

Modélisation et catégorisation du potentiel viral d'un objet culturel en ligne et du profil des individus selon les intentions de partage

Code

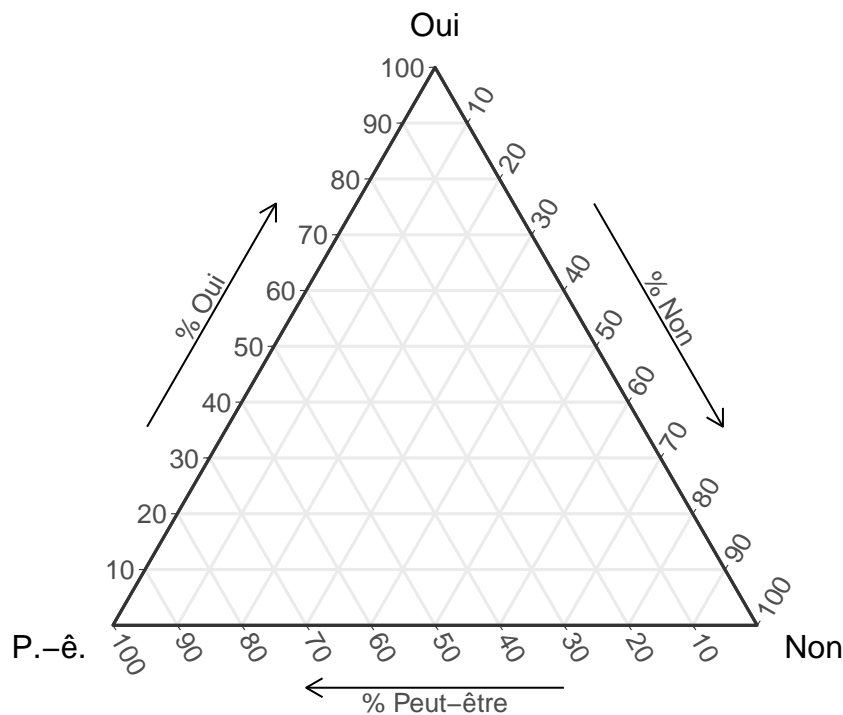
Ugo Roux, IMSIC, Aix-Marseille Université

Version du 06/10/2020

Le présent document propose le code pour générer des diagrammes ternaires qui ont pour objectif de modéliser et de catégoriser le potentiel viral d'un objet culturel en ligne ainsi que le profil des individus selon les intentions de partage. Il accompagne, telle une annexe, notre article qui présente le contexte de développement et d'utilisation de cet outil (Roux 2020).

Les différents diagrammes ternaires présentés dans ce documents ont été générés avec le logiciel *R* (R Development Core Team 2020) et l'environnement de développement intégré *RStudio* (RStudio 2020) grâce à la bibliothèque *ggtern* (Hamilton n.d., 2020a, 2020b; Hamilton and Ferry 2018). Le code utilisé pour générer ces diagrammes est inspiré du code écrit par Nicholas E. Hamilton pour reproduire le diagramme de classification de la texture des sols selon Shepard (Hamilton 2014).

