Willingness-To-Pay for Reshuffling Geographical Indications

Monia SAÏDI*
monia.saidi@inrae.fr

Jean-Sauveur AY*
jean-sauveur.ay@inrae.fr

Stéphan MARETTE†
stephan.marette@inrae.fr

Christophe MARTIN‡
christophe.martin@inrae.fr

Replication Material V2.1, April 11, 2020

Abstract

This file contents the Replication Material (RM) associated to the article named in the title and under revision in the *Journal of Wine Economics*. Data, code, figures, and tables are under the copyright license GNU GPL V3, which means that license notices must be preserved. Raw data are available from the Inrae dataverse server https://data.inrae.fr. The most recent version of this document and the detailed experimental protocol are available from the remote repository https://github.com/jsay/reshufGI.

Contents

*UMR CESAER : AgroSup Dijon, INRAE, Univ. Bourgogne Franche-Comté, F-21000 Dijon.
†UMR ECOPUB : AgroParisTech, INRAE, Univ. Paris Saclay, F-75000 Paris.
‡UMR CSGA : CNRS, INRAE, Univ. Bourgogne Franche-Comté, F-21000 Dijon.

1 Data preparation

1.1 Individual data

The data about elicited Willingness-To-Pay (WTP) are available on the Inrae dataverse: https://data.inrae.fr. The file `WTPraw.csv` can be downloaded by hand, or loaded directly in `R` with the `dataverse` package [Leeper, T. J. (2017). dataverse: R Client for Dataverse 4. R package version 0.2.0.] as below.

```
library(dataverse)
Sys.setenv("DATAVERSE_SERVER" = "data.inrae.fr")
WTPraw <- get_file("WTPraw.csv", " https://doi.org/10.15454/5ICGGD")
writeBin(WTPraw, "WTPraw.csv")
```

The dataset contains 125 rows (one row for each participant) and 37 columns (one for each question/answer). The first variable is the ID of the participant, the second variable is the result from the simulation at the beginning of the experiment in order to explain the BDM mechanism to the participants, the next 15 variables (between column 3 and column 17) correspond to the declared Willingness-To-Pay (WTP) that were asked for 3 different GI levels (`REG`, `VIL`, and `PCRU`) for each of the 5 different scenarios proposed to each participants. The last 20 columns corresponds to intermediary questions that could be used to control participants' heterogeneity. The detailed survey of the experiment is available here.

We re-code below the names of the WTP variables to melt them, and obtain a pooled data table of 1875 rows and 3 columns.

```
library(data.table)
dim(DT1 <- fread("Data/WTPraw.csv"))
names(DT1)[ 3: 17] <- paste0("X", substr(names(DT1)[ 3: 17], 2, 5))
dim(LDT1 <- melt(DT1, id.vars= "ID", measure.vars= names(DT1)[ 3: 17],
                 variable.name= "CHOIX", value.name= "WTP"))
LDT1$IDST <- paste(substr(LDT1$ID, 1, 3),
                   substr(LDT1$CHOIX, 1, 2), sep= "")
```

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

[1] 125  37

[1] 1875    3
```

These melted data contain the pooled WTP that are used in the regressions, as presented in the paper.

1.2 Scenario data

We also use the file `SCCraw.csv` about the characteristics of the scenario of GI changes that we propose to the participants. As before, this file can be downloaded by hand or with the `dataverse` package.

We perform 10 group sessions with 5 scenarios on each, so data have 50 rows. Every scenarios is replicated on average 4 times, as we have 14 different scenarios about GI reshuffling on the area of *Marsannay*. The variable `STRUCT` reports 28 unique values because the 14 scenarios are present with and without the bottle of *Fixin Premier Cru*, each appear twice. We merge below the scenario characteristics with previous WTP data elicited from the experiment.

