

Déterminants naturels et politiques des AOC viticoles de Côte-d'Or

MATÉRIEL ADDITIONNEL

Ce document contient les codes R qui permettent de charger les données disponibles sur le serveur <https://data.inrae.fr/> avec le package `dataverse` (Leeper, 2017) pour reproduire l'ensemble des résultats associés au *data paper* mentionné dans le titre. Les codes permettant de reproduire l'application *Shiny* (Chang et al., 2019) développée pour la visualisation des résultats sont également disponibles. Les scripts R peuvent être téléchargés sur le répertoire *GitHub* dédié: <https://github.com/jsay/geoInd/>.

1 Chargement des données

Les données et programmes utilisés dans cet article sont stockés dans l'entrepôt de données d'INRAE (<https://data.inrae.fr/>). Les fichiers apparaissent ci-dessous, ils devront être téléchargés individuellement.

```
library(dataverse) ; library(sp)
Sys.setenv("DATAVERSE_SERVER" = "data.inrae.fr")
(dataset <- get_dataset("doi:10.15454/ZZWQMN"))
```

Dataset (191764):

Version: 3.0, RELEASED

Release Date: 2019-07-17T12:34:28Z

License: NONE

17 Files:

	label	version	id	contentType
1	gammod.Rda	2	86823	application/gzip
2	gamod.Rda	1	86824	application/gzip
3	GeoRas.dbf	1	86819	application/x-dbf
4	GeoRas.prj	1	86820	application/x-esri-crs
5	GeoRas.Rda	1	86818	application/gzip
6	GeoRas.shp	1	86821	application/x-esri-shape
7	GeoRas.shx	1	86822	application/x-esri-shape
8	PolyRas.dbf	1	86830	application/x-dbf
9	PolyRas.prj	1	86827	application/x-esri-crs
10	PolyRas.Rda	1	86829	application/gzip
11	PolyRas.shp	1	86828	application/x-esri-shape
12	PolyRas.shx	1	86831	application/x-esri-shape

Les 2 premiers fichiers .Rda correspondent aux résultats des modèles statistiques présentés extensivement

dans (Ay, 2019). Les fichiers GeoRas sont les données parcellaires qui servent à estimer les modèles statistiques (le fichier .Rda peut être directement chargé dans R, les autres s’ouvrent avec un système d’information géographique). Les fichiers PolyRas contiennent les prédictions issues des modèles à l’échelle des lieux-dits (ils peuvent également s’ouvrir avec R ou un SIG). Les données téléchargées sont alors chargées avec le code ci-dessous.

```
load("GeoRas.Rda") ; dim(Geo.Ras)
```

```
[1] 110350      67
```

L’objet Geo.Ras est un objet de la classe `SpatialPolygonsDataFrame`, définie par le package `sp` de Bivand et al. (2013) qui doit être préalablement chargé. Nous constatons que la version actuelle de la base de données compte 110 350 parcelles et 67 variables. Le dictionnaire associé à ces variables est reporté dans l’Annexe 1 du *data paper* mentionné dans le titre.

2 Description des variables

2.1 Géométrie des parcelles et AOC

Les variables qui décrivent la géométrie des parcelles cadastrales et les AOC qui les caractérisent correspondent aux colonnes 2 à 16 des données préalablement chargée (voir le dictionnaire pour leurs intitulés).

```
names(Geo.Ras)[ 2: 16]
```

```
[1] "IDU"      "CODECOM" "AREA"     "PERIM"    "MAXDIST" "PAOC"
[7] "BGOR"     "BOUR"     "VILL"     "COMM"     "PCRU"     "GCRU"
[13] "AOC"      "AOCTp"    "AOC1b"
```

L’information brute issue de la superposition de la couche du parcellaire cadastral avec la couche sur les AOC actuelles est reportée dans les variables PAOC à GCRU (colonnes 6 à 12) qui prennent la valeur de 1 lorsque que le niveau AOC est revendicable sur la parcelle correspondante et 0 sinon. Les parcelles cadastrales hors du périmètre des AOC ont des valeurs manquantes pour chacune de ces 7 variables. Les trois variables suivantes (AOC, AOCTp et AOC1b) contiennent les mêmes informations INAO, mais recodées de façon plus opérationnelle pour l’analyse statistique comme mentionné dans le *data paper*.

Les modalités prises par la variable AOC1b sont une combinaison du nom des appellations et des dénominations. La variable AOCTp code cette combinaison. Les modalités correspondent souvent au nom de l’AOC maximale revendicable. Pour les Bourgognes régionaux, nous n’utilisons pas la dénomination “Bourgogne Côte d’Or”, créée en 2015, plus haute dans la hiérarchie que l’AOC Bourgogne mais peu connue du fait de sa faible antériorité. D’ailleurs, l’analyse se limite à la Côte d’Or où les délimitations “Bourgogne Côte d’Or” et “Bourgognes régionaux” sont très proches. C’est principalement pour les Premiers Crus que la variable AOC1b contient les dénominations géographiques, car l’AOC ne fait référence qu’au niveau village alors que les dénominations permettent d’identifier plus précisément les lieux-dits des parcelles.

