

# Déterminants biophysiques des AOC viticoles : Construction des données et modélisation

JEAN-SAUVEUR AY                      MOHAMED HILAL  
< [jean-sauveur.ay@inra.fr](mailto:jean-sauveur.ay@inra.fr) >      < [mohamed.hilal@inra.fr](mailto:mohamed.hilal@inra.fr) >

Unité Mixte de Recherche CESAER  
AgroSup / INRA / Univ. Bourgogne Franche-Comté  
26 boulevard du docteur Petitjean 21000 DIJON

*Data paper* version 0.2 du Vendredi 12 avril 2019

## Résumé

Nous présentons la construction de données à l'échelle parcellaire pour étudier statistiquement les relations entre les caractéristiques biophysiques (topographie, géologie, pédologie) des parcelles viticoles et les appellations d'origine contrôlée (AOC). La zone d'étude comprend les 31 communes de la Côte d'Or qui forment les Côtes de Beaune et de Nuits. L'intérêt de ces données est illustré par une modélisation de la hiérarchie des AOC qui permet de noter sur 100 chaque parcelle de vigne en accord avec les principes historiques de désignation. Les données et prédictions du modèle sont disponibles sous licence Creative Commons XX sur <https://data.inra.fr/geoInd> et sont consultables par le biais d'une application Shiny sur <http://github.com/jsay/geoInd/>.

**Mots-clés:** Recherche reproductible ; économie viti-vinicole ; signes de qualité ; système d'information géographique ; modélisation économétrique.

## Table des Matières

|          |                                  |          |          |  |           |
|----------|----------------------------------|----------|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>              | <b>2</b> | <b>4</b> | <b>Modèle ordonné de désignation</b>   | <b>9</b>  |
|          |                                  |          | 4.1      | Variable transformations               | 9         |
| <b>2</b> | <b>Construction des données</b>  | <b>3</b> | 4.2      | Spécification du modèle                | 10        |
| 2.1      | Du cadastre aux AOC              | 3        | 4.3      | Effets des variables biophysiques      | 11        |
| 2.2      | Enrichissement de la topographie | 4        | 4.4      | Prédiction du score et classifications | 11        |
| 2.3      | Enrichissement de la géologie    | 5        | 4.5      | Aggrégation                            | 12        |
| 2.4      | Enrichissement de la pédologie   | 6        | <b>5</b> | <b>Mise en cartographie dynamique</b>  | <b>13</b> |
| 2.5      | Enrichissement des AOC de 1936   | 7        | 5.1      | R code                                 | 13        |
| 2.6      | Enrichissement des lieux dits    | 8        | 5.2      | Deploiement                            | 15        |
| 2.7      | Enregistrement de la base        | 8        | <b>6</b> | <b>Mise en application Shiny</b>       | <b>15</b> |
| <b>3</b> | <b>Statistiques descriptives</b> | <b>9</b> | 6.1      | TEST 1                                 | 15        |
| 3.1      | Général                          | 9        | 6.2      | TEST 2                                 | 17        |
| 3.2      | Bilan surfacique des AOC         | 9        | 6.3      | TEST 3                                 | 19        |
| 3.3      | Liens avec les AOC historiques   | 9        | 6.4      | TEST 4                                 | 21        |
| 3.4      | Distribution spatiale            | 9        | 6.5      | TEST 5                                 | 23        |

|     |            |    |     |                                      |    |
|-----|------------|----|-----|--------------------------------------|----|
| 6.6 | TEST 6     | 25 | 8   | Bibliographie                        | 30 |
| 6.7 | Global     | 28 |     |                                      |    |
| 6.8 | Server     | 29 | A   | Annexes                              | 31 |
| 6.9 | ui         | 29 | A.1 | Annexe 1: incohérence des AOC        | 31 |
| 7   | Conclusion | 30 | A.2 | Annexe 2: les intitulés pédologiques | 31 |

## 1 Introduction

Les Appellations d'Origines Contrôlées (AOC) en Bourgogne sont issues de processus humains qui ont travaillées, répertoriés puis classés les parcelles en fonction de leur capacité à produire des vins de qualité. Niveaux.

D'un point de vue économique, les AOC sont un signe de qualité bla mais peuvent aussi faire l'objet d'une demande pour eux mêmes.

Une analyse économique des "signes de qualité". Labels et certification des produits Laurent Linnemer sem-link Anne Perrot Revue économique Année 2000 51-6 pp. 1397-1418

Asymétrie de l'information, réputation et certification Bénédicte Coestier Annales d'Économie et de Statistique No. 51 (Jul. - Sep., 1998

Christophe Lucand (dans Wolikow and Jacquet (2011)) cite les experts fondateurs (les mêmes qu'Olivier): Jullien, Morelot et Laval, supposent l'existence d'une hiérarchie commune en trois ou quatre catégories, avec au sommet les "têtes de cuvée" puis les premières cuvées. Puis il cite la thèse d'Olivier. A cette hiérarchie transversale se superpose une hiérarchie par villages qui ne détermine cependant en rien la réalité des zones d'approvisionnement concernées. Il s'agit plutôt d'identifications commerciales communes, investies d'un plus ou moins grand capital symbolique hérité. Ce capital symbolique hérité attribut un prestige plus ou moins grand à certaines communes ou propriétaires particulier.

le décret instaurant les Premiers Crus ne fut toutefois adopté qu'en 1943. Deux classements historiques servirent de principales références à la désignation de ces ceux-ci: celui de Jules Laval de 1855 et le Classement du Comité d'Agriculture et de Viticulture de l'Arrondissement de Beaune de 1860.

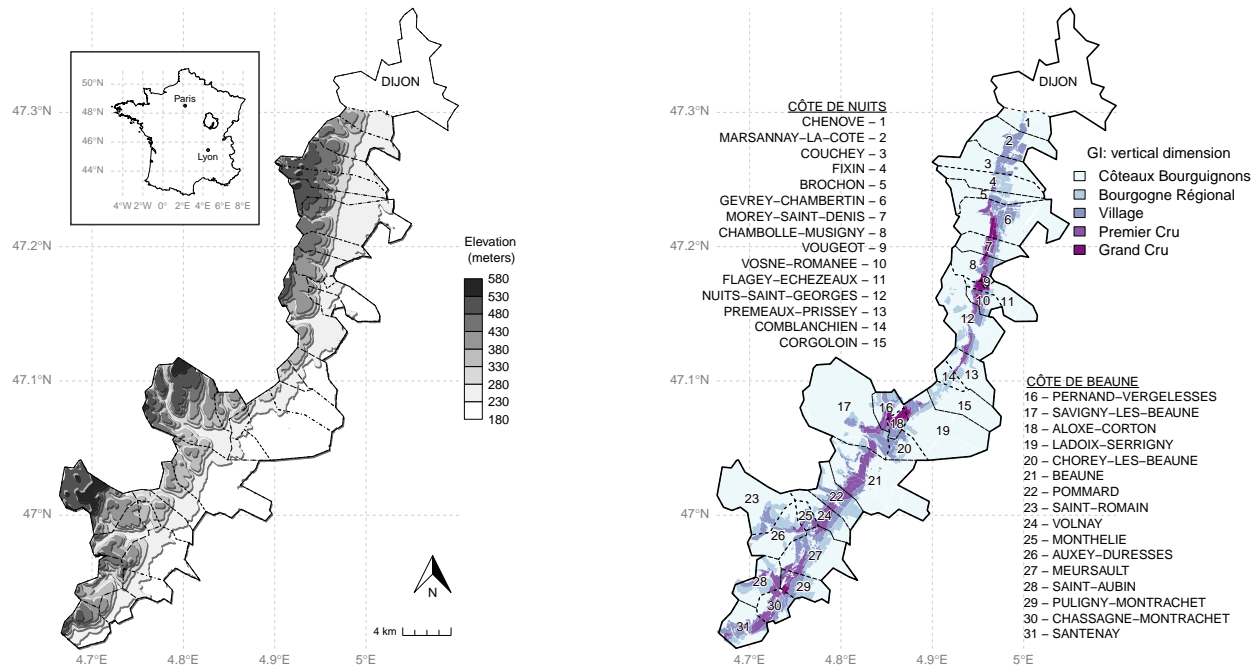
L'objet de cet article est premièrement de présenter la construction de la base de données. carte de la zone pas les hautes cotes ni le châtillonnais. correspond à l'unesco? <https://whc.unesco.org/fr/list/1425/>

Une aire parcellaire délimitée désigne une délimitation qui repose sur les limites administratives du cadastre et dont le maillage suffisamment fin permet de tenir compte de variations très localisées des éléments du milieu physique (règlements européens 510/2006 et 1234/2007) AOC en Cote d'Or: PCI-Vecteur ou cadastre de l'IGN. Peut comprendre des parcelles découpées.

Nous présentons également une application de cette base de données pour proposer une analyse statistique de l'information présente dans les AOC. En lien avec leur structuration hiérarchique en niveau,

Ce document contient le code R, packages, github, etc. Les bases de données sources qui entrent dans le travail sont disponibles auprès des auteurs sur demande.

Figure 1: Vignobles de la Côte d'Or, topographie et appellations d'origine contrôlées  
blabla.