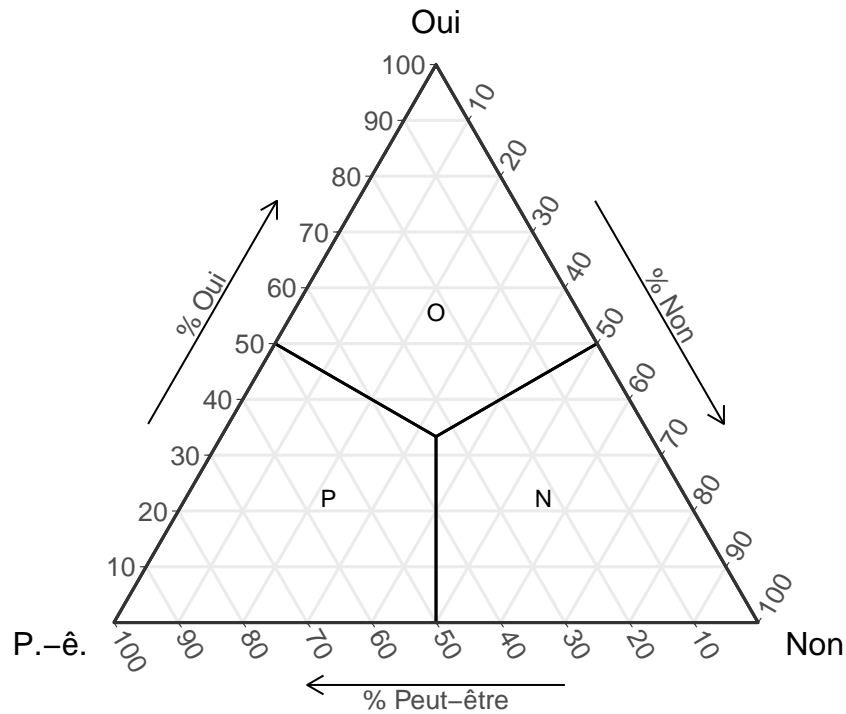
Diagramme ternaire



Diagrammes ternaires

Diagramme ternaire 3 zones

Diagramme ternaire 3 zones



```
#Créer les points du diagramme
points3 <- data.frame(
  rbind(
    c(1,1.000,0.000,0.000),
    c(2,0.500,0.500,0.000),
    c(3,0.500,0.000,0.500),
    c(4,0.500,0.500,0.500),
    c(5,0.000,1.000,0.000),
    c(6,0.000,0.500,0.500),
    c(7,0.000,0.000,1.000)
  )
)
colnames(points3)=c("IDPoint","T","L","R")

#Attribuer à chaque polygone un nombre et une étiquette
polygon.labels3 <- data.frame(Label3=c("O","P","N"))
polygon.labels3$IDLabel=1:nrow(polygon.labels3)

#Créer une carte des polygones
polygons3 <- data.frame(
  rbind(
    c(1,1),c(1,2),c(1,4),c(1,3),
```

```

      c(2,2),c(2,4),c(2,6),c(2,5),
      c(3,3),c(3,7),c(3,6),c(3,4)
    )
  )
polygons3$PointOrder <- 1:nrow(polygons3)
colnames(polygons3)=c("IDLabel","IDPoint","PointOrder")

#Fusionner les trois précédents sets en un seul
df3 <- merge(polygons3,points3)
df3 <- merge(df3,polygon.labels3)
df3 <- df3[order(df3$PointOrder),]

#Déterminer les données des étiquettes
Labs3=ddply(df3,"Label3",function(x){c(c(mean(x$T),mean(x$L),mean(x$R)))})
colnames(Labs3)=c("Label","T","L","R")

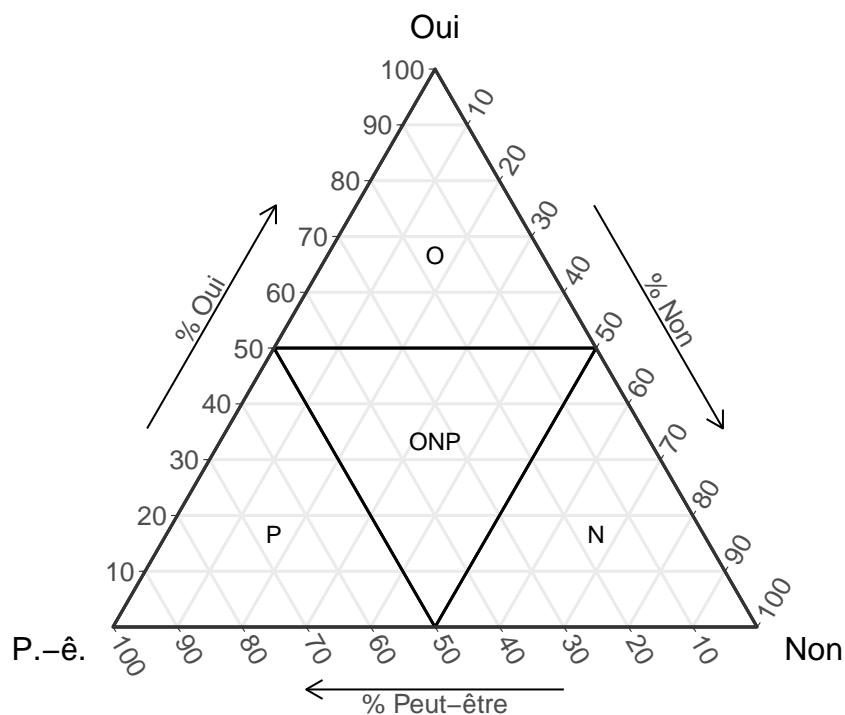
#Créer le diagramme
base3 <- ggtern(data=df3,aes(L,T,R)) +
  geom_polygon(aes(group=Label3),color="black",alpha=0) +
  geom_text(data=Labs3,aes(label=Label),size=3,color="black") +
  theme_bw() +
  theme(plot.title=element_text(hjust=0.5)) +
  tern_limits(labels=c(10,20,30,40,50,60,70,80,90,100),breaks=seq(0.1,1,by=0.1)) +
  theme_clockwise() +
  theme_showarrows() +
  labs(
    title="Diagramme ternaire 3 zones",
    T="Oui",L="P.-ê.",R="Non",
    Tarrow="% Oui",Larrow="% Peut-être",Rarrow="% Non"
  ) +
  theme(tern.axis.arrow=element_line(size=1,color="black"))

#Générer le diagramme
plot(base3)