```
SCCraw <- get_file("SCCraw.csv", " https://doi.org/10.15454/5ICGGD")
writeBin(SCCraw, "SCCraw.csv")
dim(SCC <- fread("Data/SCCraw.csv"))
length(unique(SCC$STRUCT))
SCC$IDST <- paste0(substr(SCC$GROUPE, 1, 3), "X", substr(SCC$GROUPE, 5, 5))
dim(DatPool <- merge(LDT1, SCC[, -1], by= "IDST"))
```

```
[1] 50  2

[1] 28

[1] 1875    5
```

1.3 AOC variables

We compute 3 series of variables that are used in the regressions. The first series is about the GI variables that are both coded as `factor` in the `AOC` variable and as `dummies` in the `AOCREG`, `AOCVIL`, and `AOCPCR` variables. Next, we compute the number of wine bottle in each GI for each scenario for the `STRUCT` variable from scenarios characteristics. Finally, we code the `FIXIN` dummy variable that equals to 1 for participant for which the *Fixin Premier Cru* was present. The code below reports the distribution of dummy variables.

```
DatPool$AOC <- factor(substr(DatPool$CHOIX, 3, 5),
                      levels= c("REG", "VIL", "PCR"))
DatPool$AOCREG <- ifelse(DatPool$AOC== "REG", 1, 0)
DatPool$AOCVIL <- ifelse(DatPool$AOC== "VIL", 1, 0)
DatPool$AOCPCR <- ifelse(DatPool$AOC== "PCR", 1, 0)
DatPool$NBREG <- as.numeric(substr(DatPool$STRUCT, 5, 5))
DatPool$NBVIL <- as.numeric(substr(DatPool$STRUCT, 3, 3))
DatPool$NBPCR <- as.numeric(substr(DatPool$STRUCT, 1, 1))
DatPool$FIXIN <- ifelse(rowSums(DatPool[, 10: 12])== 11, 1, 0)
sapply(DatPool[, c(7: 9, 13)], table, simplify= TRUE)
```

```
  AOCREG AOCVIL AOCPCR FIXIN
0   1250   1250   1250   900
1    625    625    625   975
```

1.4 Wine dummies

We compute the wine dummies in each scenario. We can verify the code by the reported distribution: each wine is proposed 625 times (except *Fixin Premier Cru* that is only for 60% of participants).

```r
DatPool$VIN0 <- ifelse(DatPool$FIXIN== 1 & DatPool$AOC== "PCR", 1, 0)
DatPool$VIN1 <- ifelse(DatPool$FIXIN== 1,
                ifelse(DatPool$NBPCR>= 2 & DatPool$AOC== "PCR", 1,
                ifelse(DatPool$NBPCR<= 1 & DatPool$AOC== "VIL", 1, 0)),
                ifelse(DatPool$NBPCR>= 1 & DatPool$AOC== "PCR", 1,
                ifelse(DatPool$NBPCR<= 0 & DatPool$AOC== "VIL", 1, 0)))
DatPool$VIN2 <- ifelse(DatPool$FIXIN== 1,
                ifelse(DatPool$NBPCR>= 3 & DatPool$AOC== "PCR", 1,
                ifelse(DatPool$NBPCR<= 2 & DatPool$AOC== "VIL", 1, 0)),
                ifelse(DatPool$NBPCR>= 2 & DatPool$AOC== "PCR", 1,
                ifelse(DatPool$NBPCR<= 1 & DatPool$AOC== "VIL", 1, 0)))
DatPool$VIN3 <- ifelse(DatPool$FIXIN== 1,
                ifelse(DatPool$NBPCR>= 4 & DatPool$AOC== "PCR", 1,
                ifelse(DatPool$NBPCR<= 3 & DatPool$AOC== "VIL", 1, 0)),
                ifelse(DatPool$NBPCR>= 3 & DatPool$AOC== "PCR", 1,
                ifelse(DatPool$NBPCR<= 2 & DatPool$AOC== "VIL", 1, 0)))
DatPool$VIN4 <- ifelse(DatPool$FIXIN== 1,
                ifelse(DatPool$NBPCR>= 5 & DatPool$AOC== "PCR", 1,
                ifelse(DatPool$NBPCR<= 4 & DatPool$AOC== "VIL", 1, 0)),
                ifelse(DatPool$NBPCR>= 4 & DatPool$AOC== "PCR", 1,
                ifelse(DatPool$NBPCR<= 3 & DatPool$AOC== "VIL", 1, 0)))
DatPool$VIN5 <- ifelse(DatPool$AOC== "VIL", 1, 0)
DatPool$VIN6 <- ifelse(DatPool$AOC== "VIL", 1, 0)
DatPool$VIN7 <- ifelse(DatPool$NBREG>= 4 & DatPool$AOC== "REG", 1,
                ifelse(DatPool$NBREG<  4 & DatPool$AOC== "VIL", 1, 0))
DatPool$VIN8 <- ifelse(DatPool$NBREG>= 3 & DatPool$AOC== "REG", 1,
                ifelse(DatPool$NBREG<  3 & DatPool$AOC== "VIL", 1, 0))
DatPool$VIN9 <- ifelse(DatPool$NBREG== 1,
                ifelse(DatPool$AOC== "VIL", 1, 0),
                ifelse(DatPool$AOC== "REG", 1, 0))
DatPool$VIN10<- ifelse(DatPool$AOC== "REG", 1, 0)
sapply(DatPool[, 14: 24], table)
```

