+ dcop[1, ])* 100)), 1)
```

```
              gam50 gam75 gam100 gam150 gam200 gam250 gam300
Signal         95.6  93.1   95.4   98.7   98.1   99.5   99.5
Noise           4.4   6.9    4.6    1.3    1.9    0.5    0.5
Joint Signal   78.7  63.2   55.3   75.2   47.9   75.0   45.1
Joint Noise    16.9  29.9   40.2   23.5   50.3   24.5   54.5
Rank Signal     5.8  18.1   24.1   16.4   20.6   14.9   22.7
Rank Noise     89.8  75.0   71.3   82.4   77.5   84.6   76.8
Rank Residual  72.9  45.1   31.2   58.8   27.3   60.1   22.4
```

```
Com Signal     67.5  39.6   29.4   62.3   24.0   62.7   22.6
Com Noise      28.1  53.5   66.0   36.4   74.1   36.8   77.0
Com Residual   16.0  33.3   43.7   20.9   35.3   20.6   43.7
```

7 Alternative GI designations

7.1 Change latent vineyard quality

We conclude this work with the simulations of alternative GIs designations schemes. Below are scenarios XX from XX, need to run the code. Put the equations here.

```
load("Inter/gamod.Rda")
prdd <- predict(gamod$gam900, type= 'terms')
thsld <- c(-Inf, gamod$gam900$family$getTheta(TRUE), Inf)
ltt0 <- rowSums(prdd)- (sureOGAM(gamod$gam900)- gamod$gam900$line)
ltt1 <- rowSums(prdd)
ltt2 <- mean(prdd[, 1])+ rowSums(prdd[, -1])-
    (sureOGAM(gamod$gam900)- gamod$gam900$line)
ltt3 <- mean(prdd[, 1])+ rowSums(prdd[, -1])
## CHANGER RRank$AOCavt
Simu <- data.frame(RRank, ltt= rowSums(prdd[, -1]),
                   OLD= RRank$AOCavt, S0= cut(ltt0, thsld),
                   SI= cut(ltt1, thsld), SII= cut(ltt2, thsld),
                   SIII= cut(ltt3, thsld))
table(Simu$AOCc, Simu$S0) ; table(Simu$AOCc, Simu$SI)
table(Simu$AOCc, Simu$SII) ; table(Simu$AOCc, Simu$SIII)
```

```
               OLD  CF1  CF2  CF3  CF4  CF5  CF6
Signal        97.1 97.1 97.1 97.1 97.1 97.1 97.1
Noise          2.9  2.9  2.9  2.9  2.9  2.9  2.9
Joint Signal  51.4 80.1 81.2 82.2 79.4 80.0 79.2
Joint Noise   45.8 17.1 15.9 15.0 17.7 17.1 18.0
Rank Signal   38.9 70.7 64.5 73.5 62.2 62.8 62.0
Rank Noise    58.2 26.4 32.6 23.6 34.9 34.3 35.1
Rank Residual 12.5  9.4 16.7  8.7 17.2 17.2 17.2
Com Signal    28.5 28.5 28.5 28.5 28.5 28.5 28.5
Com Noise     68.6 68.6 68.6 68.6 68.6 68.6 68.6
Com Residual  22.9 51.6 52.7 53.7 50.9 51.5 50.7
```

7.2 Add a vertical level in GIs

Below are the simulations from scenarios XX, XX, and XX, according to changing XX. Put the equations here.