```

Diagramme ternaire 4 zones

Diagramme ternaire 4 zones



```
#Créer les points du diagramme
points4 <- data.frame(
  rbind(
    c(1,1.000,0.000,0.000),
    c(2,0.500,0.500,0.000),
    c(3,0.500,0.000,0.500),
    c(4,0.000,1.000,0.000),
    c(5,0.000,0.500,0.500),
    c(6,0.000,0.000,1.000)
  )
)
colnames(points4)=c("IDPoint","T","L","R")

#Attribuer à chaque polygone un nombre et une étiquette
polygon.labels4 <- data.frame(Label4=c("O","ONP","N","P"))
polygon.labels4$IDLabel=1:nrow(polygon.labels4)

#Créer une carte des polygones
polygons4 <- data.frame(
  rbind(
    c(1,1),c(1,2),c(1,3),
    c(2,2),c(2,5),c(2,3),
    c(3,3),c(3,5),c(3,6),
    c(4,4),c(4,5),c(4,2)
  )
)
```

```

)
)
polygons4$PointOrder <- 1:nrow(polygons4)
colnames(polygons4)=c("IDLabel","IDPoint","PointOrder")

#Fusionner les trois précédents sets en un seul
df4 <- merge(polygons4,points4)
df4 <- merge(df4,polygon.labels4)
df4 <- df4[order(df4$PointOrder),]

#Déterminer les données des étiquettes
Labs4=ddply(df4,"Label4",function(x){c(c(mean(x$T),mean(x$L),mean(x$R)))})
colnames(Labs4)=c("Label","T","L","R")

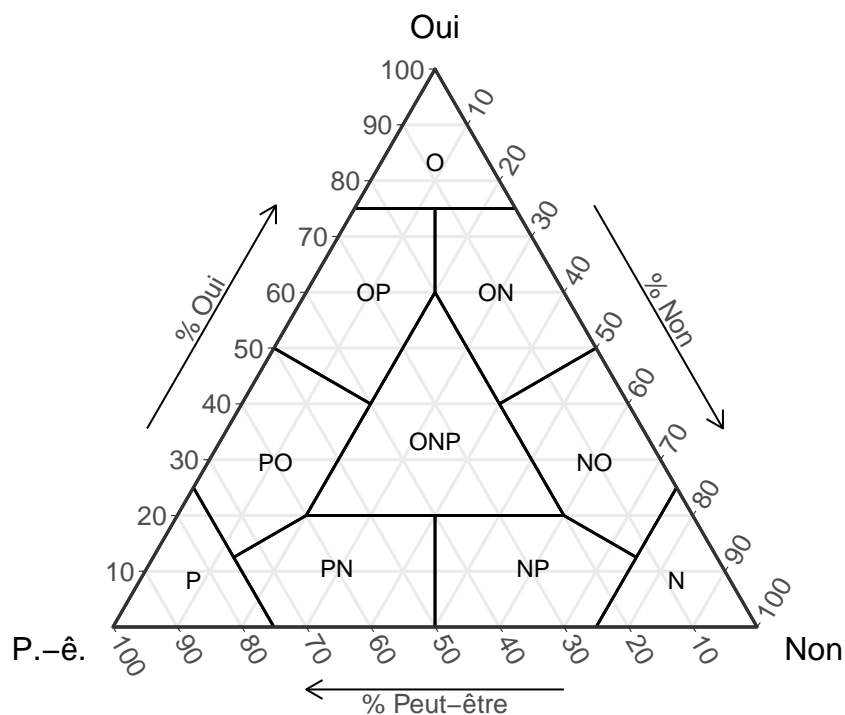
#Créer le diagramme
base4 <- ggtern(data=df4,aes(L,T,R)) +
  geom_polygon(aes(group=Label4),color="black",alpha=0) +
  geom_text(data=Labs4,aes(label=Label),size=3,color="black") +
  theme_bw() +
  theme(plot.title=element_text(hjust=0.5)) +
  tern_limits(labels=c(10,20,30,40,50,60,70,80,90,100),breaks=seq(0.1,1,by=0.1)) +
  theme_clockwise() +
  theme_showarrows() +
  labs(
    title="Diagramme ternaire 4 zones",
    T="Oui",L="P.-ê.",R="Non",
    Tarrow="% Oui",Larrow="% Peut-être",Rarrow="% Non"
  ) +
  theme(tern.axis.arrow=element_line(size=1,color="black"))

#Générer le diagramme
plot(base4)