```
  VIN0 VIN1 VIN2 VIN3 VIN4 VIN5 VIN6 VIN7 VIN8 VIN9 VIN10
0 1550 1250 1250 1250 1250 1250 1250 1250 1250 1250  1250
1  325  625  625  625  625  625  625  625  625  625   625
```

1.5 Average score

To compute the average score corresponding to each GI, we make the analysis for each GIs and then aggregate (see in the paper).

```
DatPool$REGscr <- ifelse(DatPool$NBREG== 1, 0,
                  ifelse(DatPool$NBREG== 2, .5,
                  ifelse(DatPool$NBREG== 3, 1, 1.5)))
DatPool$VILscr <- ifelse(DatPool$NBREG== 1,
                   ifelse(DatPool$NBVIL== 6, 3.5, 3),
                  ifelse(DatPool$NBREG== 2,
                   ifelse(DatPool$NBVIL== 4, 3.5,
                   ifelse(DatPool$NBVIL== 5, 4, 4.5)),
                  ifelse(DatPool$NBREG== 3,
                   ifelse(DatPool$NBVIL== 3, 4,
                   ifelse(DatPool$NBVIL== 4, 4.5,
                   ifelse(DatPool$NBVIL== 5, 5, 5.5))),
                   ifelse(DatPool$NBVIL== 3, 5,
                   ifelse(DatPool$NBVIL== 4, 5.5,
                   ifelse(DatPool$NBVIL== 5, 6,
                   ifelse(DatPool$NBVIL== 6, 6.5, 4.5)))))))
DatPool$PCRscr <- ifelse(DatPool$FIXIN== 1,
                  ifelse(DatPool$NBPCR== 1, 10,
                  ifelse(DatPool$NBPCR== 2, 9.5,
                  ifelse(DatPool$NBPCR== 3, 9,
                  ifelse(DatPool$NBPCR== 4, 8.5,
                  ifelse(DatPool$NBPCR== 5, 8, 8))))),
                  ifelse(DatPool$NBPCR== 1, 9,
                  ifelse(DatPool$NBPCR== 2, 8.5,
                  ifelse(DatPool$NBPCR== 3, 8,
                  ifelse(DatPool$NBPCR== 4, 7.5, 7.5)))))
DatPool$MEAN <- ifelse(DatPool$AOC== "PCR", DatPool$PCRscr,
                ifelse(DatPool$AOC== "VIL", DatPool$VILscr,
                       DatPool$REGscr))
sapply(DatPool[, 25: 28], summary)
```

```
        REGscr VILscr PCRscr   MEAN
Min.     0.000  3.000  7.500  0.000
1st Qu.  1.000  4.500  8.000  1.500
Median   1.500  5.000  8.500  5.000
Mean     1.102  5.102  8.568  4.924
3rd Qu.  1.500  6.000  9.000  8.000
Max.     1.500  6.500 10.000 10.000
```

1.6 Score variance

In two steps, as for the average score above.

