The Informational Content of Geographical Indications

Jean-Sauveur Ay*
INRA UMR CESAER

Reproducibility file Version 0.1 : April 13, 2019

Abstract

This file contents the R codes associated with the paper "The informational content of geographical indications" AAWE Working Paper No XXX. The data used are under licence Creative Commons Attribution Share Alike 4.0 International, available on the INRA dataverse website: https://data.inra.fr. Some R functions are reported in the appendix to preserve the visibility of codes. Additional elements and last version of the document are available from https://github.com/jsay/geoInd.

Contents

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

1 Descriptive Statistics

1.1 Data consistency

Include stat des about sample selection

```
library(sp) ; load("Inter/PolyVine.Rda")
Reg.Rank <- subset(PolyVine, PolyVine$PAOC!= 0 &
                   !is.na(PolyVine$DEM) & !is.na(PolyVine$LIBCOM))
Reg.Rank$AOCc <- ifelse(Reg.Rank$GCRU== 1, 5,
                 ifelse(Reg.Rank$PCRU== 1, 4,
                 ifelse(Reg.Rank$VILL== 1 | Reg.Rank$COMM== 1, 3,
                 ifelse(Reg.Rank$BOUR== 1, 2, 1))))
tst <- Reg.Rank@data[, 12: 17]
tst$COMM <- ifelse(tst$VILL== 1 | tst$COMM== 1, 1, 0)
tst$VILL <- 0
table(rowSums(tst), Reg.Rank$AOCc)

tmp <- Reg.Rank$LIBCOM[order(Reg.Rank$YCHF, decreasing= TRUE)]
Reg.Rank$LIBCOM <- factor(Reg.Rank$LIBCOM, levels= unique(tmp))
Reg.Rank$RAYAT <- with(Reg.Rank@data, (SOLAR- mean(SOLAR))/ sd(SOLAR))
Reg.Rank$EXPO <- cut(Reg.Rank$ASPECT,
                     breaks= c(-2, 45, 90, 135, 180, 225, 270, 315, 360))
sapply(Reg.Rank@data, function(x) sum(is.na(x)))
#table(Reg.Old$LIBCOM, Reg.Old$AOCo)
```

```
  PAR2RAS       IDU    CODECOM      AREA     PERIM   MAXDIST
        0         0          0         0         0         0
     PAOC      ALIG       BPTG      CREM      MOUS      BGOR
        0         0          0         0         0         0
     BOUR      VILL       COMM      PCRU      GCRU      XL93
        0         0          0         0         0         0
     YL93     NOMOS      URBAN    FOREST     WATER       DEM
        0         0          0         0         0         0
    SLOPE    ASPECT      SOLAR   PERMEA      CODE  NOTATION
        0         0          0         0         0         0
    DESCR TYPE_GEOL AP_LOCALE   TYPE_AP  GEOL_NAT  ISOPIQUE
        0         0         80        80         0         0
  AGE_DEB   ERA_DEB   SYS_DEB LITHOLOGIE    DURETE ENVIRONMT
        0         0         0         0        10         0
GEOCHIMIE LITHO_COM      NOUC     NO_UC  NO_ETUDE    SURFUC
        0        10       658       658       658       658
     TARG      TSAB      TLIM    TEXTAG     EPAIS       TEG
      658       658       658       658       658       658
      TMO       RUE       RUD      NOUS     OCCUP    DESCRp
      658       658       658       658       658       658
 AOC36lab  AOC36lvl   LIEUDIT    CLDVIN    LIBCOM      XCHF
       18        18       152       152       152       152
     YCHF    ALTCOM    SUPCOM    POPCOM  CODECANT    REGION
      152       152       152       152       152       152
```

```
   X          Y       AOCc      RAYAT       EXPO
   0          0          0          0          0
```