2.2 AOC historiques

L'appariement avec les AOC de 1936 s'effectue par le centroïde des parcelles car la géométrie cadastrale actuelle ne se superpose pas parfaitement avec les polygones de la carte historique (principalement du fait de la numérisation des cartes historiques et de changements du cadastre). La faible taille des parcelles (0.2 ha en moyenne) permet de faire confiance en cette procédure d'appariement, confirmée par des vérifications manuelles. La base parcellaire est ainsi enrichie des 2 variables AOC1936lab et AOC361v1.

```
names(Geo.Ras)[ 56: 57] ; table(Geo.Ras$AOC361v1, Geo.Ras$AOC)
```

```
[1] "AOC361lab" "AOC361v1"
```

	0	1	2	3	4	5
0	47056	9832	13337	10554	593	44
3	2586	15	565	15529	8226	266
5	24	0	1	14	3	1635

2.3 Lieux-dits et variables communales

La géométrie des lieux-dits et des parcelles de l'IGN se superposant parfaitement, l'appariement avec les données parcellaires est réalisé par jointure géographique des polygones. Le code ci-dessous permet de calculer la distance de chaque parcelle au chef-lieu de la commune.

```
Geo.Ras$DISTCHF <-  
sqrt(((Geo.Ras$XL93- Geo.Ras$XCHF* 100))^2  
      + ((Geo.Ras$YL93- Geo.Ras$YCHF* 100))^2)  
names(Geo.Ras)[ 58: 66] ; summary(Geo.Ras$DISTCHF)
```

```
[1] "LIEUDIT" "CLDVIN" "LIBCOM" "XCHF" "YCHF" "ALTCOM"  
[7] "SUPCOM" "POPCOM" "CODECANT"  
Min. 1st Qu. Median Mean 3rd Qu. Max. NA's  
0 595 1049 1230 1679 6314 6425
```

Ces variables correspondent aux colonnes 58 à 66 de l'objet Geo.Ras. Comme reporté dans le résultat du code, la distance au chef lieu est distribuée entre 0 et 6,5 km avec une moyenne égale à 1,25 km.

2.4 Variables topographiques

Les résultats de l'appariement avec le modèle numérique de terrain sont reportés dans les colonnes 17 à 26 de la base Geo.Ras. Hormis les moyennes par parcelles, d'autres méthodes d'agrégation de la topographie ont été testées et n'apportent pas de différences notables.

```
names(Geo.Ras)[ c(1, 17: 26)]  
Geo.Ras$RAYAT <- (Geo.Ras$SOLAR-  
                  mean(Geo.Ras$SOLAR, na.rm= TRUE))/
```

```
sd(Geo.Ras$SOLAR, na.rm= TRUE)
table(Geo.Ras$EXPO <- cut(Geo.Ras$ASPECT,
                          breaks= c(-2, 45, 90, 135,
                                     180, 225, 270, 315, 360)))
```

```
[1] "PAR2RAS" "XL93" "YL93" "NOMOS" "URBAN" "FOREST"
[7] "WATER" "DEM" "SLOPE" "ASPECT" "SOLAR"

(-2,45] (45,90] (90,135] (135,180] (180,225] (225,270] (270,315]
 5685 19909 36291 23445 12075 5548 3350
(315,360]
 1951
```

La variable SOLAR qui contient les rayonnements solaires annuels est centrée-réduite, pour éviter les problèmes d'unité de mesure dans l'analyse statistique. Toujours pour des raisons de spécification statistique, la variable ASPECT (exposition moyenne des parcelles) est discrétisée en 8 classes d'azimuts de 45 degrés d'amplitude. Lors de cette opération, certaines parcelles n'ont pas pu être appariées car elles ne contiennent aucun centroïde de pixel suite à la rasterisation du parcellaire. Ces parcelles de très petites tailles avec des formes géométriques particulières, probablement des "erreurs" du cadastre, seront enlevées au moment de l'analyse statistique. Elles couvrent une surface cumulée de 2.7 ha, soit moins de 0.01 % de la surface totale.