```
DatPool$REGvar <- ifelse(DatPool$NBREG== 1, 0,
                  ifelse(DatPool$NBREG== 2, .5,
                  ifelse(DatPool$NBREG== 3, 1, 1.667)))
DatPool$VILvar <- ifelse(DatPool$NBVIL== 6, 3.5,
                  ifelse(DatPool$NBVIL== 5, 2.5,
                  ifelse(DatPool$NBVIL== 4, 1.667,
                  ifelse(DatPool$NBVIL== 3, 1, .5))))
DatPool$PCRvar <- ifelse(DatPool$NBPCR== 1 | DatPool$NBPCR== 0, 0,
                  ifelse(DatPool$NBPCR== 2, .005,
                  ifelse(DatPool$NBPCR== 3, .01,
                  ifelse(DatPool$NBPCR== 4, .01667, .025))))* 100
DatPool$VAR <- ifelse(DatPool$AOC== "PCR", DatPool$PCRvar,
               ifelse(DatPool$AOC== "VIL", DatPool$VILvar,
                      DatPool$REGvar))
sapply(DatPool[, 29: 32], summary)
```

```
        REGvar VILvar PCRvar   VAR
Min.     0.000  0.500 0.0000 0.000
1st Qu.  1.000  1.667 0.0000 0.500
Median   1.667  2.500 1.0000 1.667
Mean     1.190  2.430 0.8934 1.504
3rd Qu.  1.667  3.500 1.6670 1.667
Max.     1.667  3.500 2.5000 3.500
```

1.7 Summary Table

We construct here the summary Table 2 of the paper.

```
DatPool$WTPreg <- ifelse(DatPool$AOC== "REG", DatPool$WTP, NA)
DatPool$WTPvil <- ifelse(DatPool$AOC== "VIL", DatPool$WTP, NA)
DatPool$WTPpcr <- ifelse(DatPool$AOC== "PCR", DatPool$WTP, NA)

DatPool$SCRreg <- ifelse(DatPool$AOC== "REG", DatPool$MEAN, NA)
DatPool$SCRvil <- ifelse(DatPool$AOC== "VIL", DatPool$MEAN, NA)
DatPool$SCRpcr <- ifelse(DatPool$AOC== "PCR", DatPool$MEAN, NA)

DatPool$VARreg <- ifelse(DatPool$AOC== "REG", DatPool$VAR, NA)
DatPool$VARvil <- ifelse(DatPool$AOC== "VIL", DatPool$VAR, NA)
DatPool$VARpcr <- ifelse(DatPool$AOC== "PCR", DatPool$VAR, NA)

DatPool$MEANpcr[ is.na(DatPool$WTPpcr)] <- NA
DatPool$VARpcr[ is.na(DatPool$WTPpcr)] <- NA

library(stargazer)
## stargazer(DatPool[, c("WTP", "WTPreg", "WTPvil", "WTPpcr",
##                        "MEAN", "SCRreg", "SCRvil", "SCRpcr",
##                        "VAR", "VARreg", "VARvil", "VARpcr")],
##            type= "html", out= "Tables/TabSumStats.html")
stargazer(DatPool[, c("WTP", "WTPreg", "WTPvil", "WTPpcr",
                      "MEAN", "SCRreg", "SCRvil", "SCRpcr",
                      "VAR", "VARreg", "VARvil", "VARpcr")], type= "text")
```