1.2 Crossing GIs dimensions

```
yop la
```

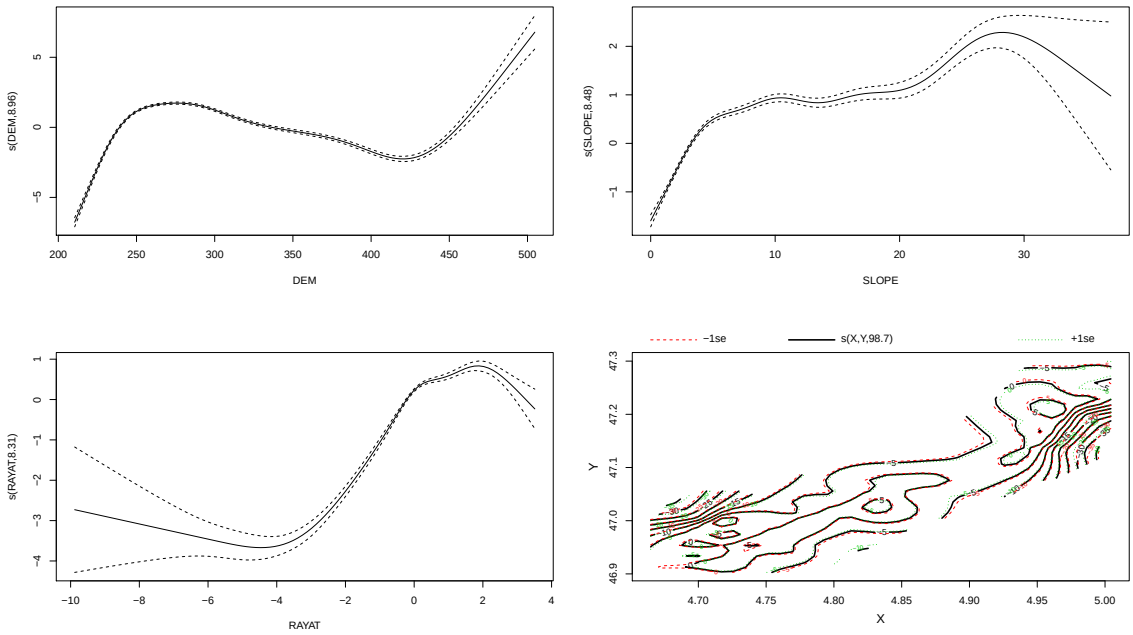

Figure 1: Effects of model XX

2 Models of GI designation

2.1 Parametric ordered logit

Benchmark parametric ordered logistic model

```
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)
```

```
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
```

Why warning message can be omitted.

2.2 Ordered generalized additive

The loop that allow to create the gamod object, the results of the models. I advice to not run the loop but to pick some value for the maximum degree of freedom and run the models individually.

```
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")
```

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

3 Diagnostics

3.1 Significance

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

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

Response: factor(AOCc)
                      LR Chisq Df Pr(>Chisq)
LIBCOM                   14625 31     <2e-16 ***
EXPO                      1212  7     <2e-16 ***
poly(DEM, 2)              5334  2     <2e-16 ***
poly(SLOPE, 2)             385  2     <2e-16 ***
poly(RAYAT, 2)            1921  2     <2e-16 ***
poly(X, 3)                2478  3     <2e-16 ***
poly(Y, 3)                 639  3     <2e-16 ***
poly(X, 3):poly(Y, 3)     9555  9     <2e-16 ***
---
codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
load("Inter/gamod.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)
```

```
          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

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

```
[1]   0.29 119.40   0.59
```

```
library(mgcv)
pcgp <- function(x){
    sum(diag(table(cut(x$line, c(-Inf, x$family$getTheta(TRUE), Inf)),
                   x$model[, 1])))/ nrow(x$model)* 100
}

rbind(sapply(gamod[ 1: 5* 2], pcgp), sapply(gamod[ 1: 5* 2], AIC))
#sapply(gamod, psR2)
```

```
       gam100   gam300   gam500   gam700   gam900
[1,]    73.89    79.94    84.23    86.94    89.15
[2,] 82412.10 64710.89 54941.54 48291.33 43535.14
```

3.3 Omitted variable

```
library(lmtest) ; library(sandwich) ; library(sure)
wal1 <- 0 ; 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))
```

```
   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 annexe.

```
load("Inter/gammod.Rda") ; source("myFcts.R")
omitVar <- function(mod, nsim= 100, old= F){
    usq <- 0
    if (!old) COM <- RRank$LIBCOM else COM <- SRank$LIBCOM
    for(i in 1: nsim) {
        if (!old) RES <- surlGAM(mod) else RES <- suroldGAM(mod)
        tmp <- lm(I(RES- mod$linear.pred)~ COM)
        usq[ i] <- waldtest(tmp, . ~ 1, vcov= vcovHC)$F[ 2]
    }
    usq
}
wal2 <- sapply(gammod, omitVar)
apply(wal2[, 1: 5* 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
```