```
thrldBOUR <- mean(ltt0[RRank$AOCc== 2])
thrldVILL <- mean(ltt0[RRank$AOCc== 3])
thrldPCRU <- mean(ltt0[RRank$AOCc== 4])
Simv <- data.frame(Simu,
                    SIV= ifelse(RRank$AOCc< 2, RRank$AOCc,
                         ifelse(RRank$AOCc== 2 & ltt0< thrldBOUR, 2,
                         ifelse(RRank$AOCc== 2 & ltt0>= thrldBOUR, 3,
                                RRank$AOCc+ 1))),
                    SV = ifelse(RRank$AOCc< 3, RRank$AOCc,
                         ifelse(RRank$AOCc== 3 & ltt0< thrldVILL, 3,
                         ifelse(RRank$AOCc== 3 & ltt0>= thrldVILL, 4,
                                RRank$AOCc+ 1))),
                    SVI= ifelse(RRank$AOCc< 4, RRank$AOCc,
                         ifelse(RRank$AOCc== 4 & ltt0< thrldPCRU, 4,
                         ifelse(RRank$AOCc== 4 & ltt0>= thrldPCRU, 5,
                                RRank$AOCc+ 1))))
table(Simv$AOCc, Simv$SIV)
table(Simv$AOCc, Simv$SV) ; table(Simv$AOCc, Simv$SVI)
```

```
   
        1     2     3     4     5     6
  1  9759     0     0     0     0     0
  2     0  8931  6577     0     0     0
  3     0     0     0 24151     0     0
  4     0     0     0     0  8577     0
  5     0     0     0     0     0  1906

   
        1     2     3     4     5     6
  1  9759     0     0     0     0     0
  2     0 15508     0     0     0     0
  3     0     0 13275 10876     0     0
  4     0     0     0     0  8577     0
  5     0     0     0     0     0  1906

   
        1     2     3     4     5     6
  1  9759     0     0     0     0     0
  2     0 15508     0     0     0     0
  3     0     0 24151     0     0     0
  4     0     0     0  4970  3607     0
  5     0     0     0     0     0  1906
```

7.3 Decomposition table

And the decomposition Table which corresponds to Table XX in the working paper.

```
decf <- sapply(names(Simv[, 100: 107]), function(x)
    c("Total Signal"= var(Simv[, "ltt"]), "Total Noise"= pi^2/ 3,
      jointSignal(Simv, "ltt", vt= x), jointNoise(Simv, "ltt", vt= x),
      vertiSignal(Simv, "ltt", vt= x), vertiResid(Simv, "ltt", vt= x), vertiNoise(Simv, "ltt", vt= x),
      horizSignal(Simv, "ltt", vt= x), horizResid(Simv, "ltt", vt= x), horizNoise(Simv, "ltt", vt= x)))
round(t(apply(decf, 1, function(x) x/ (pi^2/ 3+ decf[1, ])* 100)), 1)
```

```
                   OLD    S0    SI   SII  SIII   SIV    SV   SVI
```

```
Total Signal        97.6 97.6 97.6 97.6 97.6 97.6 97.6 97.6
Total Noise          2.4  2.4  2.4  2.4  2.4  2.4  2.4  2.4
Joint Signal        50.7 78.4 80.7 81.1 82.8 79.2 79.7 79.0
Joint Noise         46.9 19.2 16.8 16.5 14.8 18.4 17.9 18.6
Vertical Signal     35.9 56.8 59.8 70.7 73.1 58.1 58.5 58.0
Vertical Residual   14.9 21.6 21.0 10.4  9.7 21.1 21.2 21.1
Vertical Noise      61.7 40.8 37.8 26.9 24.5 39.4 39.1 39.6
Horizontal Signal   29.1 29.1 29.1 29.1 29.1 29.1 29.1 29.1
Horizontal Residual 21.6 49.3 51.7 52.0 53.7 50.1 50.6 50.0
Horizontal Noise    68.5 68.5 68.5 68.5 68.5 68.5 68.5 68.5
```

8 Session information

```
sessionInfo()
```

```
R version 3.5.3 (2019-03-11)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 18.04.2 LTS