```

Diagramme ternaire 10 zones

Diagramme ternaire 10 zones



#Créer les points du diagramme

```
points10 <- data.frame(  
  rbind(  
    c(1,1.000,0.000,0.000),  
    c(2,0.750,0.250,0.000),  
    c(3,0.750,0.125,0.125),  
    c(4,0.750,0.000,0.250),  
    c(5,0.600,0.200,0.200),  
    c(6,0.500,0.500,0.000),  
    c(7,0.500,0.000,0.500),  
    c(8,0.400,0.400,0.200),  
    c(9,0.400,0.200,0.400),  
    c(10,0.250,0.750,0.000),  
    c(11,0.250,0.000,0.750),  
    c(12,0.200,0.600,0.200),  
    c(13,0.200,0.400,0.400),  
    c(14,0.200,0.200,0.600),  
    c(15,0.125,0.750,0.125),  
    c(16,0.125,0.125,0.750),  
    c(17,0.000,1.000,0.000),  
    c(18,0.000,0.750,0.250),  
    c(19,0.000,0.500,0.500),  
    c(20,0.000,0.250,0.750),  
    c(21,0.000,0.000,1.000)
```

```

)
)
colnames(points10) = c("IDPoint", "T", "L", "R")

#Attribuer à chaque polygone un nombre et une étiquette
polygon.labels10 <- data.frame(Label10=c("O", "OP", "ON", "ONP", "PO", "NO", "P", "PN", "NP", "N"))
polygon.labels10$IDLabel=1:nrow(polygon.labels10)

#Créer une carte des polygones
polygons10 <- data.frame(
  rbind(
    c(1,1),c(1,2),c(1,4),
    c(2,6),c(2,2),c(2,3),c(2,5),c(2,8),
    c(3,3),c(3,4),c(3,7),c(3,9),c(3,5),
    c(4,5),c(4,14),c(4,12),
    c(5,6),c(5,8),c(5,12),c(5,15),c(5,10),
    c(6,7),c(6,11),c(6,16),c(6,14),c(6,9),
    c(7,17),c(7,10),c(7,18),
    c(8,15),c(8,12),c(8,13),c(8,19),c(8,18),
    c(9,13),c(9,14),c(9,16),c(9,20),c(9,19),
    c(10,11),c(10,21),c(10,20)
  )
)
polygons10$PointOrder <- 1:nrow(polygons10)
colnames(polygons10)=c("IDLabel", "IDPoint", "PointOrder")

#Fusionner les trois précédents sets en un seul
df10 <- merge(polygons10,points10)
df10 <- merge(df10,polygon.labels10)
df10 <- df10[order(df10$PointOrder),]

#Déterminer les données des étiquettes
Labs10=ddply(df10, "Label10", function(x){c(mean(x$T),mean(x$L),mean(x$R))})
colnames(Labs10)=c("Label", "T", "L", "R")

#Créer le diagramme
base10 <- ggtern(data=df10,aes(L,T,R)) +
  geom_polygon(aes(group=Label10),color="black",alpha=0) +
  geom_text(data=Labs10,aes(label=Label),size=3,color="black") +
  theme_bw() +
  theme(plot.title=element_text(hjust=0.5)) +
  tern_limits(labels=c(10,20,30,40,50,60,70,80,90,100),breaks=seq(0.1,1,by=0.1)) +
  theme_clockwise() +
  theme_showarrows() +
  labs(
    title="Diagramme ternaire 10 zones",
    T="Oui",L="P.-ê.",R="Non",
    Tarrow="% Oui",Larrow="% Peut-être",Rarrow="% Non"
  ) +
  theme(tern.axis.arrow=element_line(size=1,color="black"))

#Générer le diagramme
plot(base10)

```

Exploitation des diagrammes ternaires

Les données utilisées pour nos démonstrations sont disponibles en *open access* selon les termes prévus dans la licence *Creative Commons Attribution 4.0 International* à l'adresse suivante : <https://zenodo.org/record/3350487> (Roux 2019a). Le lecteur est invité à lire également l'article qui présente ces données (Roux 2019b).