```
library(lattice)
pltdat <- stack(data.frame(logit= wal1, wal2))
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%")})
```

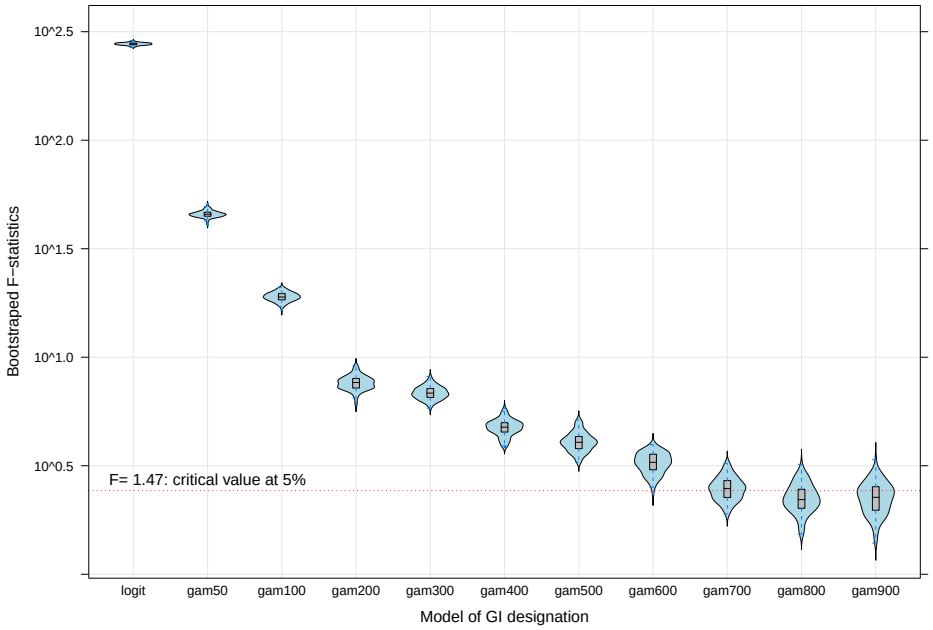

Figure 2: Effects of model XX

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= RRank@data[, .x], xlab= .x))
(atp <- autoplot(por1, what= "qq"))
do.call(grid.arrange, c(list(atp), plots))
```

Introducing `pltSURE` function.

```
restmp <- surlGAM(gamod$gam900)- gamod$gam900$line
plot(qlogis(1: nrow(RRank)/ nrow(RRank), scale= 1), sort(restmp))
abline(0, 1)
par(mfrow= c(3, 3)) ; for (i in var) pltSURE(restmp, RRank@data[, i], i)
```

4 Marginal effects

4.1 Parametric ordered logit

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

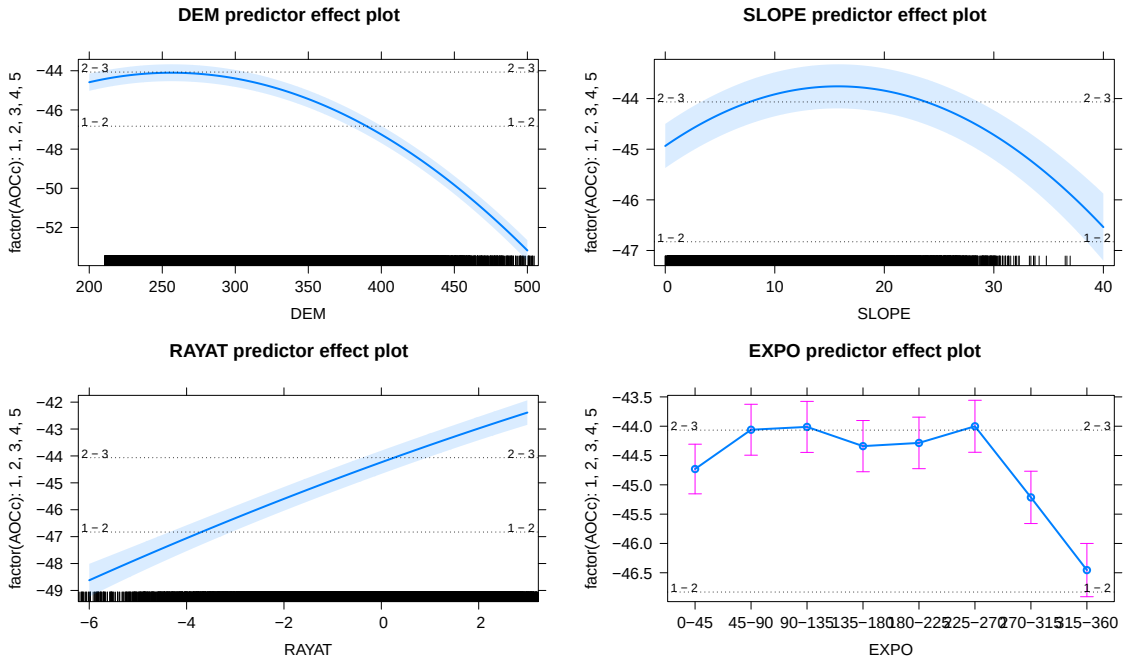

Figure 3: Effects of model XX

4.2 Ordered generalized additive

On voit bien que le lissage est le même que le papier.

```
plot(gamod$gam100, pages= 1, scale= 0)
```