2.5 Variables géologiques

La variable NOTATION issue de la couche sur les formations géologiques est une abréviation de chiffres et de lettres qui reprend la stratigraphie harmonisée (âge des formations représentées et nature des roches).

```
names(Geo.Ras)[27: 42]
Geo.Ras$NOTATION <- factor(Geo.Ras$NOTATION)
tmp <- table(Geo.Ras$NOTATION)< 1000
table(Geo.Ras$GEOL <- factor(ifelse(Geo.Ras$NOTATION %in% names(tmp[ tmp]),
                                   "0AREF",
                                   as.character(Geo.Ras$NOTATION))))
```

```
[1] "CODE" "NOTATION" "DESCR" "TYPEGEOL" "APLOCALE"
[6] "TYPEAP" "GEOLNAT" "ISOPIQUE" "AGEDEB" "ERADEB"
[11] "SYSDEB" "LITHOLOGIE" "DURETE" "ENVIRONMT" "GEOCHIMIE"
[16] "LITHOCOM"

0AREF C E Fu Fx Fy Fz GP j1-2 j3 j3a
5487 29040 2683 1653 9321 10006 7951 11181 1359 1848 3785
j3b j4a j5a j5b j6a p-IV
2887 2934 5201 5301 4827 4855
```

Dans le périmètre d'étude (qui est délimité par les frontières communales et limité aux parcelles en vignes), nous recensons 31 formations géologiques homogènes, dont la distribution spatiale et les intitulés peuvent être retrouvée à partir de la variable GEOL. Les parcelles non appariées, produisant des valeurs manquantes sur ces variables géologiques, sont peu nombreuses (entre 31 et 862 parcelles selon les variables), elles seront enlevées au moment de l'analyse statistique sans conséquence sur les résultats.

2.6 Variables pédologiques

Les colonnes 43 à 55 contiennent les variables pédologiques issues du référentiel régional (voir les précisions dans le *data paper*). Nous les recodons pour ne garder que les modalités avec plus de 1000 parcelles.

```
names(Geo.Ras)[43: 55]
Geo.Ras$NOUC <- factor(Geo.Ras$NOUC)
tmp <- table(Geo.Ras$NOUC)< 1000
table(Geo.Ras$PEDO <- factor(
  ifelse(Geo.Ras$NOUC %in% names(tmp[tmp]), "0AREF",
    as.character(Geo.Ras$NOUC))))
```

```
[1] "NOUC" "SURFUC" "TARG" "TSAB" "TLIM" "TEXTAG" "EPAIS"
[8] "TEG" "TMO" "RUE" "RUD" "OCCUP" "DESCRp"

0AREF 10 13 14 26 27 28 29 30 32 34
3265 2074 3770 23472 4750 1348 11641 7636 6983 3072 2469
35 36 38 5 61 69 7 8
8356 1602 2198 4767 1605 2116 1445 3136
```

Les valeurs manquantes, associées aux parcelles non couvertes par la couche pédologique, sont assez nombreuses, elles couvrent environ 4,25 % de la surface du périmètre étudié. Les parcelles non couvertes sont, en revanche, peu désignées en AOC car moins de 1 % des AOC ont des variables pédologiques manquantes. Les valeurs manquantes sont donc dans de rares cas des parcelles de vignes et ce sont principalement des parcelles bâties au coeur des villages. Une explication intuitive de ces valeurs manquantes est l'absence de données pédologiques sur les sols artificialisés, cela étant corroborée par une vérification manuelle. La faible précision spatiale des données pédologiques peut s'illustrer par comparaison avec les variables du MOS sur l'artificialisation. Les UCS avec les variables pédologiques manquantes regroupent des occupations du sol très différentes. Parmi les 33 modalités présentes initialement dans les UCS, seules 19 sont retenues les autres sont regroupées dans une modalité de référence 0AREF.

2.7 Statistiques descriptives

Les données issues des 7 sources présentées ci-dessus sont donc compilées dans une base unique. Le code suivant effectue les derniers traitements, à savoir la conversion du système de projection du Lambert93 vers le WGS84 utilisé pour l'application Shiny, la suppression des valeurs manquantes sur certaines variables, le codage des indicatrices (pour les AOC et l'exposition), et la normalisation des unités de mesure pour les variables continues. L'objet `tb.lab` qui contient l'intitulé des variables, nécessaire dans le code ci-dessous, est défini en Annexe 1 de ce Matériel Additionnel. Le package `stargazer` doit être chargé en préalable pour reproduire la Table 1 du *data paper* en format latex, d'autres formats sont disponibles.

```
GR84 <- spTransform(Geo.Ras, CRS("+proj=longlat +ellps=WGS84"))
dd <- coordinates(GR84) ;
Geo.Ras$X= dd[, 1] ; Geo.Ras$Y= dd[, 2]
Reg.Ras <- subset(Geo.Ras,
  !is.na(AOC1b) & !is.na(DEM) & !is.na(DESCR) & !is.na(RUD)
  & !is.na(AOC361ab) & !is.na(REGION))
Stat.Ras <- data.frame(Reg.Ras@data, model.matrix(~0+ factor(Reg.Ras$AOC)),
```

```

                                model.matrix(~ 0+ factor(Reg.Ras$EXPO))
names(Stat.Ras)[75: 79] <- paste0("AOC", 1: 5)
names(Stat.Ras)[80: 87] <- paste0("EXPO", 1: 8)
Stat.Ras$AREA <- Reg.Ras$AREA/ 1000 ; Stat.Ras$DEM <- Reg.Ras$DEM/ 1000
Stat.Ras$SOLAR <- Reg.Ras$SOLAR/ 1000000 ; library(stargazer)
stargazer(Stat.Ras[, names(tb.lab)], covariate.labels=tb.lab,
          float= F, font.size= "small", column.sep.width= "0pt",
          digit.separate= c(0, 3), type= "latex")