## 2 Construction des données

### 2.1 Du cadastre aux AOC

Le travail porte sur l'ensemble des parcelles cadastrales des 31 communes de la Côte de Beaune et de la Côte de Nuits reportées dans la Fig:1. La géométrie du parcellaires est issue de la BD parcellaire de l'IGN version X.XX téléchargée le XX/XX/2018. Deux traitements ont été effectués au préalable, nous avons calculé à l'aide d'un système d'information géographique (SIG) des caractéristiques géométriques des parcelles (surface, périmètre, et distance maximale entre deux sommets, voir Table 1) et appariée l'information sur les AOC à partir du fichier de l'INAO sur <http://data.gouv.fr/inao>. Sur ce deuxième point, nous avons retenu pour chaque parcelle l'AOC la plus haute dans la hiérarchie. Nous avons vérifiée la distribution des surface par AOC par commune, colle assez bien avec le bouquin.

```
library(sp) ; library(rgdal)
Geo.Cad <- readOGR("./Carto", "GeoCad")
sapply(Geo.Cad@data, function(x) sum(is.na(x)))
```

OGR data source with driver: ESRI Shapefile  
Source: "/home/jsay/geoInd/Carto", layer: "GeoCad"  
with 110350 features  
It has 16 fields

| IDU | CODECOM | AREA | PERIM | MAXDIST | PAR2RAS | PAOC | BGOR |
|-----|---------|------|-------|---------|---------|------|------|
|-----|---------|------|-------|---------|---------|------|------|

|       |       |       |       |       |     |       |       |
|-------|-------|-------|-------|-------|-----|-------|-------|
| 0     | 0     | 0     | 0     | 0     | 0   | 49718 | 49718 |
| BOUR  | VILL  | COMM  | PCRU  | GCRU  | AOC | AOCtp | AOC1b |
| 49718 | 49718 | 49718 | 49718 | 49718 | 0   | 49718 | 49718 |

La base parcellaire contient donc 110 350 observations et 16 variables en incluant PAR2RAS un identifiant pour l'appariement avec les données raster. L'information sur les AOC tient en 3 variables, AOC, AOCtp et AOC1b. Les valeurs omises correspondent à l'absence d'AOC. Attention ce ne sont pas des AOC, en particulier pour les Bourgognes régionaux et les premiers crus (`table(Geo.Cad$AOC, Geo.Cad$AOCtp)`). Pour les régionaux, il s'agit d'avoir les Bourgognes Côte d'Or et les hautes côtes et Notre Dame La Chapelle (on n'a pas Le Chapitre).

Table 1: Numéro, nom, type et label des variables parcellaires

| NOM     | TYPE                                   | DESCRIPTION                              |
|---------|--|--|
| IDU     | <i>Caractère issu de concaténation</i> | Identifiant cadastral                    |
| CODECOM | <i>Caractère de taille 5</i>           | Code INSEE commune                       |
| AREA    | <i>Numérique en mètres carrés</i>      | Surface de la parcelle                   |
| PERIM   | <i>Numérique en mètres</i>             | Périmètre de la parcelle                 |
| MAXDIST | 1                                      | Distance maximale entre deux sommets     |
| PAR2RAS | 2                                      | Identifiant pour appariement avec raster |
| PAOC    | 3                                      | Hors périmètre AOC                       |
| BGOR    | 4                                      | Coteaux Bourguignon                      |
| BOUR    | 5                                      | Bourgogne Régional                       |
| VILL    | 6                                      | Bourgogne Village                        |
| COMM    | 7                                      | Bourgogne Communal                       |
| PCRU    | 8                                      | Premier Cru                              |
| GCRU    | 9                                      | Grand Cru                                |
| AOC     | 10                                     | Rang dans la hiérarchie                  |
| AOCtp   | 11                                     | Type de nom                              |
| AOC1b   | 12                                     | Libellé                                  |

## 2.2 Enrichissement de la topographie

Les données raster sont également issues du cloud, avec le fichier `vitidem.csv` qui contient 15 millions de cellules pour 13 variables. Nous transformons la variable catégorielle MOS sur les modes d'occupation du sol en indicatrices afin de pouvoir l'agréger au niveau des parcelles. Ensuite, les autres variables quantitatives seront simplement moyennées au niveau des parcelles. Nous pourrions imaginer d'autres méthodes d'agrégation et reporter d'autres statistiques: pour plus tard.

```
library(data.table)
dim(Dat.Ras <- fread("./Data/DatRas.csv"))
Cad.Ras <- Dat.Ras[, lapply(.SD, mean), by= list(PAR2RAS),
                     .SDcols= names(Dat.Ras)[ -c(1, 4)]]
Geo.Ras <- merge(Geo.Cad, Cad.Ras, by= "PAR2RAS")
sapply(Geo.Ras@data[, 17: 26], function(x) sum(is.na(x))); rm(Dat.Ras)
```

data.table 1.11.4 Latest news: <http://r-datatable.com>

[1] 14253070 13

|      |      |       |       |        |       |      |       |        |       |
|------|------|-------|-------|--------|-------|------|-------|--------|-------|
| XL93 | YL93 | NOMOS | URBAN | FOREST | WATER | DEM  | SLOPE | ASPECT | SOLAR |
| 2096 | 2096 | 2096  | 2096  | 2096   | 2096  | 2096 | 2096  | 2096   | 2096  |

Il y a 2 096 parcelles pour lesquelles le code PAR2RAS ne s'apparie à aucune des cellules raster. Après une identification de ces petites parcelles (taille médiane de 10 m<sup>2</sup>), il s'agit de spécificités du cadastre que nous choisissons de négliger. Cela revient à enlever 2.7 ha, moins de 0.01 % de la surface totale.

| NOM    | TYPE             | DESCRIPTION                 |
|--------|------------------|-----------------------------|
| XL93   | <i>Numérique</i> | Latitude Lambert 93         |
| YL93   | <i>Numérique</i> | Longitude Lambert 93        |
| NOMOS  | 1                | Part hors MOS               |
| URBAN  | 2                | Part en usage urbain        |
| FOREST | 3                | Part en Forêt               |
| WATER  | 4                | Part en eau                 |
| DEM    | 5                | Altitude moyenne en mètres  |
| SLOPE  | 6                | Pente moyenne en degrés     |
| ASPECT | 7                | Exposition (degrés)         |
| SOLAR  | 8                | Radiation solaire en Joules |
| PERMEA | 9                | Perméabilité                |

## 2.3 Enrichissement de la géologie

Depuis mars 2019, le BRGM a libéré l'accès aux cartes géologiques au 1/50 000 Bd Charm-50 sous licence Ouverte / Open Licence Etalab Version 2.0 (<http://infoterre.brgm.fr/page/conditions-dutilisation-donnees>) Les données utilisées ici sont une extraction de la Côte d'Or, téléchargées en avril 2019 sur le site <http://infoterre.brgm.fr>. Les données sont constituées de différentes couches SIG décrivant les formations géologiques, les éléments linéaires et ponctuels structuraux et divers. Nous utilisons simplement la couche principale XX de laquelle nous extrayons les variables XX. On a fait une sélection sur les valeurs omises et sur la redondance d'information.

```
Geol.Map <- readOGR("./Carto/", "GeolMap")
Pts.Cad <- SpatialPoints(Geo.Ras, proj4string= CRS(proj4string(Geol.Map)))
Geo.Ras@data <- cbind(Geo.Ras@data, over(Pts.Cad, Geol.Map))
sapply(Geo.Ras@data[, 28: 43], function(x) sum(is.na(x)))
```

OGR data source with driver: ESRI Shapefile  
Source: "/home/jsay/geoInd/Carto", layer: "GeolMap"  
with 13960 features  
It has 16 fields

|      |          |       |          |          |        |
|------|----------|-------|----------|----------|--------|
| CODE | NOTATION | DESCR | TYPEGEOL | APLOCALE | TYPEAP |
| 31   | 31       | 31    | 31       | 862      | 862    |

|         |           |           |          |        |            |
|---------|-----------|-----------|----------|--------|------------|
| GEOLNAT | ISOPIQUE  | AGEDEB    | ERADEB   | SYSDEB | LITHOLOGIE |
| 31      | 31        | 31        | 31       | 31     | 31         |
| DURETE  | ENVIRONMT | GEOCHIMIE | LITHOCOM |        |            |
| 69      | 31        | 31        | 69       |        |            |

Le code précédent permet d'apparier les 14000 polygones avec les 13 variables sur la géologie (à l'échelle de la Côte d'Or). J'ai fait une demande et une relance au BRGM pour obtenir le dictionnaire des variables reportées dans le tableau ci-dessous, qui est incomplet car je n'ai pas encore eu de réponse. Les valeurs manquantes sont distinctes des valeurs manquantes de la topographie.

| NOM        | VALEURS                        | DESCRIPTION               |
|------------|--------------------------------|---------------------------|
| CODE       | <i>Caractère, 31 modalités</i> | Code de la géologie       |
| NOTATION   | <i>Caractère, 31 modalités</i> | Notation géologie         |
| DESCR      | <i>Caractère, 31 modalités</i> | Description géologie      |
| TYPEGEOL   | <i>Caractère, 4 modalités</i>  | Type superficiel          |
| APLOCALE   | <i>Caractère, 28 modalités</i> | Colluvions, Eboulis, etc. |
| TYPEAP     | <i>Caractère, 7 modalités</i>  | Type de formation         |
| GEOLNAT    | <i>Caractère, 3 modalités</i>  | Nature Géologique         |
| ISOPIQUE   | <i>Caractère, 4 modalités</i>  | Faciès des couches        |
| AGEDEB     | <i>Caractère, 24 modalités</i> | Age de la couche          |
| ERADEB     | <i>Caractère, 2 modalités</i>  | Céno ou Més               |
| SYSDEB     | <i>Caractère, 5 modalités</i>  | Age autre                 |
| LITHOLOGIE | <i>Caractère, 16 modalités</i> | Litho                     |
| DURETE     | <i>Caractère, 3 modalités</i>  | Dureté                    |
| ENVIRONMT  | <i>Caractère, 9 modalités</i>  | Environnement             |
| GEOCHIMIE  | <i>Caractère, 5 modalités</i>  | Géochimie                 |
| LITHOCOM   | <i>Caractère, 30 modalités</i> | Litho détaillée           |