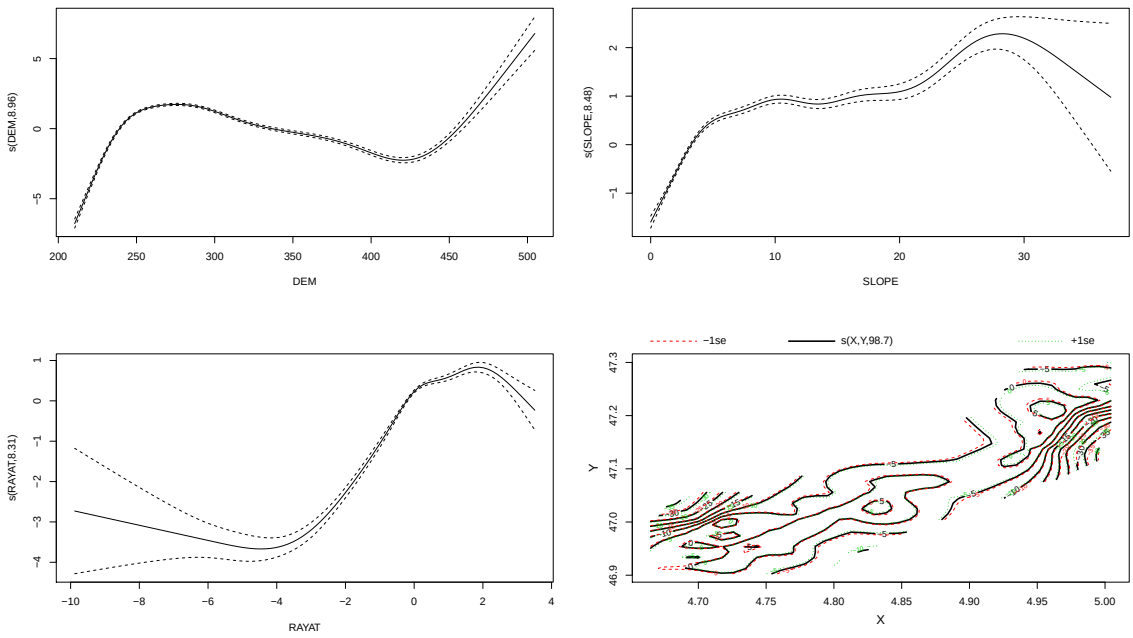

Figure 4: Effects of model XX

4.3 Ordinal superiority figure

```
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")
```

4.4 Correlation between *Communes*

```
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)
```

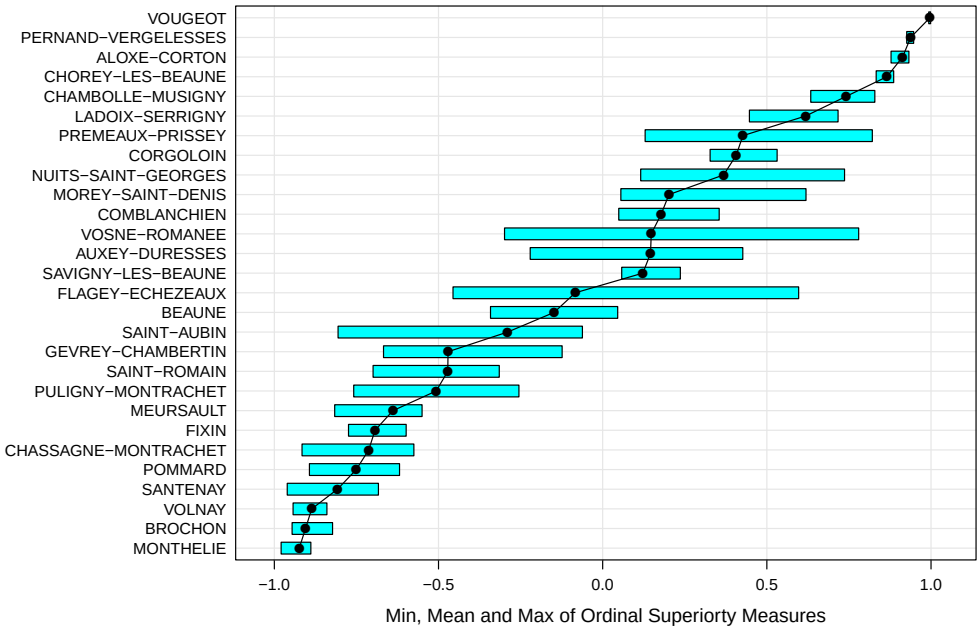

Figure 5: Effects of model XX

```
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)")
```

5 Informational content

5.1 Decomposition table

see appendix for the code of decompositions, latent un peu long à tourner.

```
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)
```