```

3 Modélisation

Nous présentons ici le chargement des résultats issus de la modélisation statistique ainsi que les manière de reproduire ces résultats. L'objet `gamod.Rda` téléchargeable sur de *dataverse* <https://data.inrae.fr/> contient 10 modèles ordonnés généralement additifs pour la hiérarchie des AOC actuelles. Ils sont présentés du moins ajusté `gam50` (c'est-à-dire le plus lissé spatialement, voir le texte) au plus ajusté `gam900`.

```

GamModRaw <- get_file("gamod.Rda", "https://doi.org/10.15454/ZZWQMN")
writeBin(GamModRaw, "gamod.Rda") ; load("gamod.Rda") ; names(gamod)

```

3.1 Estimation

Pour la reproductibilité des analyses, nous reportons ci-dessous le code pour l'estimation du modèle qui sera utilisé dans l'application, celui qui s'ajuste le mieux aux données et qui présente les meilleures prédictions. La localisation des parcelles est ajustée avec des fonctions splines cubiques pour un nombre maximal de degré de liberté effectifs de 900. L'algorithme itératif des moindres carrés pondérés pénalisés est relativement long à effectuer : environ 9 heures avec un processeur Intel Core i7-7820HQ CPU 2.90 GHz x8 et 64 Go de RAM. Le lecteur peut accéder directement au résultat de cette estimation par l'objet `gamod$gam900` téléchargé précédemment sur le serveur. En augmentant le nombre maximal de degrés de liberté effectifs noté dans la fonction `gam`, le modèle va s'ajuster plus finement aux variations locales des AOC pour prendre en compte les effets non observables.

```

## 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))
## )

```

```

library(mgcv) ; anova(gamod$gam900)

```

```

Family: Ordered Categorical(-1,5.34,14.01,20.99)
Link function: identity

```

Formula:

```

AOC ~ 0 + LIBCOM + EXPO + GEOL + PEDO + s(DEM) + s(SLOPE) + s(RAYAT) +

```

```
s(X, Y, k = 900)
```

Parametric Terms:

	df	Chi.sq	p-value
LIBCOM	31	1363	<2e-16
EXPO	7	131	<2e-16
GEOL	14	441	<2e-16
PEDO	13	388	<2e-16

Approximate significance of smooth terms:

	edf	Ref.df	Chi.sq	p-value
s(DEM)	8.81	8.98	867	<2e-16
s(SLOPE)	7.72	8.61	190	<2e-16
s(RAYAT)	7.33	8.38	531	<2e-16
s(X,Y)	841.42	870.01	86597	<2e-16

Nous obtenons avec la fonction `anova` la significativité statistique des différentes variables incluses dans le modèle au regard des statistiques de χ^2 . Les variables indicatrices et les effets fixes sont dans la partie paramétrique (Parametric Terms, soit le premier tableau de résultat) alors que les variables continues sont reportées dans le second tableau (Approximate significance of smooth terms). Les mêmes statistiques peuvent être obtenues pour les autres modèles présents dans l'objet `gamod`, moins lissés spatialement. Les résultats sur la significativité des variables et la forme des effets sont globalement robustes à l'ajustement des coordonnées géographiques. Nous pouvons également faire apparaître les capacités prédictives du modèle.

```
sum(diag(table(cut(gamod$gam900$line,
                  c(-Inf, gamod$gam900$family$getTheta(TRUE), Inf),
                  gamod$gam900$model[, 1])))/ nrow(gamod$gam900$model)*100
1- (logLik(gamod$gam900)/ logLik(update(gamod$gam900, . ~ + 1)))
```

```
[1] 89.48
'log Lik.' 0.7565 (df=964)
```

La Figure 2 du *data paper* peut être obtenue simplement par la fonction `plot` du package `mgcv` avec la commande `plot(gamod$gam900, page=1, scale=0)`.