Jeu de données

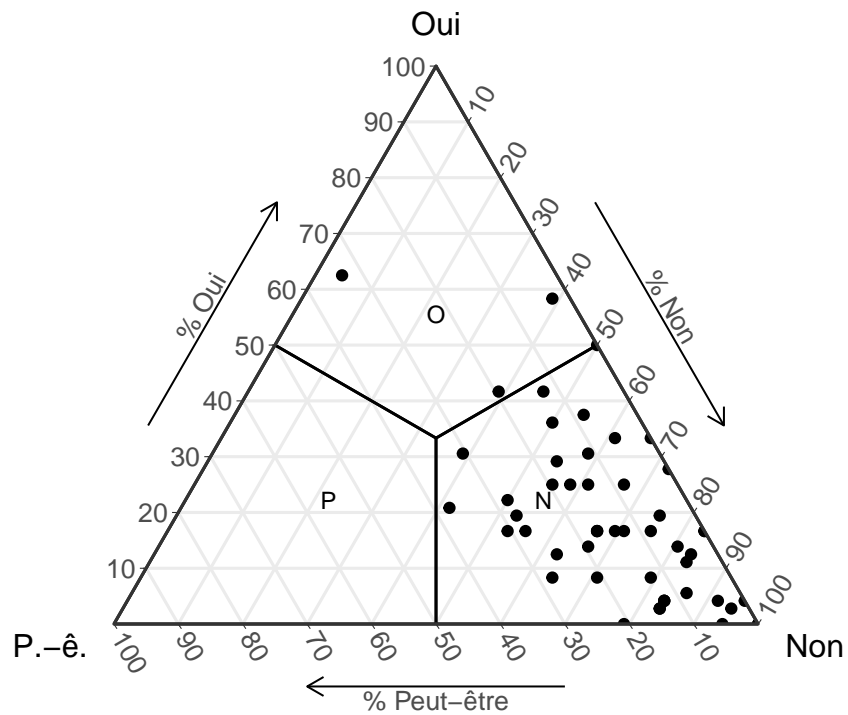
```
#Jeu de données
```

```
dfhf <- data.frame(  
  O=c(  
    11.11, 33.33, 0, 36.11, 33.33, 2.78, 13.89, 19.44, 2.78, 0,  
    16.67, 25, 30.56, 41.67, 8.33, 4.17, 25, 62.5, 0, 16.67,  
    37.5, 12.5, 25, 41.67, 16.67, 4.17, 16.67, 16.67, 16.67, 13.89,  
    19.44, 5.56, 30.56, 22.22, 58.33, 2.78, 16.67, 8.33, 0, 25,  
    27.78, 12.5, 50, 16.67, 20.83, 4.17, 4.17, 8.33, 29.17, 4.17,  
    0, 16.67, 16.67  
  ),  
  N=c(  
    83.33, 61.11, 94.44, 50, 66.67, 83.33, 80.56, 52.78, 83.33, 100,  
    69.44, 61.11, 58.33, 38.89, 79.17, 83.33, 66.67, 4.17, 79.17, 66.67,  
    54.17, 62.5, 58.33, 45.83, 66.67, 83.33, 83.33, 55.56, 52.78, 66.67,  
    75, 86.11, 38.89, 50, 38.89, 94.44, 75, 63.89, 100, 55.56,  
    72.22, 83.33, 50, 83.33, 41.67, 95.83, 91.67, 70.83, 54.17, 83.33,  
    100, 66.67, 70.83  
  ),  
  P=c(  
    5.56, 5.56, 5.56, 13.89, 0, 13.89, 5.56, 27.78, 13.89, 0,  
    13.89, 13.89, 11.11, 19.44, 12.5, 12.5, 8.33, 33.33, 20.83,  
    16.67, 8.33, 25, 16.67, 12.5, 16.67, 12.5, 0, 27.78, 30.56,  
    19.44, 5.56, 8.33, 30.56, 27.78, 2.78, 2.78, 8.33, 27.78, 0,  
    19.44, 0, 4.17, 0, 0, 37.5, 0, 4.17, 20.83, 16.67, 12.5,  
    0, 16.67, 12.5)  
)
```


Affichage des données sous forme de points

Diagramme ternaire 3 zones avec points

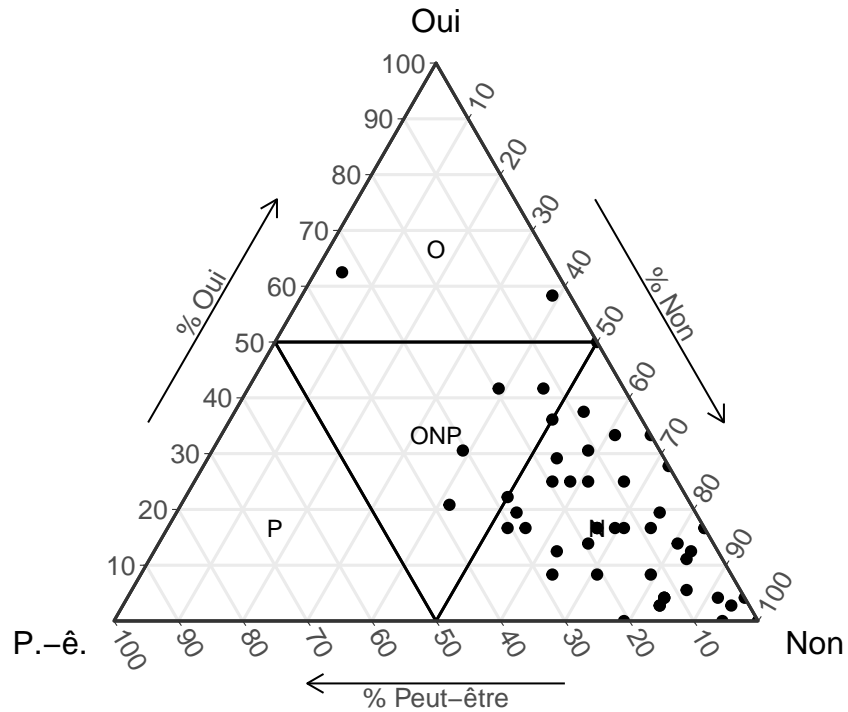
Diagramme ternaire 3 zones



```
#Diagramme 3 zones  
base3.1 <- base3 + geom_point(data=dfhf,aes(P,O,N))  
  
#Générer le diagramme  
plot(base3.1)
```