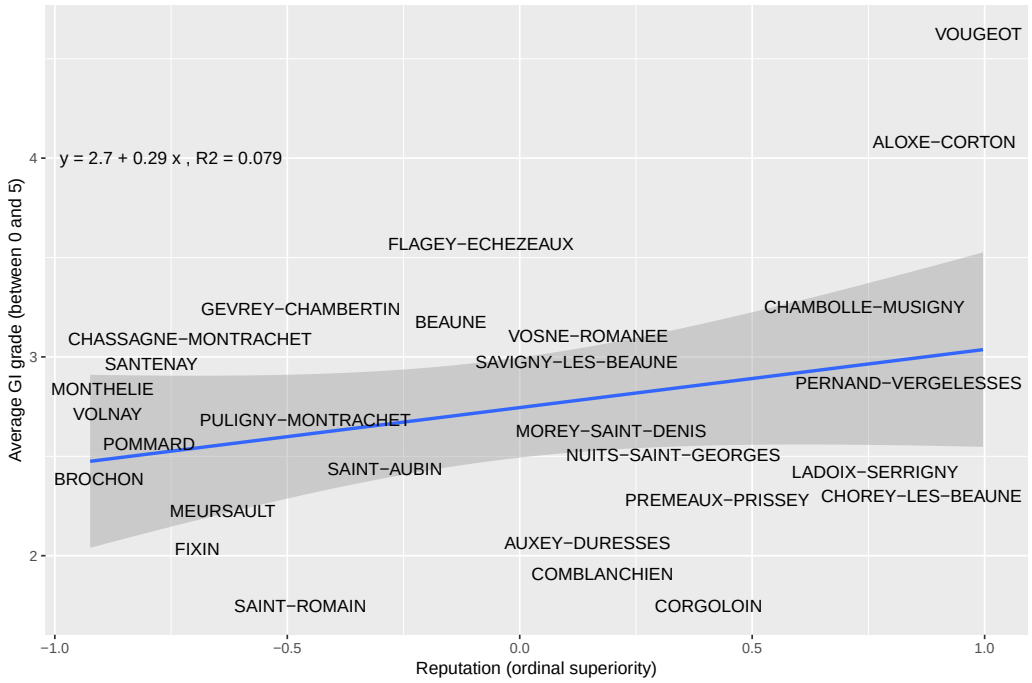

Figure 6: Effects of model XX

	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

```
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)
```

```
      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

```
library(MASS)
por2 <- polr(factor(AOCo)~ 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(AOCo)~ 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(AOCo)~ 0+ LIBCOM+ EXPO
              + 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)
gamold <- vector("list", length(listk))
system.time(
for (i in 1: length(listk)){
    gamold[[ i]] <- gam(AOCo~ 0+ LIBCOM+ EXPO+ s(DEM)+ s(SLOPE)+ s(RAYAT)
                        + s(X, Y, k= listk[ i])
                      , data= Reg.Old, family= ocat(R= 3))
})
names(gamold) <- paste0("gam", listk)
save(gamold, file= "Inter/gamold.Rda")

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

```
load("Inter/gamold.Rda")
res2a <- anova(por2, por2b)
res2 <- Anova(por2)
sapply(gamold[ 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

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

```
[1]  0.38 51.29  0.79
        gam50   gam75   gam100   gam150   gam200  gam250   gam300
[1,]    84.34    85.9    87.08    89.26    90.28    91.4    92.54
[2,] 40789.58 36833.3 33810.36 30271.01 27574.12 24526.6 22482.20
```

6.5 Omitted variable

```
library(lmtest) ; library(sandwich) ; library(sure)
wal3 <- 0 ; nsim= 100
for (i in 1: nsim){
    tmp <- surrogate(por2a)- por2a$lp
    wal3[ i] <- waldtest(lm(tmp~ Reg.Old$LIBCOM), . ~ 1, vcov= vcovHC)$F[ 2]
}
load("Inter/gammold.Rda") ; source("myFcts.R")
wal4 <- sapply(gammold, function(x) omitVar(x, old= T))
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
```

```
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%")})
```

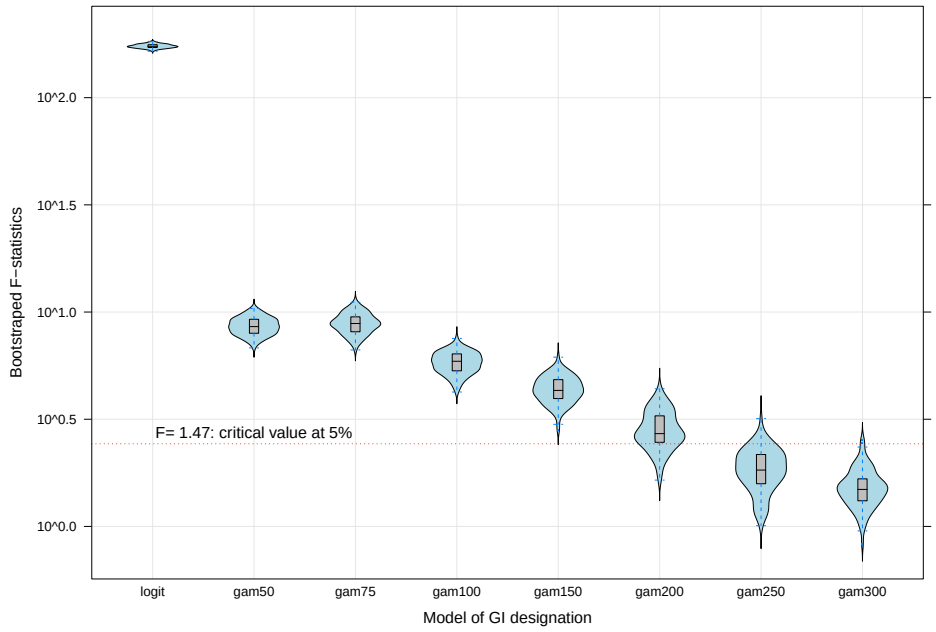

Figure 7: Effects of model XX

6.6 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(por2, what= "covariate", x= Reg.Old@data[, .x], xlab= .x))
(atp <- autoplot(por2, what= "qq"))
do.call(grid.arrange, c(list(atp), plots))
```

```
restmp <- suroldGAM(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