3.2 Prédiction

Le code suivant permet de reproduire la Figure 3 du *data paper*. Il nécessite les objets `plogi` et `mso.key` qui sont définis dans l'Annexe de ce Matériel Supplémentaire.

```
library(latticeExtra) ; resum900 <- summary(gamod$gam900)
cf <- resum900$p.coeff[ 4: 31]- mean(resum900$p.coeff[ 4: 31])
dat.fig <- data.frame(LIBCOM=substr(names(gamod$gam900$coef[ 4: 31]),7,30),
                     REGION= c(rep("tomato", 12), rep("chartreuse", 16)),
                     OS= 2* plogi(cf)- 1,
                     OSi= 2* plogi(cf- 1.5* resum900$se[ 4: 31])- 1,
```

```

        OSa= 2* plogi(cf+ 1.5* resum900$se[ 4: 31])- 1)
segplot(reorder(factor(LIBCOM), OS)~ OSi+ OSa,
        length= 5, draw.bands= T, key= mso.key,
        data= dat.fig[order(dat.fig$OS), ], center= OS, type= "o",
        col= as.character(dat.fig$REGION[order(dat.fig$OS)]),
        unit = "mm", axis = axis.grid, col.symbol= "black", cex= 1,
        xlab= "Mesure de supériorité ordinale et intervalles à 10 %")

```

Le code suivant présente le calcul des prédictions et leur normalisation pour qu'elles soient distribuées entre 0 et 100 (avec la fonction `unini`), pour l'ensemble des parcelles de la base `Prd.Ras`. Notons que la ligne sur les prédictions, commentée ci-dessous, est assez longue à évaluer dans R (5 minutes).

```

Prd.Ras <- subset(Geo.Ras, !is.na(AOC1b))
Prd.Ras$GEOL <- ifelse(Prd.Ras$NOTATION%in%levels(gamod$gam900$model$GEOL),
        as.character(Prd.Ras$NOTATION), "0AREF")
Prd.Ras$PEDO <- ifelse(Prd.Ras$NOUC %in% levels(gamod$gam900$model$PEDO),
        as.character(Prd.Ras$NOUC), "0AREF")
## prd <- predict(gamod$gam900, newdata= Prd.Ras@data, type= "terms")
Prd.Ras$LTraw <- rowSums(prd, na.rm= TRUE)
Prd.Ras$LTcor <- mean(prd[, 1], na.rm= T)+ rowSums(prd[, -1], na.rm= T)
unini <- function(x) (x- min(x))/ (max(x)- min(x))
Prd.Ras$UFraw <- round(unini(Prd.Ras$LTraw)* 100, 2)
Prd.Ras$UFcor <- round(unini(Prd.Ras$LTcor)* 100, 2)
lapply(Prd.Ras@data[, c("UFraw", "UFcor")], summary)

```

```

$UFraw
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.0    67.0    72.5    70.6    75.6    100.0

```

```

$UFcor
  Min. 1st Qu.  Median Mean 3rd Qu.    Max.
  0.0 67.4 71.7 70.8 76.6 100.0

```

Nous obtenons donc des prédictions normalisées disponibles avec (`UFcor`) et sans (`UFraw`) la correction des effets politiques associés aux communes afin de pouvoir consulter le classement corrigé et apprécier le niveau des biais communaux dans la classification actuelle des AOC.

4 Application

Nous présentons ici...

4.1 Mise en forme

Nous recodons tout d'abord les intitulés des parcelles afin qu'ils constituent un identifiant unique de localisation : noms de lieux-dits pour les niveaux Coteaux Bourguignons, Bourgogne régional, et Village, dénominations retravaillées pour les Premiers crus et appellations pour les Grands Crus.

```

NVA <- c("Coteaux b.", "Bourgogne", "Village", "Premier cru", "Grand cru")
names(NVA) <- 1: 5 ; library(plyr)
Prd.Ras$NIVEAU <- as.character(revalue(factor(Prd.Ras$AOC), NVA))
Prd.Ras$NAME <- ifelse(Prd.Ras$AOC== 5, as.character(Prd.Ras$AOC1b),
                      ifelse(Prd.Ras$AOC< 4, as.character(Prd.Ras$LIEUDIT), NA))
for (i in 1: nrow(Prd.Ras)){
  if (is.na(Prd.Ras$NAME[ i])){
    Prd.Ras$NAME[ i] <- substr(Prd.Ras$AOC1b[ i],
                              regexpr(" cru+", Prd.Ras$AOC1b[ i],
                                        perl= T)+ 5,
                              nchar(as.character(Prd.Ras$AOC1b[ i])))
  } else {(Prd.Ras$NAME[ i])}
}
Prd.Ras$Concat <- paste0(Prd.Ras$AOC, Prd.Ras$LIBCOM, Prd.Ras$NAME)
length(unique(Prd.Ras$Concat))

```

[1] 2391

Ainsi, à partir des parcelles cadastrales utilisées pour estimer le modèle statistique, nous obtenons environ 2400 localités soit une moyenne de 25 parcelles par localité.