## 2.4 Enrichissement de la pédologie

La couche pédologique est issue du Référentiel Pédologique de Bourgogne : Régions naturelles, pédopaysage et sols de Côte d'Or (étude 25021) au 1/250 000, compatible avec la base de données nationale DoneSol. La localisation des types de sol s'opère par les Unités Cartographiques de Sols, appelées aussi Pédopaysages qui regroupent de zones homogènes qui contiennent néanmoins différents types de sols mais sans que ces derniers puissent être localisés plus précisément. En l'absence de données plus fines spatialement, les données parcellaires seront enrichies des code des unités cartographiques que l'on enrichie avec des variables de description des sols. Les intitulés des UCS sont obtenus par un travail manuel reporté à l'annexe 2 (par le site <https://bourgogne.websol.fr/carto>). On peut citer ma thèse.

```
Pedo.Map <- readOGR("./Carto", "PedoMap")
Geo.Ras@data <- cbind(Geo.Ras@data, over(Pts.Cad, Pedo.Map))
Geo.Ras@data[, c(45: 48, 50: 55)] <-
  apply(Geo.Ras@data[, c(45: 48, 50: 55)], 2, as.numeric)
sapply(Geo.Ras@data[, 44: 56], function(x) sum(is.na(x)))
```

OGR data source with driver: ESRI Shapefile

Source: "/home/jsay/geoInd/Carto", layer: "PedoMap"

with 194 features

It has 13 fields

|       |        |        |       |       |        |       |       |       |       |
|-------|--------|--------|-------|-------|--------|-------|-------|-------|-------|
| NOUC  | SURFUC | TARG   | TSAB  | TLIM  | TEXTAG | EPAIS | TEG   | TMO   | RUE   |
| 14645 | 14645  | 14645  | 14645 | 14645 | 14645  | 14645 | 14645 | 14645 | 14645 |
| RUD   | OCCUP  | DESCRp |       |       |        |       |       |       |       |
| 14645 | 14645  | 14645  |       |       |        |       |       |       |       |

Il apparaît que les descriptions des Pédopaysages combinent des caractéristiques topographiques (Plaines, massifs, piedmonts), des caractéristiques d'occupation (forestiers, vignoble) et des caractéristiques géologiques (plio-pléistocènes, calcaires). Le redondance de ce découpage avec les variables topographiques, le découpage géologique et le mode d'occupation des sols se pose effectivement. Les valeurs manquantes correspondent *a priori* aux espaces urbanisés (mais cela ne se retrouve pas vraiment à partir du MOS).

| NOM    | TYPE                               | DESCRIPTION                               |
|--------|------------------------------------|---|
| NOUC   | <i>Caractère de taille 2</i>       | Numéro de l'unité cartographique          |
| SURFUC | <i>Numérique en mètres carrés</i>  | Surface de l'unité cartographique         |
| TARG   | <i>Numérique en pourcentage</i>    | Taux d'argile dans l'unité cartographique |
| TSAB   | <i>Numérique en pourcentage</i>    | Taux de sable dans l'unité cartographique |
| TLIM   | <i>Numérique en pourcentage</i>    | Taux de limon dans l'unité cartographique |
| TEXTAG | <i>Caractère avec 9 modalités</i>  | Classes de textures agrégées              |
| EPAIS  | <i>Numérique en centimètres</i>    | Épaisseur des sols                        |
| TEG    | <i>Numérique en pour mille</i>     | Taux d'éléments grossiers                 |
| TMO    | <i>Numérique en pourcentage</i>    | Taux de Matière organique                 |
| RUE    | <i>Numérique en millimètres</i>    | Réserve Utile par excès                   |
| RUD    | <i>Numérique en millimètres</i>    | Réserve Utile par défaut                  |
| OCCUP  | <i>Numérique en pourcentage</i>    | Part de l'unité typologique               |
| DESCRp | <i>Caractère avec 33 modalités</i> | Description de la classe pédologique      |

## 2.5 Enrichissement des AOC de 1936

Un travail préalable a été effectué sur les AOC de 1936 afin de compiler les différentes années de 1936 à 1940. La localisation est effectuée par le centroïde des parcelles cadastrales car la géométrie des polygones ne correspond pas parfaitement (à la fois par la numérisation et parce que le cadastre a changé).

```
Hist.Aoc <- readOGR("Carto/", "Aoc1936")
Geo.Ras@data <- cbind(Geo.Ras@data, over(Pts.Cad, Hist.Aoc))
sapply(Geo.Ras@data[, 57: 58], function(x) sum(is.na(x)))
```

OGR data source with driver: ESRI Shapefile

Source: "/home/jsay/geoInd/Carto", layer: "Aoc1936"

with 56 features

It has 2 fields

AOC36lab AOC36lvl

70 70

Nous obtenons des aires sensiblement plus réduites que les actuelles, 27% au lieu de 55% trouvés ci-dessus. Hormis le creux de 1938, entre 10 et 15% des parcelles sont classées chaque années, sachant qu'il y a du double compte. Les premiers crus n'apparaissent pas car ils n'existaient pas à l'époque (création en 1948).

## 2.6 Enrichissement des lieux dits

Il s'agit ici d'inclure de l'information cadastrale à partir des sources `data.gouv.fr`. Nous utilisons le Plan Cadastral Informatisé Vecteur (Format EDIGÉO, <https://cadastre.data.gouv.fr/datasets/plan-cadastral-informatise>) téléchargé pour la Côte d'Or (21) le <2019-01-13 dim.>. License ouverte Etalab. La difficulté avec les lieux dit est qu'ils doivent être croisés avec les communes car un même nom lieu dit peut être présent sur plusieurs communes. Comme la géométrie des lieux dits et des parcelles colle parfaitement, nous pouvons enrichir les données parcellaires directement par le centroïde. Ajout <2019-01-23 mer.>, des données communales, nous extrayons également les coordonnées des chefs-lieux pour calculer une distance à vol d'oiseaux, la population (peuvent être des sur-identifications sur le land use) et la distinction Côte de Beaune / Côtes de Nuits. Nous enregistrons également une shapefile MapCom qui permet de cartographier les contours communaux dans les figures.

---

```
Lieu.Dit <- readOGR("./Carto/", "LieuDit")
Geo.Ras@data <- cbind(Geo.Ras@data, over(Pts.Cad, Lieu.Dit[, -1]))
sapply(Geo.Ras@data[, 59: 68], function(x) sum(is.na(x)))
```

---

```
OGR data source with driver: ESRI Shapefile
Source: "/home/jsay/geoInd/Carto", layer: "LieuDit"
with 3285 features
It has 11 fields
```

|         |          |        |      |      |        |        |
|---------|----------|--------|------|------|--------|--------|
| LIEUDIT | CLDVIN   | LIBCOM | XCHF | YCHF | ALTCOM | SUPCOM |
| 4494    | 4494     | 4494   | 4494 | 4494 | 4494   | 4494   |
| POPCOM  | CODECANT | REGION |      |      |        |        |
| 4494    | 4494     | 4494   |      |      |        |        |

Pour 4% des parcelles, aucun lieu dit n'a été apparié. Ces parcelles se concentrent sur les communes de Chenôve, Marsannay-la-Côte et Beaune (Corgoloin dans une moindre mesure). Ces "trous" apparaissent déjà dans le fichier source et ne sont donc pas un résultat de l'appariement. Ils semblent être des espaces bâtis sur la carte, mais ce n'est toujours pas confirmé par le MOS.

## 2.7 Enregistrement de la base

Pour l'instant, on est à moins de 500 Mo.

---

```
dim(Geo.Ras)
save(Geo.Ras, file= "Inter/GeoRas.Rda")
writeOGR(Geo.Ras, "Carto/", "GeoRas", driver= "ESRI Shapefile")
```

---

[1] 110350      68



## 3 Statistiques descriptives

### 3.1 Général

Avant ici pas de stat des

---

```
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)
```

---

yop

### 3.2 Bilan surfacique des AOC

Définition de nos niveaux et implications en termes de surfaces sur la pyramides des AOC.

The endogeneity is about the size or the shape of parcels, but not the pedoclimatic variables. The endogeneity of the size/ shape of parcel can be due both to simultaneity and omitted land quality effects. Both seems to be intuitively treated. Size of parcels multiples of ha, m2 or ouvrée (= 428 m2)?

### 3.3 Liens avec les AOC historiques

First load the .shp file in the R workspace.

The database contains ...

Dans le Data paper, il s'agira d'identifier les grands crus des villages avec et sans nom reconnus pour retrouver la structure hiérarchique.