```
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)
```

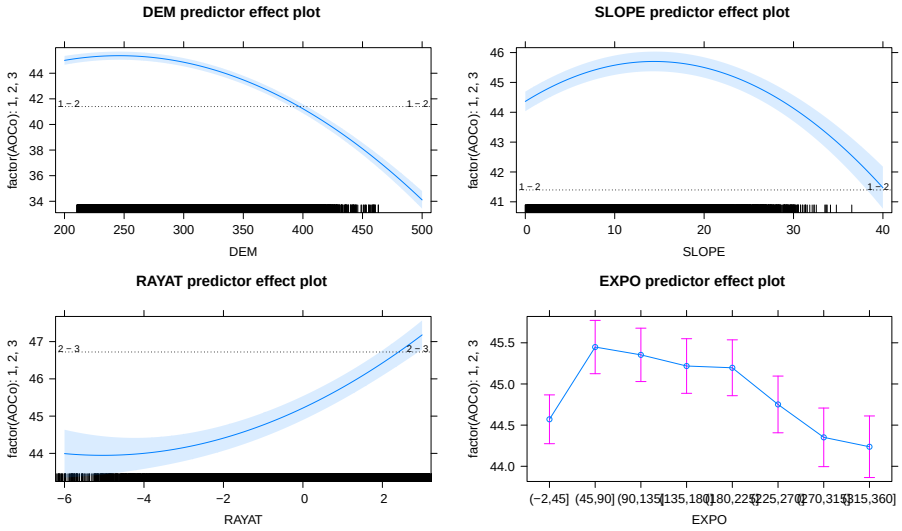

Figure 8: Effects of model XX

6.8 Ordinal superiority

```
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")
```

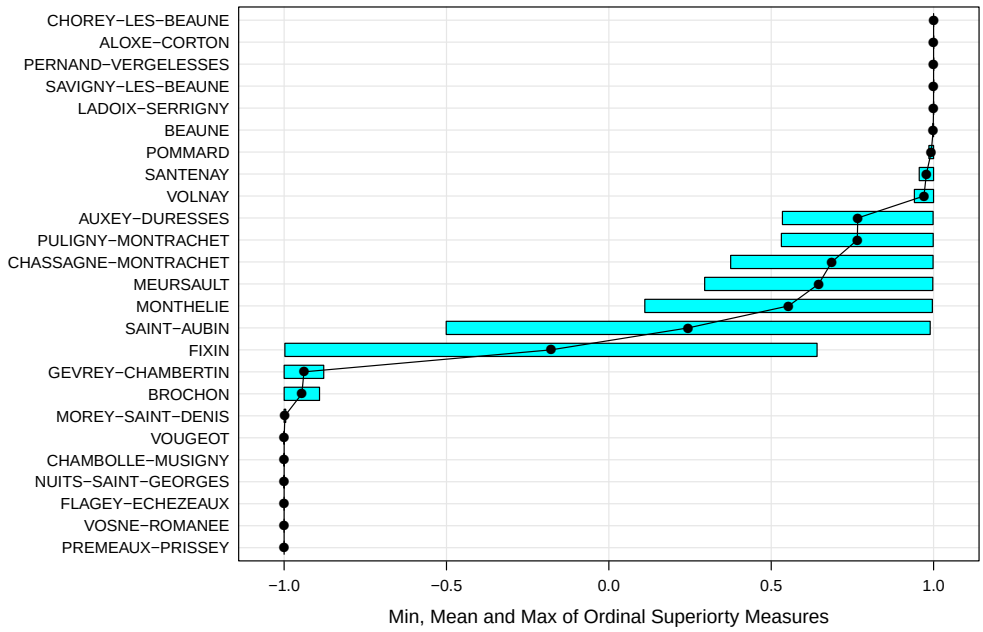

Figure 9: Effects of model XX

6.9 Correlation between models

```
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

```
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"),
```