Diagramme ternaire 4 zones avec points

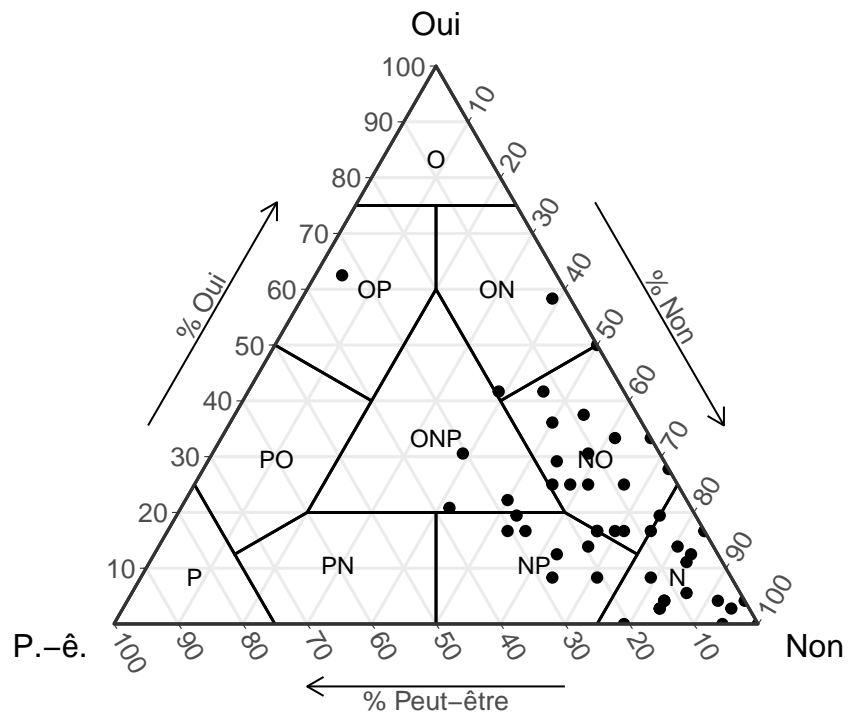
Diagramme ternaire 4 zones



```
#Diagramme 4 zones  
base4.1 <- base4 + geom_point(data=dfhf, aes(P,O,N))  
  
#Générer le diagramme  
plot(base4.1)
```

Diagramme ternaire 10 zones avec points

Diagramme ternaire 10 zones

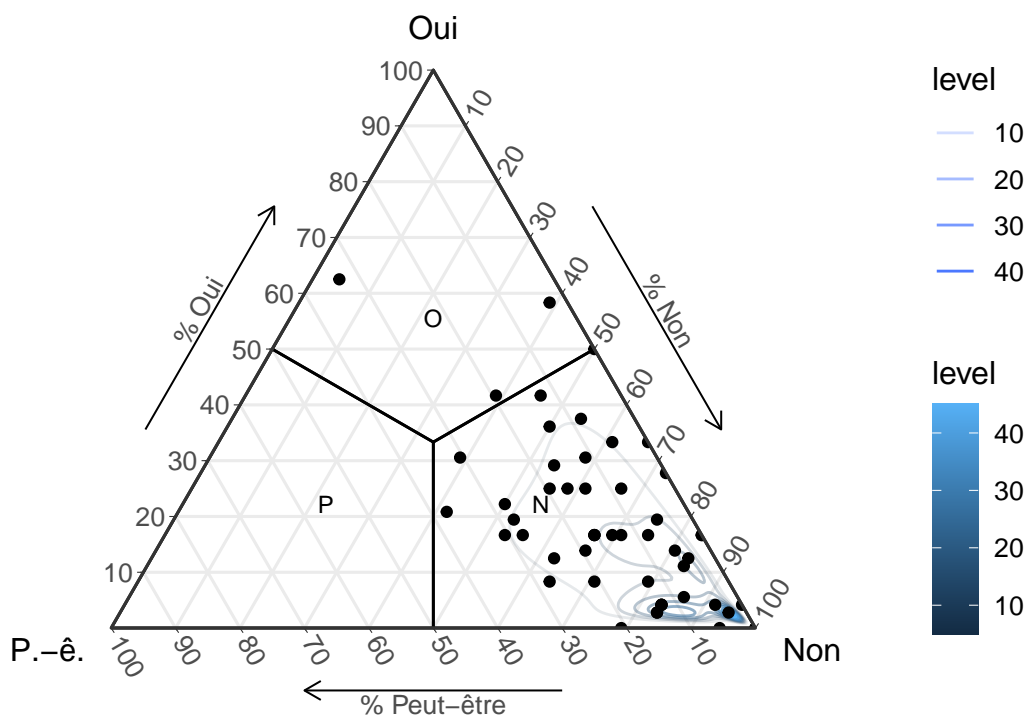


```
#Diagramme 10 zones  
base10.1 <- base10 + geom_point(data=dfhf, aes(P,O,N))  
  
#Générer le diagramme  
plot(base10.1)
```