Nous allons désormais fusionner la géographie des parcelles selon la variable `Concat` tout juste créée pour agréger les scores prédits. Les scores alors reportés au niveau des nouvelles localités seront calculés par moyennes pondérées par la surface de chaque parcelle qui les compose. Nous calculons également la position de chaque localité dans la hiérarchie continue issue de la modélisation par rapport à l'ensemble des localités de la zone (avec les variables `RANG_br` et `RANG_co`), ce qui permet de présenter, en sortie du code ci-dessous, les 10 localités les mieux notées sur la base des scores corrigés.

```

library(data.table) ; Prd.Dtb <- data.table(Prd.Ras@data)
Dat.Ldt <- Prd.Dtb[, .(LIBCOM= LIBCOM[ 1], NOM= NAME[ 1],
                    NIVEAU= NIVEAU[ 1],
                    SURFACE_ha= round(sum(AREA)/ 1e4, 2),
                    SCORE_brut= round(weighted.mean(UFraw, AREA), 2),
                    SCORE_corrige=round(weighted.mean(UFcor, AREA), 2)), by= Concat]
library(rgdal) ; library(rgeos) ; library(maptools)
tmp_geo <- gBuffer(Prd.Ras, byid= TRUE, width= 0)
Poly.ldt <- unionSpatialPolygons(tmp_geo, Prd.Ras$Concat)
Poly.ldt$Concat <- as.character(row.names(Poly.ldt))
Poly.Ras <- merge(Poly.ldt, Dat.Ldt, by= "Concat")
Poly.Ras$RANG_brut<- round(rank(Poly.Ras$SCORE_brut)/ nrow(Poly.Ras)*100,2)
Poly.Ras$RANG_corrige <- round(rank(Poly.Ras$SCORE_corrige)/
                               nrow(Poly.Ras)*100,2)
head(Poly.Ras@data[order(Poly.Ras$RANG_corrige, decreasing= T), c(3, 4, 6, 7)], n= 10)
Poly.Ras$NIVEAU <- factor(Poly.Ras$NIVEAU, levels= NVA)

```

	NOM	NIVEAU	SCORE_brut	SCORE_corrige
2364	Chambertin	Grand cru	94.22	94.11
2363	Grands-Echezeaux	Grand cru	87.73	90.76
2384	Montrachet	Grand cru	88.72	90.69
2381	Bâtard-Montrachet	Grand cru	87.73	89.68
2361	Montrachet	Grand cru	87.05	89.58

2362	Echezeaux	Grand cru	86.13	89.12
2369	Latricières-Chambertin	Grand cru	88.73	88.53
2371	Mazoyères-Chambertin	Grand cru	88.71	88.50
2359	Bâtard-Montrachet	Grand cru	85.80	88.30
2010	La Combe d'Orveau	Premier cru	91.01	87.83

Nous enregistrons ensuite les résultats agrégés de la prédiction dans un objet de type `sf` défini par le package du même nom (Pebesma, 2018). Cette objet nommé `Poly.Ras` pourra directement être utilisée dans l'application *Shiny* pour consulter, lancer, ou modifier le classement statistique que nous avons obtenu. Les résultats issus du recodage des dénominations et de l'agrégation des scores prédits sont accessibles sur le serveur data de l'INRA à l'adresse <https://data.inrae.fr/>, qui peut être aussi chargée comme précédemment avec le package `dataverse`.

```
library(sf) ; Poly.Ras <- st_as_sf(Poly.Ras)
Poly.Ras <- st_transform(Poly.Ras, crs= 4326)
save(Poly.Ras, file= "Inter/PolyRas.Rda")
st_write(Poly.Ras, "/PolyRas.shp", delete_layer= TRUE)
```

Toutes les données pour lancer l'application cartographique sont désormais construites. Nous allons désormais présenter la procédure à suivre pour consulter les résultats.

4.2 Lancement de l'application

Une première étape pour lancer l'application localement consiste à définir une carte dynamique, de type Leaflet, grâce au package `mapview` (Appelhans et al., 2018), qui sera ensuite intégrée à l'application. Cela nécessite la présence de l'objet `Poly.Ras`, issu des traitements précédents, dans le répertoire de travail.

```
library(RColorBrewer) ; library(mapview) ; library(sf)
Poly.Ras <- st_read("Inter/PolyRas.shp")
Poly.Ras$NIVEAU <-
  factor(Poly.Ras$NIVEAU, levels= c("Coteaux b.", "Bourgogne", "Village",
                                   "Premier cru", "Grand cru"))
AocPal <- brewer.pal(5, "BuPu")
mapviewOptions(basemaps= c("Esri.WorldImagery", "OpenStreetMap",
                           "OpenTopoMap", "CartoDB.Positron"),
               raster.palette= colorRampPalette(brewer.pal(9, "Greys")),
               vector.palette= colorRampPalette(brewer.pal(9, "YlGnBu")),
               na.color= "magenta", layers.control.pos = "topleft")
map <- mapview(Poly.Ras, zcol= "NIVEAU", label= Poly.Ras$NOM,
               layerId= Poly.Ras$Concat, alpha.regions= .5,
               col.regions = AocPal, color= "white", legend.opacity= .5,
               popup = popupTable(Poly.Ras,
                                   feature.id=F, zcol=names(Poly.Ras)[ -1]))
```