```
==============================================================
Statistic   N    Mean  St. Dev.  Min  Pctl(25) Pctl(75)  Max
--------------------------------------------------------------
WTP       1,815 9.644   6.359   0.000  5.500    12.500  42.000
WTPreg     625  6.765   4.628   0.000  4.000    9.000   38.000
WTPvil     625  9.480   5.599   0.000  6.000    12.900  37.000
WTPpcr     565  13.010  7.149   0.000  8.200    17.000  42.000
MEAN      1,875 4.924   3.159     0     1.5       8       10
SCRreg     625  1.102   0.493   0.000  1.000    1.500   1.500
SCRvil     625  5.102   1.069   3.000  4.500    6.000   6.500
SCRpcr     525  8.568   0.791   7.500  8.000    9.000   10.000
VAR       1,875 1.504   1.039     0     0.5      1.7       4
VARreg     625  1.190   0.565   0.000  1.000    1.667   1.667
VARvil     625  2.430   0.966   0.500  1.667    3.500   3.500
VARpcr     525  0.893   0.812   0.000  0.000    1.667   2.500
--------------------------------------------------------------
```

2 Regression analysis

2.1 Table SM1 cited in the paper

```r
library(lfe) ; library(texreg)
m1 <- felm(WTP~ AOC | 0  | 0 | ID, data= DatPool)
m2 <- felm(WTP~ AOC | ID | 0 | ID, data= DatPool)
m3 <- felm(WTP~ VIN0+ VIN1+ VIN2+ VIN3+ VIN4+ VIN5+ VIN7+ VIN8+ VIN9+ VIN10
           | 0 | 0 | ID, data= DatPool)
m4 <- felm(WTP~ VIN0+ VIN1+ VIN2+ VIN3+ VIN4+ VIN5+ VIN7+ VIN8+ VIN9+ VIN10
           | ID | 0 | ID, data= DatPool)
m5 <- felm(WTP~ VIN0+ VIN1+ VIN2+ VIN3+ VIN4+ VIN7+ VIN8+ VIN9+ AOC
           | 0 | 0 | ID, data= DatPool)
m6 <- felm(WTP~ VIN0+ VIN1+ VIN2+ VIN3+ VIN4+ VIN7+ VIN8+ VIN9+ AOC
           | ID | 0 | ID, data= DatPool)
screenreg(list(m1, m2, m3, m4, m5, m6))
```

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
(Intercept)	6.77 ***		12.06 ***		6.63 ***	
	(0.41)		(0.91)		(0.43)	
AOCVIL	2.71 ***	2.71 ***			2.80 ***	2.80 ***
	(0.20)	(0.20)			(0.25)	(0.26)
AOCPCR	6.25 ***	6.22 ***			5.43 ***	5.41 ***
	(0.40)	(0.40)			(0.78)	(0.55)
VIN0			1.70	1.73 **	1.70	1.73 **
			(1.26)	(0.61)	(1.26)	(0.61)
VIN1			-0.14	-0.14	-0.14	-0.14
			(0.15)	(0.15)	(0.15)	(0.15)
VIN2			0.13 *	0.13	0.13 *	0.13
			(0.07)	(0.07)	(0.07)	(0.07)
VIN3			0.02	0.02	0.02	0.02
			(0.06)	(0.06)	(0.06)	(0.06)
VIN4			0.02	0.02	0.02	0.02
			(0.07)	(0.07)	(0.07)	(0.07)
VIN5			-2.63 ***	-2.61 ***		
			(0.67)	(0.38)		
VIN7			-0.04	-0.04	-0.04	-0.04
			(0.10)	(0.11)	(0.10)	(0.11)
VIN8			0.02	0.02	0.02	0.02
			(0.11)	(0.12)	(0.11)	(0.12)
VIN9			0.16	0.16	0.16	0.16
			(0.22)	(0.23)	(0.22)	(0.23)
VIN10			-5.43 ***	-5.41 ***		
			(0.78)	(0.55)		
Number obs.	1815	1815	1815	1815	1815	1815
R^2 (full model)	0.16	0.89	0.16	0.89	0.16	0.89
R^2 (proj model)	0.16	0.59	0.16	0.60	0.16	0.60
Adj. R^2 (full model)	0.16	0.88	0.16	0.89	0.16	0.89
Adj. R^2 (proj model)	0.16	0.56	0.16	0.57	0.16	0.57

*** p < 0.001, ** p < 0.01, * p < 0.05

2.2 Table SM2 cited in the paper