Hiérarchisation des données historiques par les nom de crus et s'il sont présents dans les nouvelles données.

$$y = ax + b \tag{1}$$

Retravail des données brutes AOC (XX et XXI) et création des niveaux hiérarchiques.

### 3.4 Distribution spatiale

## 4 Modèle ordonné de désignation

### 4.1 Variable transformations

---

```
Geo.Ras$RAYAT <- (Geo.Ras$SOLAR- mean(Geo.Ras$SOLAR, na.rm= TRUE))/
sd(Geo.Ras$SOLAR, na.rm= TRUE)
Geo.Ras$EXPO <- factor(ifelse(Geo.Ras$ASPECT< 45, "0-45",
```

```

      ifelse(Geo.Ras$ASPECT< 90, "45-90",
      ifelse(Geo.Ras$ASPECT<135, "90-135",
      ifelse(Geo.Ras$ASPECT<180, "135-180",
      ifelse(Geo.Ras$ASPECT<225, "180-225",
      ifelse(Geo.Ras$ASPECT<270, "225-270",
      ifelse(Geo.Ras$ASPECT<315, "270-315", "315-360"))))))) ,
      levels= c("0-45", "45-90", "90-135", "135-180",
      "180-225", "225-270", "270-315", "315-360"))

GR84 <- spTransform(Geo.Ras, CRS("+proj=longlat +ellps=WGS84"))
dd <- coordinates(GR84)
Geo.Ras$X= dd[, 1]
Geo.Ras$Y= dd[, 2]

```

---

## 4.2 Spécification du modèle

La différence avec le multinomial c'est dans l'interprétation des données. Dans le MNL, tu dis c'est VILL est la meilleure AOC pour cette parcelle. Dans le OP, tu dis cette parcelle peut est mieux que Bourgogne, mieux que VILL, mais moins bien que PCRU et Grand cru. L'OP intègre mieux l'information, il ne faut pas mettre les 2 en concurrence. cette pratique est liée au principe de hiérarchisation des appellations d'origine, qui [...] s'emboîtent de manière pyramidale à partir d'une appellation régionale socle [...]. Dans cette optique, le vin élaboré selon le cahier des charges d'une appellation hiérarchiquement supérieure répondrait de facto aux exigences de l'appellation régionale, dont les conditions de production sont moins contraignantes.

---

```

library(mgcv) ## ASSEZ LONG
gam2 <- gam(AOCc~ s(DEM, k= 10)+ s(SLOPE)+ s(ASPECT)+ s(RAYAT)+ s(PERMEABILITY)
      + s(XREG, YREG, k= 200)+ LIBCOM
      , data= RegRank, family= ocat(R= 5))

summary(gam2)
plot(gam2, scale= 0)

plot(density((gam2$linear.pred- min(gam2$linear.pred))/
      (max(gam2$linear.pred)- min(gam2$linear.pred))))
prdat <- RegRank
prdat$LIBCOM <- "BROCHON"

gg <- predict(gam2, type= "response", newdata= prdat)
hh <- ifelse(gg[, 1]> 1- 1/1e16, 1- 1/1e16, gg[, 1])
prdat$score <- qlogis(1- hh)
RegRank$SCORE <- (prdat$score- min(prdat$score))/
      (max(prdat$score)- min(prdat$score))

plot(density(RegRank$SCORE))
library(plyr)
ee <- ddply(RegRank, .(CODEld),
      function(x) data.frame(Mean= mean(x$SCORE),
      Median= median(x$SCORE),
      WMean= weighted.mean(x$SCORE, x$Area)))
head(ee[order(ee$Mean, decreasing= TRUE), ], 20)

ff <- ddply(RegRank, .(LIBCOM),
      function(x) data.frame(Mean= mean(x$SCORE),
      Median= median(x$SCORE),

```

```

WMean= weighted.mean(x$SCORE, x$Area)))
ff[order(ff$Mean, decreasing= TRUE), ]
ff[order(ff$WMean, decreasing= TRUE), ]

```

---

### 4.3 Effets des variables biophysiques

### 4.4 Prédiction du score et classifications

---

```

load("Inter/gamod.Rda")
GAM900 <- predict(gamod$gam900, newdata= Geo.Ras@data, type= "terms")

Geo.Ras$G900raw <- rowSums(GAM900, na.rm= TRUE)
Geo.Ras$G900cor <- mean(GAM900[, 1], na.rm= T)+
  rowSums(GAM900[, -1], na.rm= TRUE)

Geo.Plt <- subset(Geo.Ras@data, PAOC== 1)
unini <- function(x) (x- min(x))/ (max(x)- min(x))
Geo.Plt$G900raw <- round(unini(Geo.Plt$G900raw)* 100, 2)
Geo.Plt$G900cor <- round(unini(Geo.Plt$G900cor)* 100, 2)

library(plotly)
## https://plot.ly/r/violin/

library(ggplot2)

## overlay
p <- ggplot(Geo.Plt, aes(factor(AOC), G900raw))
p <- p + geom_violin(aes(colour = "#1268FF"), alpha = 0.3)
q <- p + geom_violin(aes(y = G900cor, colour = "#3268FF"), alpha = 0.3)
q+ ylim(40, 100)

## split
library(plyr)
revAOC <- c("1"= "Coteaux b.", "2"= "Bourgogne", "3"= "Village",
           "4"= "Premier cru", "5"= "Grand Cru")
cc <- rbind(
  data.frame(AOC= revalue(factor(Geo.Plt$AOC), revAOC),
             Score= Geo.Plt$G900raw,
             Pr= "Uncorrected: without communes fixed effects"),
  data.frame(AOC= revalue(factor(Geo.Plt$AOC), revAOC),
             Score= Geo.Plt$G900cor,
             Pr= "Corrected: with communes fixed effects"))

ggplot(cc, aes(factor(AOC), Score, fill= Pr))+
  geom_split_violin()+
  ylab("100-Point Vineyard Quality Score")+
  ylim(40, 100)+ theme_minimal()+ xlab("")+
  geom_split_violin(draw_quantiles = c(0.25, 0.5, 0.75))+
  theme(legend.justification=c(0, 1), legend.position=c(0, 1),
       legend.title = element_blank())
#ggsave("Figures/MainPlot.png")

## PAS DE C-c C-c

```

```

GeomSplitViolin <- ggproto("GeomSplitViolin", GeomViolin,
  draw_group = function(self, data, ..., draw_quantiles = NULL) {
    # Original function by Jan Gleixner (@jan-glx)
    # Adjustments by Wouter van der Bijl (@Axeman)
    data <- transform(data, xminv = x - violinwidth * (x - xmin), xmaxv = x + violinwidth * (xmax - x))
    grp <- data[1, "group"]
    newdata <- plyr::arrange(transform(data, x = if (grp %% 2 == 1) xminv else xmaxv), if (grp %% 2 == 1) y else -y)
    newdata <- rbind(newdata[1, ], newdata, newdata[nrow(newdata), ], newdata[1, ])
    newdata[c(1, nrow(newdata) - 1, nrow(newdata)), "x"] <- round(newdata[1, "x"])
    if (length(draw_quantiles) > 0 & !scales::zero_range(range(data$y))) {
      stopifnot(all(draw_quantiles >= 0), all(draw_quantiles <= 1))
      quantiles <- create_quantile_segment_frame(data, draw_quantiles, split = TRUE, grp = grp)
      aesthetics <- data[rep(1, nrow(quantiles)), setdiff(names(data), c("x", "y")), drop = FALSE]
      aesthetics$alpha <- rep(1, nrow(quantiles))
      both <- cbind(quantiles, aesthetics)
      quantile_grob <- GeomPath$draw_panel(both, ...)
      ggplot2:::ggname("geom_split_violin", grid::grobTree(GeomPolygon$draw_panel(newdata, ...), quantile_grob))
    }
    else {
      ggplot2:::ggname("geom_split_violin", GeomPolygon$draw_panel(newdata, ...))
    }
  }
)

create_quantile_segment_frame <- function(data, draw_quantiles, split = FALSE, grp = NULL) {
  dens <- cumsum(data$density) / sum(data$density)
  ecdf <- stats::approxfun(dens, data$y)
  ys <- ecdf(draw_quantiles)
  violin.xminvs <- (stats::approxfun(data$y, data$xminv))(ys)
  violin.xmaxvs <- (stats::approxfun(data$y, data$xmaxv))(ys)
  violin.xs <- (stats::approxfun(data$y, data$x))(ys)
  if (grp %% 2 == 0) {
    data.frame(
      x = ggplot2:::interleave(violin.xs, violin.xmaxvs),
      y = rep(ys, each = 2), group = rep(ys, each = 2)
    )
  } else {
    data.frame(
      x = ggplot2:::interleave(violin.xminvs, violin.xs),
      y = rep(ys, each = 2), group = rep(ys, each = 2)
    )
  }
}

geom_split_violin <- function(mapping = NULL, data = NULL, stat = "ydensity", position = "identity", ...,
  draw_quantiles = NULL, trim = TRUE, scale = "area", na.rm = FALSE,
  show.legend = NA, inherit.aes = TRUE) {
  layer(data = data, mapping = mapping, stat = stat, geom = GeomSplitViolin, position = position,
    show.legend = show.legend, inherit.aes = inherit.aes,
    params = list(trim = trim, scale = scale, draw_quantiles = draw_quantiles, na.rm = na.rm, ...))
}