Matrix products: default
BLAS: /usr/lib/x86_64-linux-gnu/blas/libblas.so.3.7.1
LAPACK: /usr/lib/x86_64-linux-gnu/lapack/liblapack.so.3.7.1

locale:
 [1] LC_CTYPE=fr_FR.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=fr_FR.UTF-8        LC_COLLATE=fr_FR.UTF-8
 [5] LC_MONETARY=fr_FR.UTF-8    LC_MESSAGES=fr_FR.UTF-8
 [7] LC_PAPER=fr_FR.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=fr_FR.UTF-8 LC_IDENTIFICATION=C

attached base packages:
[1] stats4    stats     graphics  grDevices utils     datasets
[7] methods   base

other attached packages:
 [1] gridExtra_2.3      xtable_1.8-3       ggrepel_0.8.0
 [4] ggplot2_3.1.0      plyr_1.8.4         latticeExtra_0.6-28
 [7] RColorBrewer_1.1-2 effects_4.0-3      lattice_0.20-38
[10] truncdist_1.0-2    evd_2.3-3          sure_0.2.0
[13] sandwich_2.5-0     lmtest_0.9-36      zoo_1.8-4
[16] mgcv_1.8-28        nlme_3.1-137       car_3.0-2
[19] carData_3.0-1      MASS_7.3-51.1      sp_1.3-1

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.0        assertthat_0.2.0  R6_2.3.0
 [4] cellranger_1.1.0  survey_3.33-2     pillar_1.3.0
 [7] rlang_0.3.0.1     lazyeval_0.2.1    curl_3.2
[10] readxl_1.1.0      minqa_1.2.4       data.table_1.11.4
[13] nloptr_1.0.4      Matrix_1.2-17     labeling_0.3
[16] splines_3.5.3     rgdal_1.3-6       lme4_1.1-18-1
[19] foreign_0.8-71    munsell_0.5.0     compiler_3.5.3
[22] pkgconfig_2.0.2   nnet_7.3-12       tidyselect_0.2.5
[25] tibble_1.4.2      rio_0.5.10        crayon_1.3.4
[28] dplyr_0.7.8       withr_2.1.2       grid_3.5.3
[31] gtable_0.2.0      magrittr_1.5      scales_1.0.0
[34] zip_1.0.0         bindrcpp_0.2.2    openxlsx_4.1.0
```

```
[37] tools_3.5.3       forcats_0.3.0     glue_1.3.0
[40] purrr_0.2.5       hms_0.4.2         abind_1.4-5
[43] survival_2.43-3   colorspace_1.3-2  bindr_0.1.1
[46] haven_1.1.2
```

A Custom functions

A.1 Translation of geology

```
trans_geol <- data.frame(
    GEOL= Reg.Ras$GEOL[!duplicated(Reg.Ras$GEOL)],
    GEOf= c(
        "Calcaires massifs de \"Comblanchien\" (Bathonien sup.)",
        "Marnes et calcaires divers (Callovien inférieur)",
        "Marnes et calcaires argileux (Oxfordien moyen)",
        "Eboulis ordonnés cryoclastiques et colluvions diverses",
        "Oolithe ferrugineuse (Oxfordien moyen-sup)",
        "Calcaires hydrauliques de Molesmes et Noiron (Oxfordien sup.)",
        "Colluvions diverses",
        "Dépôts argilo-limoneux, sables et graviers du Villafranchien",
        "Calcaires de Tonnerre, Oisellemont et calcaires á Astartes",
        "Eboulis et glissements de terrains",
        "Calcaires grenus bicolores (Bathonien terminal)",
        "Terrasse argilo-limoneuse de Saint-Usage",
        "Formation de Saint-Cosme (marnes fluvio-lacustres varvées)",
        "Alluvions anciennes indifférenciées, argilo-limoneuses",
        "Calcaires bioclastiques, graveleux, á oolithes (Bathonien inf.)"
    ),
    GEOe= c(
        "Massive limestones from \"Comblanchien\" (upper Bathonian)",
        "Various marls and limestones (lower Callovian)",
        "Marls and argillaceous limestones (middle Oxfordian)",
        "Ordered cryoclastic scree and various colluviums",
        "Ferruginous Oolite (middle-upper Oxfordian)",
        "Hydraulic limestones of Molesmes and Noiron (upper Oxfordian)",
        "Various colluviums",
        "Clay-silt deposits, sand and gravel from Villafranchien",
        "Limestones of Thunder, Oisellemont and limestones in Astartes",
        "Screes and landslides",
        "Two-tone gray limestones (terminal Bathonian)",
        "Clay-silty terrace of Saint-Usage",
        "Formation of Saint-Cosme (varnished fluvio-lacustrine marls)",
        "Undifferentiated ancient alluvium, clay-silty",
        "Bioclastic limestones, gravelly, with oolites (lower Bathonian)")
)
```

A.2 Translation of pedology