Affichage des données sous forme de points avec estimation par noyau

Diagramme ternaire 3 zones avec points et estimation par noyau

Diagramme ternaire 3 zones

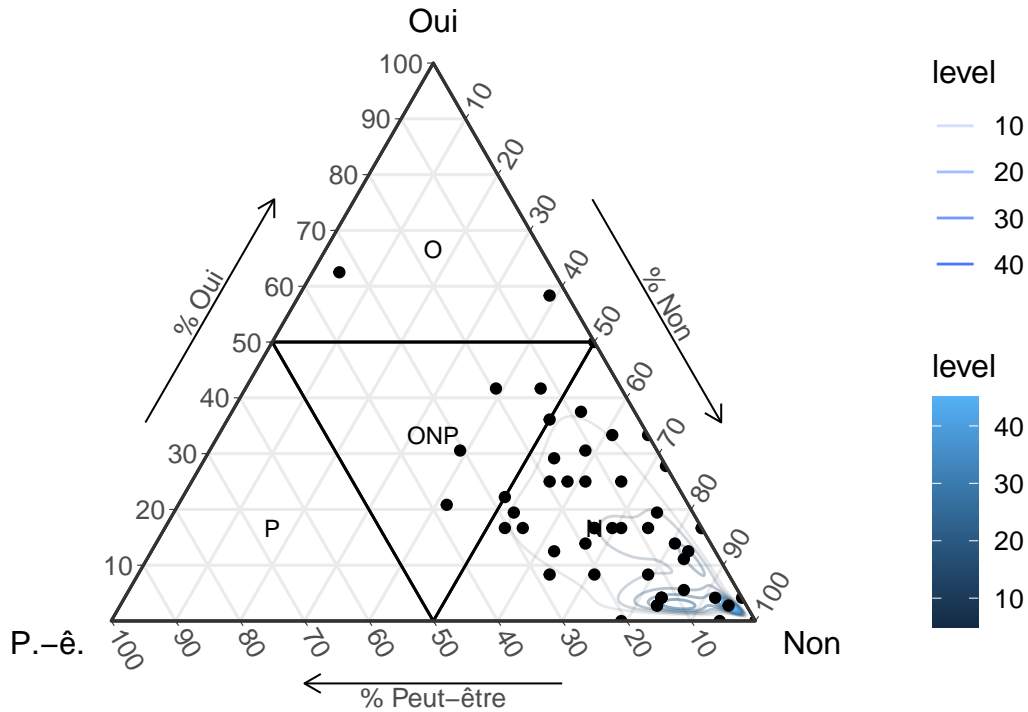


```
#Diagramme 3 zones + KDE
base3.2 <- base3 +
  geom_density_tern(
    data=dfhf,
    stat="DensityTern",
    n=200,aes(P,O,N,color=..level..,alpha=..level..),
    bdl.val=NA,
    inherit.aes=FALSE,
    base="ilr"
  ) +
  geom_point(data=dfhf,aes(P,O,N))

#Générer le diagramme
plot(base3.2)
```