```

---

## 4.5 Aggrégation

---

```

gg <- aggregate(Geo.Ras$G900raw, by= list(Geo.Ras$A0C1b), mean, na.rm= T)

```

```
hh <- aggregate(Geo.Ras$G900cor, by= list(Geo.Ras$AOC1b), mean, na.rm= T)

head(gg[order(gg$x, decreasing= TRUE), ], n= 100)
head(hh[order(hh$x, decreasing= TRUE), ], n= 100)
hh[hh$Group.1== 'Romanée-Conti',]
```

---

## 5 Mise en cartographie dynamique

### 5.1 R code

#### AGGREGATION PAR LIEUX DITS

On utilise mapview, <https://r-spatial.github.io/mapview/>

- sudo apt install libgdal-dev
- sudo ln -s /usr/lib/rstudio/bin/pandoc/pandoc /usr/local/bin
- webshot::installphantomjs()

On pourrait également utiliser:

- <http://symbolixau.github.io/googleway/articles/googleway-vignette.html>
- <https://www.osgeo.org/projects/mapguide-open-source/>
- <http://geoserver.org/>
- <https://rstudio.github.io/leaflet/shiny.html>
- <https://github.com/mtennekes/tmap>

On peut mettre des graphiques quand on clique sur un polygone: [https://r-spatial.github.io/mapview/articles/articles/mapview\\_04-popups.html](https://r-spatial.github.io/mapview/articles/articles/mapview_04-popups.html)

also show info on the epsg code and the proj4string press and hold Ctrl and move the mouse. addMouseCoordinates also allows us to copy the info about the current mouse position to the clipboard by holding the Ctrl and left-clicking on the map.

---

```
library(rgdal) ; library(rgeos) ; library(maptools)
Geo.Aoc <- subset(Geo.Ras, PAOC== 1)
Geo.Aoc$U900raw <- Geo.Plt$G900raw
Geo.Aoc$U900cor <- Geo.Plt$G900cor

NVA <- c("Coteaux b.", "Bourgogne", "Village", "Premier cru", "Grand cru")
names(NVA) <- 1: 5
library(plyr)
Geo.Aoc$NIVEAU <- revalue(factor(Geo.Aoc$AOC), NVA)
Geo.Aoc$NAME <- ifelse(Geo.Aoc$AOC== 5, as.character(Geo.Aoc$AOC1b),
  ifelse(Geo.Aoc$AOC< 4, as.character(Geo.Aoc$LIEUDIT), NA))
for (i in 1: nrow(Geo.Aoc)){
```

```

    if (is.na(Geo.Aoc$NAME[ i])){
      Geo.Aoc$NAME[ i] <- substr(Geo.Aoc$AOC1b[ i],
                                regexpr(" cru+", Geo.Aoc$AOC1b[ i],
                                           perl= T)+ 5,
                                nchar(as.character(Geo.Aoc$AOC1b[ i])))
    } else {(Geo.Aoc$NAME[ i])}
  }
  Geo.Aoc$Concat <- paste0(Geo.Aoc$AOC, Geo.Aoc$LIBCOM, Geo.Aoc$NAME)
  tmp_geo <- gBuffer(Geo.Aoc, byid= TRUE, width= 0)
  Poly.tst <- unionSpatialPolygons(tmp_geo, Geo.Aoc$Concat)

  Poly.tst$xx <- as.character(row.names(Poly.tst))
  attrTb <- data.frame(xx= Geo.Aoc$Concat[ !duplicated(Geo.Aoc$Concat)],
                      NIVEAU= Geo.Aoc$NIVEAU[ !duplicated(Geo.Aoc$Concat)],
                      LIBCOM= Geo.Aoc$LIBCOM[ !duplicated(Geo.Aoc$Concat)],
                      NAME= Geo.Aoc$NAME[ !duplicated(Geo.Aoc$Concat)])
  Poly.tst <- merge(Poly.tst, attrTb, by= "xx")
  Poly.tst$PrdRaw <- aggregate(Geo.Aoc$U900raw,
                              by= list(Geo.Aoc$Concat, mean)[, 2])
  Poly.tst$PrdCor <- aggregate(Geo.Aoc$U900cor,
                              by= list(Geo.Aoc$Concat, mean)[, 2])
  Poly.tst$Area <- aggregate(Geo.Aoc$AREA,
                              by= list(Geo.Aoc$Concat, sum)[, 2])

  Poly.tst@data[, 5: 7] <- round(Poly.tst@data[, 5: 7], 2)

  library(RColorBrewer)
  AocPal <- brewer.pal(5, "BuPu")
  library(mapview)

  mapviewOptions(basemaps = mapviewGetOption("basemaps")[ c(4, 1: 3, 5)])
  tst <- mapview(Poly.tst,
                zcol= "NIVEAU", alpha.regions= .5,
                col.regions = AocPal,
                color= "white", legend.opacity= .5,
                label= Poly.tst$NAME, layerId= Poly.tst$xx,
                popup = popupTable(Poly.tst@data, feature.id= FALSE,
                                   zcol = c("NIVEAU", "LIBCOM", "NAME",
                                             "Area", "PrdRaw", "PrdCor")))
  ## addLogo(tst, "http://www7.inra.fr/fournisseurs/images/logo.jpg",
  ##        width = 200, height = 100)
  ## mapviewOptions(basemaps = mapviewGetOption("basemaps")[ c(4, 1: 3, 5)],
  ##               raster.palette = colorRampPalette(brewer.pal(9, "Greys")),
  ##               vector.palette = colorRampPalette(brewer.pal(9, "YlGnBu")),
  ##               na.color = "magenta",
  ##               layers.control.pos = "topright")

  ## create standalone .html
  mapshot(tst, url = paste0(getwd(), "/DynMap/LieuxDits.html"))

  ## create .html and .png
  mapshot(m, url = paste0(getwd(), "/DynMap/test.html"),
         file = paste0(getwd(), "/DynMap/test.png"),
         remove_controls = c("homeButton", "layersControl"))

```

## 5.2 Deploiement

<https://firebase.google.com/>

```
wget -qO- https://raw.githubusercontent.com/creationix/nvm/v0.34.0/install.sh | bash
https://websiteforstudents.com/install-the-latest-node-js-and-nmp-packages-on-ubuntu-16-04-18
nvm install 10.15.3
```

---

```
from https://github.com/creationix/nvm:
You can list available versions using ls-remote:
nvm ls-remote
And then in any new shell just use the installed version:
nvm use node
Or you can just run it:
nvm run node --version
Or, you can run any arbitrary command in a subshell with the desired version of node:
nvm exec 10.15.3 node --version
nvm install 10.15.3
```

---

npm install -g firebase-tools To update to the latest version of the Firebase CLI, re-run the same npm install command. firebase login firebase list se mettre dans le répertoire DynMap firebase init, choisir hosting mettre le fichier, l'appeler index.html firebase deploy

## 6 Mise en application Shiny

### 6.1 TEST 1

---

```
library(shiny)
library(leaflet)
library(shinydashboard)
library(shiny)
library(shinyjs)

Pts.Crd <- spTransform(Poly.Lxd, CRS("+proj=longlat +ellps=WGS84"))

r_colors <- rgb(t(col2rgb(colors()) / 255))
names(r_colors) <- colors()

ui <- dashboardPage(
  dashboardHeader(
    titleWidth = 500,
    title = "Informational Content of Geographical Indications",
    dashboardSidebar(
      width = 75,
      sidebarMenu(
        id = "tabs",
        menuItem("T1", tabName = "T1", icon = icon("angle-right")),
        menuItem("T2", tabName = "T2", icon = icon("angle-right"))
      )
    ),
  dashboardBody(
    tabItems(
```

```

tabItem(
  tabName = "T1",
  fluidRow(
    box(width = 4, # taille de la box
      column(
        width = 12,
        checkboxGroupInput("aoc", "Sélectionner une AOC",
          choices= 1: 5,
          selected= 1: 5, inline= TRUE),
        selectInput("aocld",
          label = "Sélectionner un lieux dit",
          choices = unique(Poly.Lxd$AOCLD),
          selected = 1),
        plotOutput("miplot", width='100%')
      ),
    box(width = 8,
      column(
        width = 12,
        leafletOutput("mymap", height = 700)
      )
    )
  ),
  tabItem(
    tabName = "T2",
    fluidRow(
      box(width = 5,
        column(
          width = 12,
          plotOutput("myplot2",width='100%')
        )
      )
    )
  )
)

server <- function(input, output, session) {
  observe({
    x <- input$aoc
    if (is.null(x)) x <- 1: 5
    updateSelectInput(session, "aocld",
      label= paste("Select input label", length(x)),
      choices= unique(Poly.Lxd$AOCLD[ Poly.Lxd$AOC %in% x])
    )
    getData <- reactive({
      Poly.Lxd$PrdCor[Poly.Lxd$AOC%in% x ]
    })
    getCrD <- reactive({
      coordinates(Pts.Crd[Pts.Crd$AOCLD == input$aocld, ])
    })
    getPts <- reactive({
      SpatialPoints(Pts.Crd[Pts.Crd$AOCLD == input$aocld, ])
    })
    output$mymap <- renderLeaflet({
      tst@map %>%
        setView(getCrD()[ 1], getCrD()[ 2], zoom= 17) %>%
        addCircleMarkers(data = getPts()) })
  })
}