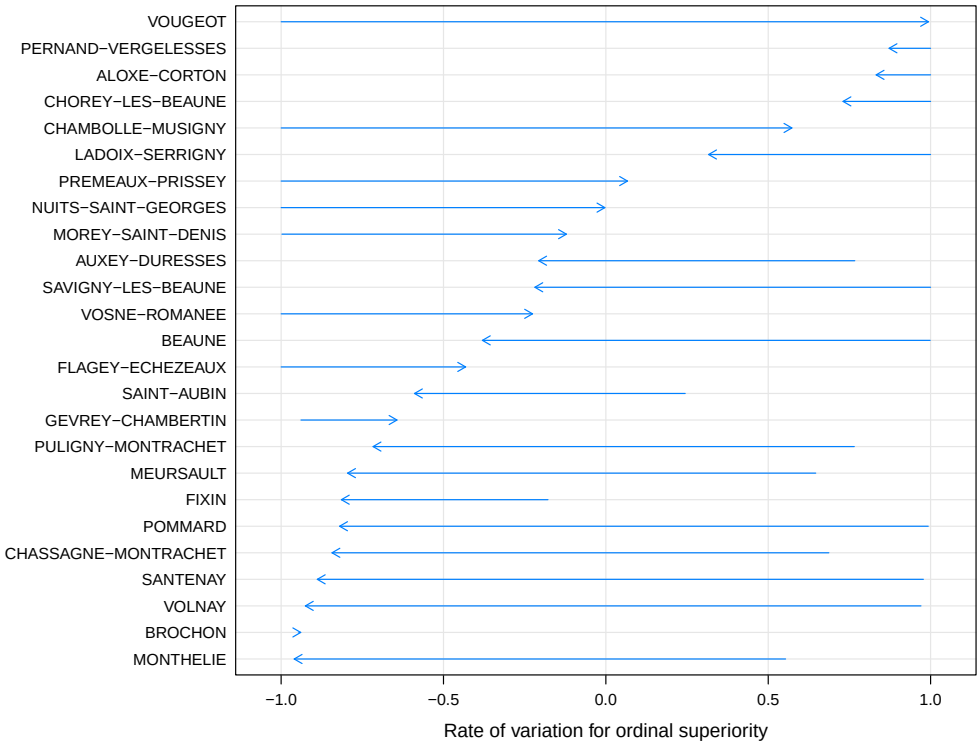

Figure 10: Effects of model XX

```
      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

```
load("Inter/gamod.Rda")
prdd <- predict(gamod$gam900, type= 'terms')
thsld <- c(-Inf, gamod$gam900$family$getTheta(TRUE), Inf)
ltt0 <- mean(prdd[, 1])+ rowSums(prdd[, -1])-
    (surlGAM(gamod$gam900)- gamod$gam900$line)
ltt1 <- rowSums(prdd)
ltt2 <- mean(prdd[, 1])+ rowSums(prdd[, -1])-
    (surlGAM(gamod$gam800)- gamod$gam800$line)
ltt3 <- mean(prdd[, 1])+ rowSums(prdd[, -1])
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

```
thrldBOUR <- mean(ltt1[RRank$AOCc== 2])
thrldVILL <- mean(ltt1[RRank$AOCc== 3])
thrldPCRU <- mean(ltt1[RRank$AOCc== 4])
Simv <- data.frame(Simu,
                   SIV= ifelse(RRank$AOCc< 2, RRank$AOCc,
                        ifelse(RRank$AOCc== 2 & ltt1< thrldBOUR, 2,
                        ifelse(RRank$AOCc== 2 & ltt1>= thrldBOUR, 3,
                               RRank$AOCc+ 1))),
                   SV = ifelse(RRank$AOCc< 3, RRank$AOCc,
                        ifelse(RRank$AOCc== 3 & ltt1< thrldVILL, 3,
                        ifelse(RRank$AOCc== 3 & ltt1>= thrldVILL, 4,
                               RRank$AOCc+ 1))),
                   SVI= ifelse(RRank$AOCc< 4, RRank$AOCc,
                        ifelse(RRank$AOCc== 4 & ltt1< thrldPCRU, 4,
                        ifelse(RRank$AOCc== 4 & ltt1>= thrldPCRU, 5,
                               RRank$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

```
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	81.1	80.7	81.2	82.8	79.2	79.7	79.0
Joint Noise	46.9	16.5	16.8	16.4	14.8	18.4	17.9	18.6
Vertical Signal	35.9	70.7	59.8	70.7	73.1	58.1	58.5	58.0
Vertical Residual	14.9	10.4	21.0	10.4	9.7	21.1	21.2	21.1
Vertical Noise	61.7	26.8	37.8	26.8	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	52.0	51.7	52.1	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 Surrogate Residuals

```
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")
}
```

1. function

```
surlOLR <- function(mod, newd= NULL){
    if (mod$method!= "logistic") stop("Logistic required")
    gg <- as.numeric(mod$zeta)
    if (is.null(newd)){
        g1 <- unname(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)))
    rtrunc(nn, spec= "logis", a= sls[, 1], b= sls[, 2],
           location= g6, scale= 1)
}
```