L'objet `map` ainsi créé permet de faire apparaître sur un navigateur internet une carte dynamique pour visualiser les différentes parcelles viticoles de la zone et faire apparaître le score (corrigé des effets communaux ou pas) suite à un clic sur une parcelle donnée. Il apparaît alors le nom de la commune d'appartenance (LIBCOM), le nom du lieu-dit (NOM), le niveau dans la hiérarchie des AOC (NIVEAU), la surface du lieu-dit (SURFACE), le score brut (SCORE_b) et corrigé (SCORE_c), ainsi que la position du lieu dit dans la hiérarchie générale de la région (RANG_br pour la version brute et RANG_cr pour la version corrigée).

Une fois la carte interactive `map` créée, le code précédent et le code suivant doivent être enregistrés dans un fichier `global.R` en accord avec la structuration habituelle des applications Shiny (Chang et al., 2019). Les deux autres scripts `ui.R` et `server.R` nécessaires pour lancer localement l'application Shiny sont reportés en Annexes 2 et 3 de ce document. Ils contiennent respectivement le paramétrage de l'interface utilisateur et les calculs qui sont effectués sur le serveur pour l'interactivité de l'application. Le code ci-dessous source ces deux fichiers qui sont également disponibles sur un répertoire distant <https://github.com/jsay/geoInd/>.

```
library(shiny) ; library(shinydashboard) ; library(shinyjs)
library(leaflet) ; library(maptools) ; library(ggplot2) ; library(markdown)
Pts.Crd <- st_centroid(Poly.Ras)
source("ui.R") ; source("server.R")
enableBookmarking(store = "url")
shinyApp(ui, server)
```

La commande `shinyApp(ui, server)` lance l'application dans le navigateur internet par défaut, et permet d'obtenir localement le même résultat que la version en ligne de l'application.

References

- Ay, J.-S. (2019). The informational content of geographical indications. *en révision* .
- Bivand, R. S., Pebesma, E. and Gomez-Rubio, V. (2013). *Applied spatial data analysis with R, Second edition*. Springer, NY.
- Chang, W., Cheng, J., Allaire, J., Xie, Y. and McPherson, J. (2019). shiny: Web Application Framework for R. R package version 1.4.0.
- Leeper, T. J. (2017). dataverse: R Client for Dataverse 4. R package version 0.2.0.

A Annexes

A.1 Codes additionnels

```
tb.lab <-  
  c(AREA= "Surface [1000 m$^2$]", DEM= "Altitude [1000 m]",  
    SLOPE= "Pente [degrés]", SOLAR= "Radiation solaire [millions J]",  
    X= "Longitude [degrés]", Y= "Latitude [degrés]",  
    AOC1= "Niveau AOC Coteaux", AOC2= "Niveau AOC Régional",  
    AOC3= "Niveau AOC Village", AOC4= "Niveau AOC Premier Cru",  
    AOC5= "Niveau AOC Grand Cru",  
    EXPO1= "Exposition [$0-45$]" , EXPO2= "Exposition [$45-90$]",  
    EXPO3= "Exposition [$90-135$]" , EXPO4= "Exposition [$135-180$]",  
    EXPO5= "Exposition [$180-225$]", EXPO6= "Exposition [$225-270$]",  
    EXPO7= "Exposition [$270-315$]", EXPO8= "Exposition [$315-360$]")  
## Pour Figure 4  
plogi <- function(x) exp(x/ sqrt(2))/ (1+ exp(x/ sqrt(2)))  
mso.key <- list(x = .35, y = .95, corner = c(1, 1),  
              text = list(c("Côte de Beaune", "Côte de Nuits")),  
              rectangle = list(col = c("chartreuse", "tomato")))
```

A.2 Fichier ui.R

```
ui <- dashboardPage(
  dashboardHeader(
    titleWidth= 550,
    title= "Classification statistique des vignobles de la Côte d'Or"),
  dashboardSidebar(disable = TRUE),
  dashboardBody(
    fluidRow(
      box(width= 4,
        column(width= 12,
          selectInput(
            "niveau", label= "Niveau de l'appellation",
            choices=
              c(as.character(unique(Poly.Ras$NIVEAU)),
                "TOUS"),
            selected= 1),
          selectInput(
            "commune",
            label= "Commune de la parcelle",
            choices= c(
              as.character(unique(Poly.Ras$LIBCOM)),
              "TOUTES"), selected= 1),
          selectInput(
            "nom",
            label= "Lieu dit de la parcelle",
            choices= c(
              as.character(unique(Poly.Ras$NOM)),
              "TOUS"), selected = 1),
          plotOutput("miplot", height= 510, width= 200)),
      box(width= 7,
        column(width = 12,
          leafletOutput("mymap", height= 745),
          fluidRow(verbatimTextOutput("mymap_shape_click"))
        )
      )
    )
  )
)
```