```



```

    output$miplot <- renderPlot({
      plot(density(getData()),
        main= paste0("AOC numero ", paste(x, collapse= ", ")))
      abline(v= Poly.Lxd$PrdCor[Poly.Lxd$AOCLD == input$aocld] )})
    })
  }

shinyApp(ui, server)

```

---

## 6.2 TEST 2

---

```

library(shiny)
library(leaflet)
library(shinydashboard)
library(shiny)
library(shinyjs)
library(ggplot2)
library(plyr)

Pts.Crd <- spTransform(Poly.Lxd, CRS("+proj=longlat +ellps=WGS84"))
r_colors <- rgb(t(col2rgb(colors()) / 255))
names(r_colors) <- colors()

ui <- dashboardPage(
  dashboardHeader(
    titleWidth = 500,
    title = "Informational Content of Geographical Indications"),
  dashboardSidebar(
    width = 75,
    sidebarMenu(
      id = "tabs",
      menuItem("T1", tabName = "T1", icon = icon("angle-right ")),
      menuItem("T2", tabName = "T2", icon = icon("angle-right "))
    )
  ),
  dashboardBody(
    tabItems(
      tabItem(
        tabName = "T1",
        fluidRow(
          box(width = 4, # taille de la box
            column(
              width = 12,
              checkboxGroupInput("aoc", "Sélectionner une AOC",
                choices= 1: 5,
                selected= 1: 5, inline= TRUE),
              selectInput("aocld",
                label = "Sélectionner un lieux dit",
                choices = unique(Poly.Lxd$AOCLD),
                selected = 1),
              plotOutput("miplot", width='100%')
            ),
          box(width = 8,
            column(
              width = 12,

```

```

        leafletOutput("mymap", height = 700),
        fluidRow(verbatimTextOutput("mymap_shape_click"))
      )
    )
  ),
  tabItem(
    tabName = "T2",
    fluidRow(
      box(width = 5,
        column(
          width = 12,
          plotOutput("myplot2",width='100%')
        )
      )
    )
  )
)
)
)
)

## click, mouseover, and mouseout, null before the first click
server <- function(input, output, session) {
  observe({
    x <- input$aoc
    if (is.null(x)) x <- 1: 5
    updateSelectInput(session, "aocId",
      label = paste("Select input label", length(x)),
      choices = unique(Poly.Lxd$AOCID[ Poly.Lxd$AOC %in% x])
    )
    getData <- reactive({## LA DENSITE
      Poly.Lxd$PrdCor[Poly.Lxd$AOC%in% x ]
    })
    getCrd <- reactive({## LE ZOOM
      coordinates(Pts.Crd[Pts.Crd$AOCID == input$aocId, ])
    })
    getPts <- reactive({## LE POINT
      Pts.Crd[Pts.Crd$AOCID == input$aocId, ]
    })
    getClk <- reactive({## LE CLICK
      input$mymap_shape_click$id
    })
    output$mymap <- renderLeaflet({
      tst@map %>%
        setView(getCrd()[ 1], getCrd()[ 2], zoom= 17) %>%
        addCircleMarkers(data= SpatialPoints(getPts()))})
    output$miplot <- renderPlot({
      plot(density(getData()),
        main= paste0("AOC numero ", paste(x, collapse= ", ")))
      abline(v= Poly.Lxd$PrdCor[Poly.Lxd$AOCID == input$aocId])
      abline(v= Poly.Lxd$PrdCor[Poly.Lxd$AOCID == getClk()])})
    })
  }
)

shinyApp(ui, server)

```

## 6.3 TEST 3

---

```
library(shiny)
library(leaflet)
library(shinydashboard)
library(shiny)
library(shinyjs)
library(ggplot2)
library(plyr)

Pts.Crd <- spTransform(Poly.Lxd, CRS("+proj=longlat +ellps=WGS84"))
yop <- c("Coteaux b.", "Bourgogne", "Village", "Premier cru", "Grand cru")
names(yop) <- 1: 5

r_colors <- rgb(t(col2rgb(colors()) / 255))
names(r_colors) <- colors()

ui <- dashboardPage(
  dashboardHeader(
    titleWidth = 500,
    title = "Informational Content of Geographical Indications"),
  dashboardSidebar(
    width = 75,
    sidebarMenu(
      id = "tabs",
      menuItem("T1", tabName = "T1", icon = icon("angle-right ")),
      menuItem("T2", tabName = "T2", icon = icon("angle-right "))
    )
  ),
  dashboardBody(
    tabItems(
      tabItem(
        tabName = "T1",
        fluidRow(
          box(width = 4, # taille de la box
            column(
              width = 12,
              checkboxGroupInput("aoc", "Sélectionner une AOC",
                choices= 1: 5,
                selected= 1: 5, inline= TRUE),
              selectInput("aocld",
                label = "Sélectionner un lieux dit",
                choices = unique(Poly.Lxd$AOCLD),
                selected = 1),
              plotOutput("mipLOT", width='100%'))
            ),
          box(width = 8,
            column(
              width = 12,
              leafletOutput("mymap", height = 700),
              fluidRow(verbatimTextOutput("mymap_shape_click"))
            )
          )
        )
      ),
      tabItem(
        tabName = "T2",
```

```

    fluidRow(
      box(width = 5,
        column(
          width = 12,
          plotOutput("myplot2",width='100%')
        )
      )
    )
  )
}

## click, mouseover, and mouseout, null before the first click
server <- function(input, output, session) {
  observe({
    x <- input$aoc
    if (is.null(x)) x <- 1: 5
    updateSelectInput(session, "aocId",
      label = paste("Select input label", length(x)),
      choices = unique(Poly.Lxd$AOCID[ Poly.Lxd$AOC %in% x])
    )
    getData <- reactive({## LA DENSITE
      Poly.Lxd@data[Poly.Lxd$AOC%in% x ,]
    })
    getCrD <- reactive({## LE ZOOM
      coordinates(Pts.Crd[Pts.Crd$AOCID == input$aocId, ])
    })
    getPts <- reactive({## LE POINT
      Pts.Crd[Pts.Crd$AOCID == input$aocId, ]
    })
    getClk <- reactive({## LE CLICK
      input$mymap_shape_click$id
    })
    output$mymap <- renderLeaflet({
      tst@map %>%
        setView(getCrD()[ 1], getCrD()[ 2], zoom= 17) %>%
        addCircleMarkers(data= SpatialPoints(getPts()))})
    output$miplot <- renderPlot({
      ggplot(getData(), aes(x= revalue(factor(AOC), yop),
        y= PrdCor, fill= factor(AOC)))+
      geom_violin(trim=FALSE)+ theme_minimal()+ ylim(40, 100)+
      geom_boxplot(width=0.1, fill="white")+
      labs(title= paste("Reference: ", input$aocId,
        " ; Current: ", getClk()),
        x= "", y= "100-Point Vineyard Quality Score")+
      scale_fill_manual(values= pal(5))+
      theme(legend.position= "none")+
      scale_x_discrete(expand= expand_scale(mult= 0, add= 1),drop=T)+
      geom_hline(yintercept= Poly.Lxd$PrdCor[Poly.Lxd$AOCID == input$aocId],
        lty= 3, col= "red")+
      geom_hline(yintercept= Poly.Lxd$PrdCor[Poly.Lxd$AOCID == getClk()], lty= 2, col= "blue")+
      annotate("text", x= 0.4,
        y= Poly.Lxd$PrdCor[Poly.Lxd$AOCID == input$aocId]+ 2,
        label = "Reference", col= "red")+
      annotate("text", x= length(x)+ .6,
        y= ifelse(is.null(input$mymap_shape_click), -100,
          Poly.Lxd$PrdCor[Poly.Lxd$AOCID==getClk()])+
          2, label = "Current", col= "blue")
    })
  })
}

```

```

    })
  })
})

shinyApp(ui, server)

```

---

## 6.4 TEST 4

---

```

library(shiny)
library(leaflet)
library(shinydashboard)
library(shiny)
library(shinyjs)
library(ggplot2)
library(plyr)

Pts.Crd <- spTransform(Poly.Lxd, CRS("+proj=longlat +ellps=WGS84"))
yop <- c("Coteaux b.", "Bourgogne", "Village", "Premier cru", "Grand cru")
names(yop) <- 1: 5

r_colors <- rgb(t(col2rgb(colors()) / 255))
names(r_colors) <- colors()

ui <- dashboardPage(
  dashboardHeader(
    titleWidth = 500,
    title = "Informational Content of Geographical Indications"),
  dashboardSidebar(
    width = 75,
    sidebarMenu(
      id = "tabs",
      menuItem("T1", tabName = "T1", icon = icon("angle-right ")),
      menuItem("T2", tabName = "T2", icon = icon("angle-right "))
    )
  ),
  dashboardBody(
    tabItems(
      tabItem(
        tabName = "T1",
        fluidRow(
          box(width = 5,
            column(
              width = 12,
              checkboxGroupInput("aoc", "Sélectionner une AOC",
                choices= 1: 5,
                selected= 1: 5, inline= TRUE),
              selectInput("aocld",
                label = "Sélectionner un lieux dit",
                choices = unique(Poly.Lxd$AOCLD),
                selected = 1),
              plotOutput("mipLOT", width='100%'),
              tableOutput('table'))
            ),
          box(width = 7,
            column(