```
trans_pedo <- data.frame(
    PEDO= Reg.Ras$PEDO[!duplicated(Reg.Ras$PEDO)],
    PEDf= c(
        "Vignoble de la Côte de de Beaune",
        "Cônes de déjection du pied de Côte",
        "Côteaux viticoles des Hautes Côtes de Nuits",
        "Courtes pentes marneuses des plateaux plio-pléistocène",
        "Piedmont de la côte viticole",
        "Versants pentus des Hautes Côtes de Beaune",
        "Sommets des collines des Hautes Côtes de Beaune",
        "Alluvions récentes calcaires des vallées (Ouche, Tille, Meuzin)",
        "Pentes liasiques du Haut-Auxois",
        "Basses terrasses gravelo-caillouteuses des plaines alluviales",
        "Basses terrasses argileuses des plaines alluviales",
        "Terrasse argilo-limoneuse de Saint-Usage",
        "Vignoble de la Côte de Nuits",
        "Rebord oriental des plateaux calcaires dominant la Côte viticole"
    ),
    PEDe= c(
        "Vineyard of the Côte de Beaune",
        "Coot footing cones",
        "Wine hills of Hautes Côtes de Nuits",
        "Oxfordian limestone-marly trays of the Hautes Côtes",
        "Short marly slopes of Plio-Pleistocene plateaus",
        "Piedmont of the vineyard of the Côte",
        "Sloping slopes of the Hautes Côtes de Beaune",
        "Summits of the hills of the Hautes Côtes de Beaune",
        "Recent alluvial limestone valleys (Ouche, Tille, Meuzin)",
        "Liastic slopes of Haut-Auxois",
        "Gravelo-stony low terraces of alluvial plains",
        "Low clay terraces of alluvial plains",
        "Vineyard of the Côte de Nuits",
        "Eastern edge of the limestone plateaus overlooking the Côte"
    )
)
```

A.3 Surrogate Residuals

The R package sure allows to simulate the surrogate residuals from a large panel of ordered parametric models (https://koalaverse.github.io/sure/index.html) but not for the semiparametric ordered generalized additive model fitted with the package mgcv. We first define the truncLogis function for the simulation of random draws from a truncated logistic distribution with a vector of inputs (locations and thresholds) as the package truncdist is only designed for a given value of location and thresholds. Then, we code the function surePOLR which simulate surrogate residuals from polr models from the MASS package. The code is test against the surrogate simulations from sure for a random ordered logistic model.

```
truncLogis <- function(n, spec, a = -Inf, b = Inf, ...) {
    p <- runif(n, min = 0, max = 1)
    G <- get(paste("p", spec, sep = ""), mode = "function")
    Gin <- get(paste("q", spec, sep = ""), mode = "function")
    G.a <- G(a, ...)
    G.b <- G(b, ...)
    pmin(pmax(a, Gin(G(a, ...) + p * (G(b, ...) - G(a, ...)), ...)), b)
}