A.3 Fichier server.R

```
server <- function(input, output, session) {
  ## Reactive values
  values <- reactiveValues(niveau= NULL, commune= NULL, nom= NULL)
  ## Initialisation reactive values
  observe({
    if (is.null(values$niveau)) values$niveau <- input$niveau
    if (is.null(values$commune)) values$commune <- input$commune
    if (is.null(values$nom)) values$nom <- input$nom
  })
  ## MAJ des reactive values apres un click sur un polygone
  observeEvent(input$mymap_shape_click, {
    values$niveau <- Pts.Crd$NIVEAU[Pts.Crd$Concat==
      input$mymap_shape_click$id]
    values$nom <- Pts.Crd$NOM[Pts.Crd$Concat==
      input$mymap_shape_click$id]
    values$commune <- Pts.Crd$LIBCOM[Pts.Crd$Concat==
      input$mymap_shape_click$id]
  })
  ## MAJ des reactive values apres un choix dans menus deroulants
  observeEvent(c(input$commune, input$niveau, input$nom), {
    if (values$niveau != input$niveau) {
      values$niveau <- input$niveau
      values$commune <- Pts.Crd$LIBCOM[Pts.Crd$NIVEAU==
        values$niveau][ 1]
      values$nom <- Pts.Crd$NOM[Pts.Crd$LIBCOM==
        values$commune][ 1]
    }
    else if (values$commune != input$commune) {
      values$commune <- input$commune
      values$nom <- Pts.Crd$NOM[Pts.Crd$LIBCOM== values$commune][ 1]
    }
    else if (values$nom != input$nom) {
      values$nom <- input$nom
    }
  })
  ## MAJ menus deroulants
  observeEvent(c(values$commune, values$niveau, values$nom), {
    updateSelectInput(session, "niveau",
      choices= c(as.character(
        unique(Poly.Ras$NIVEAU))),
      selected=values$niveau)
    updateSelectInput(session, "commune",
      choices= c(as.character(
        unique(Poly.Ras$LIBCOM[Poly.Ras$NIVEAU %in%
          values$niveau]))),
      selected=values$commune)
    updateSelectInput(session, "nom",
      choices= c(as.character(
        unique(Poly.Ras$NOM[Poly.Ras$LIBCOM %in%
          values$commune &
          Poly.Ras$NIVEAU %in%
          values$niveau]))),
      selected=values$nom)
  })
  ## Subset donnees
  getPts <- reactive({
```

```

    Pts.Crd[Pts.Crd$NIVEAU %in% values$niveau &
      Pts.Crd$LIBCOM %in% values$commune &
      Pts.Crd$NOM %in% values$nom, ]})
## Carte de base
output$mymap <- renderLeaflet({
  map@map
})
## Rafraichissement carte
observe({
  gg <- getPts()
  if (nrow(gg)== 0) return(NULL)
  else {
    bound_box <- as.numeric(st_bbox(Poly.Ras[Poly.Ras$Concat %in%
      gg$Concat,]))

    leafletProxy("mymap") %>%
      clearMarkers() %>%
      fitBounds(lng1= bound_box[ 3], lng2= bound_box[ 1],
        lat1= bound_box[ 4], lat2= bound_box[ 2]) %>%
      addCircleMarkers(data= (getPts()))}
})
## Violon Plot de base
output$miplot <- renderPlot({
  yop <- getPts()$SCORE_c
  if (length(yop)==0) return(NULL)
  top <- round(100-
    aggregate(I(Poly.Ras$SCORE_c< yop)* 100,
      by= list(Poly.Ras$NIVEAU), mean)[, 2])
  ggplot(Poly.Ras, aes(x= factor(NIVEAU),
    y= SCORE_c, fill= factor(NIVEAU)))+
  geom_violin(trim= FALSE)+ theme_minimal()+ ylim(40, 100)+
  geom_boxplot(width=0.1, fill= "white")+
  annotate("text", x= 1: 5, y= 100,
    label= paste("", top, "%"), col= "red", size= 5)+
  labs(title= "Comparaison avec les autres parcelles",
    x= "",
    y = "Niveau sur une échelle de 1 à 100")+
  scale_fill_manual(values= AocPal)+
  theme(legend.position= "none",
    plot.title = element_text(hjust = 0, size = 16),
    axis.text.x = element_text(size= 12),
    axis.title.x = element_text(hjust= 0, size= 14),
    axis.title.y = element_text(size= 14))+
  scale_x_discrete(expand= expand_scale(mult= 0, add= 1),
    drop= T)+
  geom_hline(yintercept= yop, lty= 2, col= "red")+
  annotate("text", x= 0.35, y= yop+ 2,
    label= round(yop, 2), col= "red", size= 5)
}, height = 500, width = 400)}

```