```

```

        width = 12,
        leafletOutput("mymap", height = 700),
        fluidRow(verbatimTextOutput("mymap_shape_click"))
      )
    )
  ),
  tabItem(
    tabName = "T2",
    fluidRow(
      box(width = 5,
        column(
          width = 12,
          plotOutput("myplot2",width='100%')
        )
      )
    )
  )
)

## click, mouseover, and mouseout, null before the first click
server <- function(input, output, session) {
  observe({
    x <- input$aoc
    if (is.null(x)) x <- 1: 5
    updateSelectInput(session, "aocId",
      label = paste("Select input label", length(x)),
      choices = unique(Poly.Lxd$AOCID[ Poly.Lxd$AOC %in% x])
    )
    getData <- reactive({## LA DENSITE
      Poly.Lxd@data[Poly.Lxd$AOC%in% x ,]
    })
    getCrD <- reactive({## LE ZOOM
      coordinates(Pts.Crd[Pts.Crd$AOCID == input$aocId, ])
    })
    getPts <- reactive({## LE POINT
      Pts.Crd[Pts.Crd$AOCID == input$aocId, ]
    })
    getClk <- reactive({## LE CLICK
      input$mymap_shape_click$id
    })
    output$mymap <- renderLeaflet({
      tst@map %>%
        setView(getCrD()[ 1], getCrD()[ 2], zoom= 17) %>%
        addCircleMarkers(data= SpatialPoints(getPts()))})
    output$table <- renderTable(
      data.frame("Vineyard"= c("Reference", "Current"),
        "Coteaux b."= c(100, NA),
        "Bourgogne"= c(100, NA),
        "Village"= c(100, NA),
        "Premier Cru"= c(100, NA),
        "Grand Cru"= c(100, NA)))
    output$miplot <- renderPlot({
      ggplot(getData(), aes(x= revalue(factor(AOC), yop),
        y= PrdCor, fill= factor(AOC)))+
        geom_violin(trim=FALSE)+ theme_minimal()+ ylim(40, 100)+
        geom_boxplot(width=0.1, fill="white")+

```

```

labs(title= paste("Reference: ", input$aocld,
                  " ; Current: ", getClk()),
      x= "", y = "100-Point Vineyard Quality Score")+
scale_fill_manual(values= pal(5))+
theme(legend.position= "none")+
scale_x_discrete(expand= expand_scale(mult= 0, add= 1),drop=T)+
geom_hline(yintercept= Poly.Lxd$PrdCor[Poly.Lxd$AOCLD == input$aocld],
           lty= 3, col= "red")+
geom_hline(yintercept= Poly.Lxd$PrdCor[Poly.Lxd$AOCLD == getClk()], lty= 2, col= "blue")+
annotate("text", x= 0.4,
          y= Poly.Lxd$PrdCor[Poly.Lxd$AOCLD == input$aocld]+ 2,
          label = "Reference", col= "red")+
annotate("text", x= length(x)+ .6,
          y= ifelse(is.null(input$mymap_shape_click), -100,
                    Poly.Lxd$PrdCor[Poly.Lxd$AOCLD==getClk()])+
            2, label = "Current", col= "blue")
})
})
})

shinyApp(ui, server)

```

---

## 6.5 TEST 5

---

```

library(shiny)
library(leaflet)
library(shinydashboard)
library(shinyjs)
library(ggplot2)
library(plyr)

Pts.Crd <- spTransform(Poly.tst, CRS("+proj=longlat +ellps=WGS84"))

ui <- dashboardPage(
  dashboardHeader(
    titleWidth = 500,
    title = "Informational Content of Geographical Indications"),
  dashboardSidebar(
    width = 75,
    sidebarMenu(
      id = "tabs",
      menuItem("T1", tabName = "T1", icon = icon("angle-right ")),
      menuItem("T2", tabName = "T2", icon = icon("angle-right "))
    )
  ),
  dashboardBody(
    tabItems(
      tabItem(
        tabName = "T1",
        fluidRow(
          box(width = 5,
              column(
                width = 12,
                selectInput("niveau",
                           label= "Sélectionner un niveau",

```

```

        choices= unique(Poly.tst$NIVEAU),
        selected = 1),
    selectInput("commune",
        label = "Sélectionner une commune",
        choices = unique(Poly.tst$LIBCOM),
        selected = 1),
    selectInput("name",
        label = "Sélectionner un lieu dit",
        choices = unique(Poly.tst$NAME),
        selected = 1),
    plotOutput("miplot", width='100%'),
    tableOutput('table'))
  ),
  box(width = 7,
    column(
      width = 12,
      leafletOutput("mymap", height = 700),
      fluidRow(verbatimTextOutput("mymap_shape_click"))
    )
  )
),
tabItem(
  tabName = "T2",
  fluidRow(
    box(width = 5,
      column(
        width = 12,
        plotOutput("myplot2",width='100%')
      )
    )
  )
)
)
)
)
)
)
)

## click, mouseover, and mouseout, null before the first click
server <- function(input, output, session) {
  observe({
    updateSelectInput(session, "commune",
      choices = unique(Poly.tst$LIBCOM[ Poly.tst$NIVEAU %in% input$niveau])
    )
    updateSelectInput(session, "name",
      choices = unique(Poly.tst$NAME[ Poly.tst$NIVEAU %in% input$niveau &
        Poly.tst$LIBCOM %in% input$commune ])
    )
  })
  getCrd <- reactive({## LE ZOOM
    coordinates(Pts.Crd[Pts.Crd$NIVEAU == input$niveau &
      Pts.Crd$LIBCOM == input$commune &
      Pts.Crd$NAME == input$name , ])
  })
  getPts <- reactive({## LE POINT
    Pts.Crd[Pts.Crd$NIVEAU == input$niveau &
      Pts.Crd$LIBCOM == input$commune &
      Pts.Crd$NAME == input$name , ]
  })
  getClk <- reactive({## LE CLICK
    input$mymap_shape_click$id

```



```

})
output$mymap <- renderLeaflet({
  tst@map %>%
    setView(getCrd()[ 1], getCrd()[ 2], zoom= 17) %>%
    addCircleMarkers(data= SpatialPoints(getPts()))})
output$table <- renderTable(
  data.frame("Vineyard"= c("Reference", "Current"),
    "Coteaux b."= c(100, NA),
    "Bourgogne"= c(100, NA),
    "Village"= c(100, NA),
    "Premier Cru"= c(100, NA),
    "Grand Cru"= c(100, NA)))
output$mipplot <- renderPlot({
  ggplot(Poly.tst@data, aes(x= factor(NIVEAU),
    y= PrdCor, fill= factor(NIVEAU)))+
    geom_violin(trim=FALSE)+ theme_minimal()+ ylim(40, 100)+
    geom_boxplot(width=0.1, fill="white")+
    labs(title= paste("Reference: ", input$name,
      " ; Current: ", getClk()),
      x= "", y= "100-Point Vineyard Quality Score")+
    scale_fill_manual(values= AocPal)+
    theme(legend.position= "none")+
    scale_x_discrete(expand= expand_scale(mult= 0, add= 1),drop=T)+
    geom_hline(yintercept=
      Poly.tst$PrdCor[Poly.tst$NIVEAU %in% input$niveau &
        Poly.tst$LIBCOM %in% input$commune &
        Poly.tst$NAME %in% input$name ],
      lty= 3, col= "red")+
    geom_hline(yintercept= Poly.tst$PrdCor[Poly.tst$xx== getClk()],
      lty= 2, col= "blue")+
    annotate("text", x= 0.4,
      y= Poly.tst$PrdCor[Poly.tst$NIVEAU %in% input$niveau &
        Poly.tst$LIBCOM %in% input$commune &
        Poly.tst$NAME %in% input$name ]+
        2, label = "Reference", col= "red")+
    annotate("text", x= 5.6,
      y= ifelse(is.null(input$mymap_shape_click), -100,
        Poly.tst$PrdCor[Poly.tst$xx== getClk()])+
        2, label = "Current", col= "blue")
  })
})
shinyApp(ui, server)

```

---

## 6.6 TEST 6

---

```

library(shiny)
library(leaflet)
library(shinydashboard)
library(shinyjs)
library(ggplot2)
library(plyr)

```

```

Pts.Crd <- spTransform(Poly.tst, CRS("+proj=longlat +ellps=WGS84"))

```

```

ui <- dashboardPage(
  dashboardHeader(
    titleWidth = 700,
    title = "Essai de classification continue des vignobles de la Côte d'Or"),
  dashboardSidebar(
    width = 75,
    sidebarMenu(
      id = "tabs",
      menuItem("T1", tabName = "T1", icon = icon("angle-right ")),
      menuItem("T2", tabName = "T2", icon = icon("angle-right "))
    )
  ),
  dashboardBody(
    tabItems(
      tabItem(
        tabName = "T1",
        fluidRow(
          box(width = 5,
            column(
              width = 12,
              selectInput("niveau",
                label=
                  "Niveau de l'appellation de référence",
                choices= unique(Poly.tst$NIVEAU),
                selected= 1),
              selectInput("commune",
                label=
                  "Commune de la parcelle de référence",
                choices= unique(Poly.tst$LIBCOM),
                selected= 1),
              selectInput("name",
                label=
                  "Lieu dit de la parcelle de référence",
                choices= unique(Poly.tst$NAME),
                selected = 1),
              plotOutput("miplot", width='100%'),
              tableOutput('table'))
            ),
          box(width = 7,
            column(
              width = 12,
              leafletOutput("mymap", height = 700),
              fluidRow(verbatimTextOutput("mymap_shape_click"))
            )
          )
        ),
      tabItem(
        tabName = "T2",
        fluidRow(
          box(width = 5,
            column(
              width = 12,
              plotOutput("myplot2",width='100%')
            )
          )
        )
      )
    )
  )
)