surePOLR <- function(mod, newd= NULL){
    if (mod$method!= "logistic") stop("Logistic required")
    gg <- as.numeric(mod$zeta)
    if (is.null(newd)){
        g1 <- as.integer(model.response(model.frame(mod)))
        g6 <- mod$lp
    } else {
        g1 <- as.integer(newd[, "AOCc"])
        g6 <- gg[ 1]-qlogis(predict(mod, newdata= newd, type= 'probs')[, 1])
    }
    nn <- length(g1)
    suls <- sapply(g1, switch,
                   "1"= c(-Inf  , gg[ 1]), "2"= c(gg[ 1], gg[ 2]),
                   "3"= c(gg[ 2], gg[ 3]), "4"= c(gg[ 3], gg[ 4]),
                   "5"= c(gg[ 4], Inf   ))
    sls <- data.frame(unlist(t(suls)))
    truncLogis(nn, spec= "logis", a= sls[, 1], b= sls[, 2],
               location= g6, scale= 1)
}
```

```
library(MASS) ; library(sure) ; library(truncdist)
fit.polr <- polr(factor(AOCc)~ poly(DEM, 2)+ poly(SLOPE, 2)
                + poly(RAYAT, 2)+ poly(ASPECT, 2)+ poly(PERMEA, 2)
              , data= Reg.Rank)
sure1 <- surrogate(fit.polr)+ fit.polr$zeta[ 1]
sure2 <- resids(fit.polr)
polr1 <- surePOLR(fit.polr) ; polr2 <- surePOLR(fit.polr)- fit.polr$lp
```

Now we use the same structure to simulate the surrogate residuals for the OGAM through the function sureOGAM. Again, the function is tested for a random OGAM.

```
sureOGAM <- function(mod, newd= NULL){
    if (is.null(newd)){
        g1 <- as.integer(mod$y)
        g6 <- mod$linear.predictors
    } else {
        g1 <- as.integer(newd[, names(mod$model[ 1])])
        g6 <- predict(mod, newdata= newd)
    }
    nn <- length(g1)
    gt <- data.frame(rep(NA, nn), rep(NA, nn))
    gg <- c(mod$family$getTheta(TRUE), Inf)
    kk <- c(- Inf, gg[ 1])
    for (j in 2: length(unique(g1))) kk <- rbind(kk, c(gg[ j- 1], gg[ j]))
    gt <- data.frame(t(sapply(g1, function(x) kk[x, ])))
    truncLogis(nn, spec= "logis", a= gt[, 1], b= gt[, 2], location= g6)
}
```

```
library(mgcv)
fit.ogam <- gam(AOCc~ poly(DEM, 2)+ poly(SLOPE, 2)
                + poly(RAYAT, 2)+ poly(ASPECT, 2)+ poly(PERMEA, 2)
              , family= ocat(R= 5), data= Reg.Rank)
ogam1 <- sureOGAM(fit.ogam)
ogam2 <- sureOGAM(fit.ogam)- fit.ogam$linear.pred

par(mfrow= c(3, 2))
plot(sure1, polr1)
abline(h= fit.polr$zeta, v= fit.polr$zeta, lty= 2, col= "blue")
abline(0, 1, col= "orange")
plot(sure2, polr2)
abline(0, 1, col= "orange")

plot(polr1, ogam1- mean(ogam1))
abline(h= fit.ogam$family$getTheta(TRUE)- mean(ogam1),
       v= fit.polr$zeta, lty= 2, col= "blue")
abline(0, 1, col= "orange")
plot(polr2, ogam2)
abline(0, 1, col= "orange")