Diagramme ternaire 4 zones avec points et estimation par noyau

Diagramme ternaire 4 zones

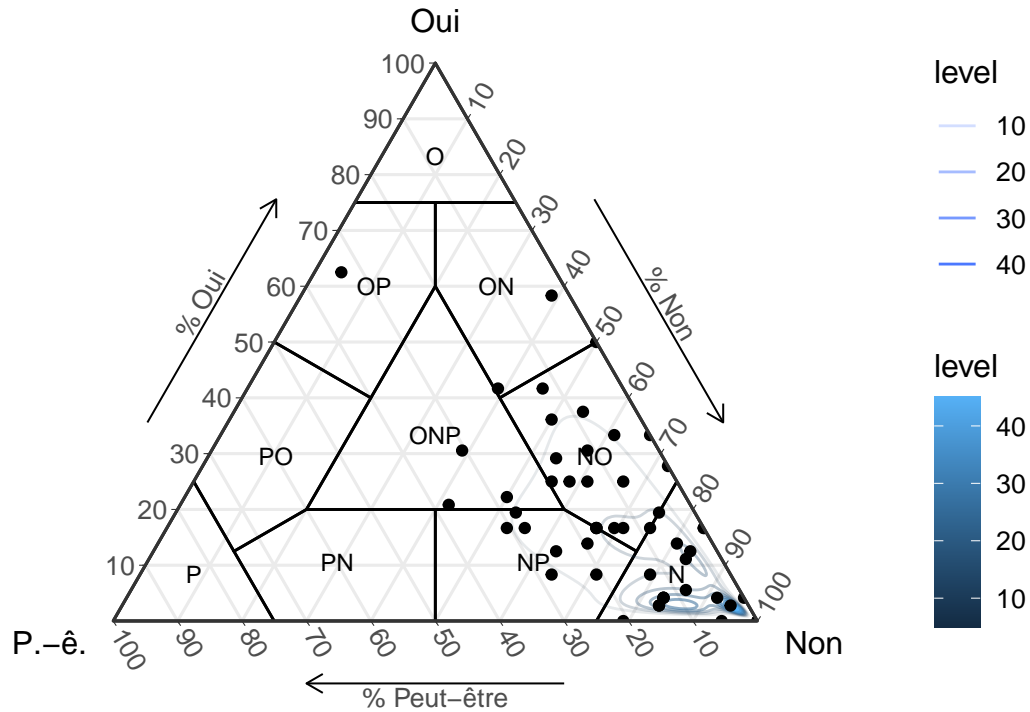


```
#Diagramme 4 zones + KDE
base4.2 <- base4 +
  geom_density_tern(
    data=dfhf,
    stat="DensityTern",
    n=200,aes(P,O,N,color=..level..,alpha=..level..),
    bdl.val=NA,
    inherit.aes=FALSE,
    base="ilr"
  ) +
  geom_point(data=dfhf,aes(P,O,N))

#Générer le diagramme
plot(base4.2)
```

Diagramme ternaire 10 zones avec points et estimation par noyau

Diagramme ternaire 10 zones



```
#Diagramme 10 zones + KDE
base10.2 <- base10 +
  geom_density_tern(
    data=dfhf,
    stat="DensityTern",
    n=200,aes(P,O,N,color=..level..,alpha=..level..),
    bdl.val=NA,
    inherit.aes=FALSE,
    base="ilr"
  ) +
  geom_point(data=dfhf,aes(P,O,N))

#Générer le diagramme
plot(base10.2)
```

Bibliographie

- Hamilton, Nicholas E. 2014. “Shepards sediment classificaton.” ggtern: ternary diagrams in R. July 26, 2014. <http://www.ggtern.com/2014/07/26/user-request-shepards-classification-sediments/>.
- . 2020a. *ggtern: An extension to 'ggplot2', for the creation of ternary diagrams* (version 3.3.0). <https://CRAN.R-project.org/package=ggtern>.
- . 2020b. “Package ‘ggtern.’” <https://cran.r-project.org/web/packages/ggtern/ggtern.pdf>.
- . n.d. “What is ggtern?” ggtern: ternary diagrams in R. Accessed June 7, 2019. <http://www.ggtern.com/>.
- Hamilton, Nicholas E., and Michael Ferry. 2018. “ggtern: ternary diagrams using ggplot2.” *Journal of Statistical Software* 87 (1): 1–17. <https://doi.org/10.18637/jss.v087.c03>.
- R Development Core Team. 2020. *R* (version 4.0.2). <https://www.r-project.org/>.
- Roux, Ugo. 2019a. “Communication virale dans la publicité au sein des espaces numériques : jeux de données.” Zenodo. <https://doi.org/10.5281/zenodo.3350487>.
- . 2019b. “Communication virale dans la publicité au sein des espaces numériques : jeux de données.” *Revue française des sciences de l'information et de la communication*, no. 18. <https://doi.org/10.4000/rfsic.7784>.
- . 2020. “Modélisation et catégorisation du potentiel viral d'un objet culturel en ligne et du profil des individus selon les intentions de partage.” *Communication* 37 (2). <https://doi.org/10.4000/communication.12842>.
- RStudio. 2020. *RStudio Desktop* (version 1.3.1093). <https://www.rstudio.com/>.