```

```

    )
  )
)

## click, mouseover, and mouseout, null before the first click
server <- function(input, output, session) {
  observe({
    updateSelectInput(session, "commune",
      choices = unique(Poly.tst$LIBCOM[ Poly.tst$NIVEAU %in% input$niveau])
    })
    observe({
      updateSelectInput(session, "name",
        choices = unique(Poly.tst$NAME[ Poly.tst$NIVEAU %in% input$niveau &
          Poly.tst$LIBCOM %in% input$commune ])
      })
    observe({
      yop <- Poly.tst$PrdCor[Poly.tst$NIVEAU %in% input$niveau &
        Poly.tst$LIBCOM %in% input$commune &
        Poly.tst$NAME %in% input$name ]
      getCrđ <- reactive({## LE ZOOM
        coordinates(Pts.Crd[Pts.Crd$NIVEAU == input$niveau &
          Pts.Crd$LIBCOM == input$commune &
          Pts.Crd$NAME == input$name , ])
      })
      getPts <- reactive({## LE POINT
        Pts.Crd[Pts.Crd$NIVEAU == input$niveau &
          Pts.Crd$LIBCOM == input$commune &
          Pts.Crd$NAME == input$name , ]
      })
      getClk <- reactive({## LE CLICK
        input$mymap_shape_click$id
      })
      yap <- Poly.tst$PrdCor[Poly.tst$xx== getClk()]
      output$mymap <- renderLeaflet({
        tst@map %>%
          setView(getCrđ()[ 1], getCrđ()[ 2], zoom= 17) %>%
          addCircleMarkers(data= SpatialPoints(getPts()))})
      output$table <- renderTable(
        data.frame("Meilleur que x pc des"= c("Reference", "Current"),
          "Coteaux b."= round(c(mean(
            Poly.tst$PrdCor[Poly.tst$NIVEAU==
              "Coteaux b."< yop)* 100, mean(
            Poly.tst$PrdCor[Poly.tst$NIVEAU==
              "Coteaux b."< yap)* 100), 2),
          "Bourgogne"= round(c(mean(
            Poly.tst$PrdCor[Poly.tst$NIVEAU==
              "Bourgogne"< yop)* 100, mean(
            Poly.tst$PrdCor[Poly.tst$NIVEAU==
              "Bourgogne"< yap)* 100), 2),
          "Village"= round(c(mean(
            Poly.tst$PrdCor[Poly.tst$NIVEAU==
              "Village"< yop)* 100, mean(
            Poly.tst$PrdCor[Poly.tst$NIVEAU==
              "Village"< yap)* 100), 2),
          "Premier cru"= round(c(mean(
            Poly.tst$PrdCor[Poly.tst$NIVEAU==
              "Premier cru"< yop)* 100, mean(
            Poly.tst$PrdCor[Poly.tst$NIVEAU==
              "Premier cru"< yap)* 100), 2),

```

```

      "Grand cru" = round(c(mean(
        Poly.tst$PrdCor[Poly.tst$NIVEAU==
          "Grand cru"] < yop) * 100, mean(
        Poly.tst$PrdCor[Poly.tst$NIVEAU==
          "Grand cru"] < yap) * 100), 2)))
output$miplot <- renderPlot({
  ggplot(Poly.tst@data, aes(x= factor(NIVEAU),
    y= PrdCor, fill= factor(NIVEAU)))+
  geom_violin(trim=FALSE)+ theme_minimal()+ ylim(40, 100)+
  geom_boxplot(width=0.1, fill="white")+
  labs(title= paste("Référence: ", input$name,
    " ; Current: ",
    Poly.tst$NAME[Poly.tst$xx== getClk()]),
    x= "", y= "100-Point Vineyard Quality Score")+
  scale_fill_manual(values= AocPal)+
  theme(legend.position= "none")+
  scale_x_discrete(expand= expand_scale(mult= 0, add= 1), drop=T)+
  geom_hline(yintercept= yop, lty= 3, col= "red")+
  geom_hline(yintercept= yap, lty= 2, col= "blue")+
  annotate("text", x= 0.4,
    y= yop+ 2, label= "Référence", col= "red")+
  annotate("text", x= 5.6,
    y= ifelse(is.null(input$mymap_shape_click),
      -100, yap)+ 2, label="Current", col= "blue")
})
})
})

shinyApp(ui, server)

```

---

## 6.7 Global

---

```

library(dygraphs)
library(shinydashboard)
library(shiny)
library(shinyjs)
library(mapview)
library(datasets)
library(dplyr)
library(leaflet)
#
latent <- Geo.Ras@data
#
# print(getwd())
# print(test)
#
my_map <-
  leaflet(Poly.Lxd) %>% addTiles() %>% addMouseCoordinates()

# #
# source("./ui.R")
# source("./server.R")
# #

```

```
shinyApp(ui, server)
```

---

## 6.8 Server

---

```
server <- shinyServer(function(input, output, session) {
  getData <- reactive({ latent[, as.character(input$prdd)] })
  output$myplot <- renderPlot({
    input$pprd
  }) # créer l'objet à plot => pour PlotOutput côté server
  output$myplot2 <- renderPlot({
    plot(density(getData()))
  }) # créer l'objet à plot => pour PlotOutput côté server
  output$mymap <- renderLeaflet({
    tst@map
  })
})
})
```

---

## 6.9 ui

---

```
ui <- dashboardPage(
  dashboardHeader(
    titleWidth = 150,
    title = "TITRE"),
  dashboardSidebar(width = 75,
    sidebarMenu(id = "tabs",
      menuItem("T1", tabName = "T1",
        icon = icon("angle-right ")),
      menuItem("T2", tabName = "T2",
        icon = icon("angle-right "))),
  dashboardBody(
    tabItems(tabItem(tabName = "T1",
      fluidRow(
        box(width= 2,
          column(width = 12,
            selectInput("prdd",
              label = "Correction",
              choices=
                c("G900raw", "G900cor"),
              selected = 1))),
        box(width= 5,
          column(width = 12,
            plotOutput("myplot")))),
        # leafletOutput("mymap")))),
    tabItem(tabName = "T2",
      fluidRow(
        box(width = 5,
          column(width= 12,
            plotOutput("myplot2")))),
        fluidRow(
          box(width = 5,
            column(width = 12,
```

```
plotOutput("myplot2"))))))))
```

---

## 7 Conclusion

Le chiffres d'affaire des signes de qualité c'est 32 milliards d'euros et le budget de l'INAO 32 millions d'euros, c'est un millième du chiffre d'affaires.

---

```
sessionInfo()
```

---

## 8 Bibliographie

### References

Wolikow, S. and Jacquet, O. (2011). *Territoires et terroirs du vin du XVIIIe au XXIe siècles*. Éditions Universitaires de Dijon.

## A Annexes

### A.1 Annexe 1: incohérence des AOC

---

```
as.vector(Geo.CDem$IDU[Geo.CDem$AOC== "BGOR" & rowSums(Geo.CDem@data[, c(24, 26: 29)])> 0])
as.vector(Geo.CDem$IDU[Geo.CDem$AOC== "BOUR" & rowSums(Geo.CDem@data[, 26: 29])> 0])
as.vector(Geo.CDem$IDU[Geo.CDem$AOC== "VILL" & rowSums(Geo.CDem@data[, c(26, 29)])> 0])
as.vector(Geo.CDem$IDU[Geo.CDem$AOC== "PCRU" & Geo.CDem@data[, 26]> 0])
```

---

```
[1] "21412000AZ0139" "21464000AN0094" "21492000AR0011"
```

```
[4] "21492000BN0045" "215690000C0840"
```

```
[1] "210370000A0507" "21110000AK0116" "21150000AM0096"
```

```
[4] "21428000AA0019" "21582000BC0069"
```

```
[1] "21037000AH0094" "21037000AH0096" "21110000AM0101"
```

```
[4] "21133000AB0401" "21133000AC0005" "21133000AC0003"
```

```
[7] "21133000AC0002" "21133000AC0004" "21512000AE0292"
```

```
[10] "21582000AL0049"
```

```
[1] "21442000AB0315"
```

### A.2 Annexe 2: les intitulés pédologiques

Pour retrouver les intitulés des UCS, nous utilisons le site web <https://bourgogne.websol.fr/carto> où les différents types de sols qui composent les UCS sont consultables. Le travail manuel a consisté à extraire les coordonnées Lambert 93 d'au moins une parcelle par UCS et d'aller chercher sur le site le nom de l'UCS correspondante. Nous voyons également que lorsque l'UCS est un numéro manquant c'est qu'il s'agit de sols artificialisés (Chenôve, Nuits et Beaune). Il y a un léger effet frontière au sud sur les valeurs qui ne sont pas appariées.

---

```
yy <- data.frame(coordinates(GCDtmp3), GCDtmp3$NOUC)
yy[!duplicated(GCDtmp3$NOUC), ]
plot(GCDtmp3)
plot(GCDtmp3[GCDtmp3$NOUC== "0",], col= "blue", add= T, pch= 20)
```

---