plot(sure1, ogam1- mean(ogam1))
abline(h= fit.ogam$family$getTheta(TRUE)- mean(ogam1),
       v= fit.polr$zeta, lty= 2, col= "blue")
abline(0, 1, col= "orange")
plot(sure2, ogam2)
abline(0, 1, col= "orange")
```

A.4 Decomposition terms

For each terms of the decomposition presented in the main text, we code a different functions presented below. For the ease of notations, we note for the values of the latent and the probabilities of being in each GIs $x = y, p$:

$$\overline{x}_{jc} = \frac{1}{N}\sum\nolimits_{i=1}^{N} x_i \quad \text{and} \quad \overline{x}_{j.} = \frac{1}{C}\sum\nolimits_{c=1}^{C} x_{jc} \quad \text{and} \quad \overline{x}_{.c} = \frac{1}{J}\sum\nolimits_{j=1}^{J} x_{jc} \tag{1}$$

The **joint signal** terms is the variance of the expected quality conditionally on vertical and horizontal dummies:

$$\mathbb{V}\{\,\mathbb{E}[q(X^*) \mid y, c]\,\} = \frac{1}{N+J+H}\sum_{i=1}^{N}\sum_{j=1}^{J}\sum_{h=1}^{H}[\mathbb{E}(q(X^*) \mid y = j, c = h) - \overline{q}_{jh}]^2 \tag{2}$$

```
jointSignal <- function(dat, lt, vt= "AOCc", hz= "LIBCOM"){
    jS <- rep(0, nrow(dat))
    for (i in unique(dat[, vt])){
        for (j in unique(dat[, hz])){
            tmp <- dat[, vt]== i & dat[, hz]== j
            jS[ tmp] <- mean(dat[tmp, lt])
        }
    }
    c("Joint Signal"= var(jS))
}
```

The **joint noise** terms is the expectation of the variance quality conditionally on vertical and horizontal dummies:

$$\mathbb{E}\{\,\mathbb{V}[q(X^*) \mid y, c]\,\} = \frac{1}{N+J+H}\sum_{i=1}^{N}\sum_{j=1}^{J}\sum_{h=1}^{H}[\mathbb{E}(q(X^*) \mid y = j, c = h) - \overline{q}_{jh}]^2 \tag{3}$$

```
jointNoise <- function(dat, lt, vt= "AOCc", hz= "LIBCOM"){
    jN <- 0
    for (i in unique(dat[, vt])){
        for (j in unique(dat[, hz])){
            tmp <- dat[, vt]== i & dat[, hz]== j
            if (sum(tmp)> 1) jN <- jN+ var(dat[ tmp, lt])* mean(tmp)
        }
    }
    c("Joint Noise"= jN)
}
```

The **vertical signal** terms is the variance of the expectation quality conditionally on vertical GI dummies:

$$\mathbb{V}\{\,\mathbb{E}[q(X^*) \mid y]\,\} = \frac{1}{N+J+H}\sum_{i=1}^{N}\sum_{j=1}^{J}\sum_{h=1}^{H}[\mathbb{E}(q(X^*) \mid y = j, c = h) - \overline{q}_{jh}]^2 \tag{4}$$

```
vertiSignal <- function(dat, lt, vt= "AOCc", hz= "LIBCOM"){
    vS <- rep(0, nrow(dat))
    for (i in unique(dat[, vt])){
        vS[ dat[, vt]== i] <- mean(dat[dat[, vt]== i, lt])
    }
    c("Vertical Signal"= var(vS))
}
```

The **vertical residual** terms is the expectation of the conditional on horizontal variance of the expectation quality conditionally on vertical GI dummies:

$$\mathbb{E}\{\,\mathbb{V}[\mathbb{E}(q(X^*)\mid y,c)\mid y]\,\} = \frac{1}{N+J+H}\sum_{i=1}^{N}\sum_{j=1}^{J}\sum_{h=1}^{H}[\mathbb{E}(q(X^*)\mid y=j,c=h)-\overline{q}_{jh}]^2 \tag{5}$$