2. test

```
summary(por1)
```

```
library(sure)
library(truncdist)
surpOLR <- function(mod, newd= NULL){
    if (mod$method!= "probit") stop("Probit required")
    gg <- as.numeric(mod$zeta)
    if (is.null(newd)){
        g1 <- unname(as.integer(model.response(model.frame(mod))))
```

```
        g6 <- mod$lp
    } else {
        g1 <- as.integer(newd[, "AOCc"])
        g6 <- gg[ 1]-qnorm(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)))
    rtrunc(nn, spec= "norm", a= sls[, 1], b= sls[, 2],
           mean= g6, sd= sqrt(1+ var(g6)))
}
```

```
kk <- surrogate(por1)+ por1$zeta[ 1]
hh <- surpOLR(por1)
plot(kk, hh)
abline(h= gg)
abline(v= gg)
abline(0, 1, col= "blue")

ll <- surrogate(por1)+ gg[ 1]
plot(kk, ll)
abline(h= gg)
abline(v= gg)
abline(0, 1, col= "blue")

oo <- surpOLR(por1, newd= RegRank)
plot(oo, ll)
abline(h= gg)
abline(v= gg)
abline(0, 1, col= "blue")
```

```
surlGAM <- function(mod, newd= NULL){
    gg <- as.numeric(mod$family$getTheta(TRUE))
    if (is.null(newd)){
        g1 <- as.integer(mod$y)
        g6 <- mod$linear.predictors
    } else {
        g1 <- as.integer(newd[, "AOCc"])
        g6 <- predict(mod, newdata= newd)
    }
    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)))
    rtrunc(nn, spec= "logis", a= sls[, 1], b= sls[, 2], location= g6)
}
suroldGAM <- function(mod, newd= NULL){
    gg <- as.numeric(mod$family$getTheta(TRUE))
    if (is.null(newd)){
        g1 <- as.integer(mod$y)
```

```
        g6 <- mod$linear.predictors
    } else {
        g1 <- as.integer(newd[, "AOCavt"])
        g6 <- predict(mod, newdata= newd)
    }
    nn <- length(g1)
    suls <- sapply(g1, switch,
                   "1"= c(-Inf  , gg[ 1]), "2"= c(gg[ 1], gg[ 2]),
                   "3"= c(gg[ 2], Inf   ))
    sls <- data.frame(unlist(t(suls)))
    rtrunc(nn, spec= "logis", a= sls[, 1], b= sls[, 2], location= g6)
}
```

```
fit.ogam <- gam(AOCc~ poly(DEM, 2)+ poly(SLOPE, 2)
                + poly(RAYAT, 2)+ poly(ASPECT, 2)+ poly(PERMEABILITY, 2)
               , family= ocat(R= 5), data= RegRank)
fit.oglm <- polr(factor(AOCc)~ poly(DEM, 2)+ poly(SLOPE, 2)
                + poly(RAYAT, 2)+ poly(ASPECT, 2)+ poly(PERMEABILITY, 2)
               , method= "logistic", data= RegRank)
plot(fit.ogam$line, fit.oglm$lp-fit.oglm$zeta[1]- 1)
abline(0, 1)

hh <- surrogate(fit.oglm)+ fit.oglm$zeta[ 1]+ 1
gg <- surlGAM(fit.ogam)
plot(gg, hh)
abline(v= fit.ogam$family$getTheta(TRUE))
abline(h= fit.oglm$zeta+ 1)
abline(0, 1, col= "blue")
kk <- surlGAM(fit.ogam, newd= RegRank)
plot(kk, hh)
abline(v= fit.ogam$family$getTheta(TRUE))
abline(h= fit.oglm$zeta+ 1)
abline(0, 1, col= "blue")
```

---

3. function

```
surlGLM <- function(mod, newd= NULL){
    if (mod$family$link!= "logit") stop("Logit required")
    if (is.null(newd)){
        g1 <- as.integer(mod$y)
        g6 <- mod$linear.predictors
    } else {
        g1 <- as.integer(newd[, "AOCc"])
        g6 <- predict(mod, newdata= newd, type= "link")
    }
    nn <- length(g1)
    ifelse(g1== 0,
           rtrunc(nn, spec= "logis", a= -Inf, b= 0, location= g6,
                  scale= 1),
           rtrunc(nn, spec= "logis", a= 0, b=  Inf, location= g6,
                  scale= 1))
}
```

---

4. test

```
surpGLM <- function(mod, newd= NULL){
    if (mod$family$link!= "probit") stop("Probit required")
    if (is.null(newd)){
        g1 <- as.integer(mod$y)
        g6 <- mod$linear.predictors
    } else {
        g1 <- as.integer(newd[, "AOCc"])
        g6 <- predict(mod, newdata= newd, type= "link")
    }
    nn <- length(g1)
    ifelse(g1== 0, rtrunc(nn, spec= "norm", a= -Inf, b= 0, mean= g6),
           rtrunc(nn, spec= "norm", a= 0, b=  Inf, mean= g6))
}
```

A.2 Decomposition terms

We code different functions for the terms.

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

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

```
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:

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

```
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:

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

```
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:

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

```
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:

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

```
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:

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

```
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:

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

```
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:

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

```
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)
}
```