```
vertiResid <- function(dat, lt, vt= "AOCc", hz= "LIBCOM"){
    sig <- rep(0, nrow(dat)) ; vR <- 0
    for (i in unique(dat[, vt])){
        for (j in unique(dat[, hz])){
            tmp <- dat[, vt]== i & dat[, hz]== j
            sig[ tmp] <- mean(dat[ tmp, lt])
        }
    }
    for (i in unique(dat[, vt])){
        vR <- vR+ var(sig[dat[, vt]== i])* mean(dat[, vt]== i)
    }
    c("Vertical Residual"= vR)
}
```

The **vertical Noise** terms is the expectation of the variance of the quality conditionally on vertical GI dummies:

$$\mathbb{E}\{\,\mathbb{V}[q(X^*)\mid y]\,\} = \frac{1}{N+J+H}\sum_{i=1}^{N}\sum_{j=1}^{J}\sum_{h=1}^{H}[\mathbb{E}(q(X^*)\mid y=j,c=h)-\overline{q}_{jh}]^2 \tag{6}$$

```
vertiNoise <- function(dat, lt, vt= "AOCc", hz= "LIBCOM"){
    vN <- 0
    for (i in unique(dat[, vt])){
        vN <- vN+ var(dat[dat[, vt]== i, lt])* mean(dat[, vt]== i)
    }
    c("Vertical Noise"= vN)
}
```

The **horizontal signal** terms is the variance of the expectation quality conditionally on horizontal GI dummies:

$$\mathbb{V}\{\,\mathbb{E}[q(X^{*})\mid c]\,\} = \frac{1}{N+J+H}\sum_{i=1}^{N}\sum_{j=1}^{J}\sum_{h=1}^{H}[\mathbb{E}(q(X^{*})\mid y=j,c=h)-\overline{q}_{jh}]^{2} \tag{7}$$

```
horizSignal <- function(dat, lt, vt= "AOCc", hz= "LIBCOM"){
    hS <- rep(0, nrow(dat))
    for (j in unique(dat[, hz])){
        hS[ dat[, hz]== j] <- mean(dat[dat[, hz]== j, lt])
    }
    c("Horizontal Signal"= var(hS))
}
```

The **horizontal residual** terms is the expectation of the conditional on vertical variance of the expectation quality conditionally on horizontal GI dummies:

$$\mathbb{E}\{\,\mathbb{V}[\mathbb{E}(q(X^{*})\mid y,c)\mid y]\,\} = \frac{1}{N+J+H}\sum_{i=1}^{N}\sum_{j=1}^{J}\sum_{h=1}^{H}[\mathbb{E}(q(X^{*})\mid y=j,c=h)-\overline{q}_{jh}]^{2} \tag{8}$$

```
horizResid <- function(dat, lt, vt= "AOCc", hz= "LIBCOM"){
    sig <- rep(0, nrow(dat)) ; hR <- 0
    for (i in unique(dat[, vt])){
        for (j in unique(dat[, hz])){
            tmp <- dat[, vt]== i & dat[, hz]== j
            sig[ tmp] <- mean(dat[ tmp, lt])
        }
    }
    for (j in unique(dat[, hz])){
        hR <- hR+ var(sig[dat[, hz]== j])* mean(dat[, hz]== j)
    }
    c("Horizontal Residual"= hR)
}
```

The **horizontal Noise** terms is the expectation of the variance of the quality conditionally on horizontal GI dummies:

$$\mathbb{E}\{\,\mathbb{V}[q(X^{*})\mid c]\,\} = \frac{1}{N+J+H}\sum_{i=1}^{N}\sum_{j=1}^{J}\sum_{h=1}^{H}[\mathbb{E}(q(X^{*})\mid y=j,c=h)-\overline{q}_{jh}]^{2} \tag{9}$$

```
horizNoise <- function(dat, lt, vt= "AOCc", hz= "LIBCOM"){
    hN <- 0
    for (j in unique(dat[, hz])){
        hN <- hN+ (var(dat[dat[, hz]== j, lt])* mean(dat[, hz]== j))
    }
    c("Horizontal Noise"= hN)
```

```
 }
```