```r
m1a <- felm(WTP~ MEAN+ VAR+ AOCPCR:VIN0 | 0  | 0 | ID, data= DatPool)
m1b <- felm(WTP~ MEAN+ VAR+ AOCPCR:VIN0 | ID | 0 | ID, data= DatPool)
m2a <- felm(WTP~ AOC+ MEAN+ AOCPCR:VIN0 | 0  | 0 | ID, data= DatPool)
m2b <- felm(WTP~ AOC+ MEAN+ AOCPCR:VIN0 | ID | 0 | ID, data= DatPool)
m3a <- felm(WTP~ AOC+ VAR+ AOCPCR:VIN0  | 0  | 0 | ID, data= DatPool)
m3b <- felm(WTP~ AOC+ VAR+ AOCPCR:VIN0  | ID | 0 |ID, data= DatPool)
m4a <- felm(WTP~ AOC+ MEAN+ VAR+ AOCPCR:VIN0| 0 | 0 | ID, data= DatPool)
m4b <- felm(WTP~ AOC+ MEAN+ VAR+ AOCPCR:VIN0| ID | 0 |ID, data= DatPool)
## htmlreg(list(m1a, m1b, m2a, m2b, m4a, m4b), file= "Tables/Reg2A.xls",
##         inline.css= F, doctype= T, html.tag= T, head.tag= T, body.tag= T)
screenreg(list(m1a, m1b, m2a, m2b, m4a, m4b))
```

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
(Intercept)	6.07 ***		6.41 ***		6.38 ***	
	(0.42)		(0.41)		(0.41)	
MEAN	0.79 ***	0.79 ***	0.32 ***	0.36 ***	0.32 ***	0.36 ***
	(0.05)	(0.05)	(0.09)	(0.05)	(0.09)	(0.05)
VAR	-0.17 ***	-0.18 ***			0.03	0.02
	(0.05)	(0.04)			(0.06)	(0.04)
AOCVIL			1.44 ***	1.28 ***	1.42 ***	1.27 ***
			(0.39)	(0.26)	(0.37)	(0.26)
AOCPCR			2.98 **	2.69 ***	3.02 **	2.72 ***
			(0.98)	(0.55)	(1.03)	(0.56)
AOCPCR:VIN0			1.46	1.46 *	1.45	1.46 *
			(1.25)	(0.61)	(1.26)	(0.61)
Number obs.	1815	1815	1815	1815	1815	1815
R^2 (full model)	0.16	0.89	0.17	0.90	0.17	0.90
R^2 (proj model)	0.16	0.59	0.17	0.61	0.17	0.61
Adj. R^2 (full model)	0.16	0.88	0.16	0.89	0.16	0.89
Adj. R^2 (proj model)	0.16	0.56	0.16	0.58	0.16	0.58

*** p < 0.001, ** p < 0.01, * p < 0.05

2.3 Table SM3 cited in the paper

```r
m5a <- felm(WTP~ AOC+ AOCREG:MEAN+ AOCVIL:MEAN+ AOCPCR:MEAN+ AOCPCR:VIN0
           | 0 | 0 | ID, data= DatPool)
m5b <- felm(WTP~ AOC+ AOCREG:MEAN+ AOCVIL:MEAN+ AOCPCR:MEAN+ AOCPCR:VIN0
           | ID | 0 | ID, data= DatPool)
m6a <- felm(WTP~ AOC+ MEAN+ AOCPCR:VIN0
           + AOCREG:VAR+ AOCVIL:VAR+ AOCPCR:VAR
           | 0 | 0 | ID, data= DatPool)
m6b <- felm(WTP~ AOC+ MEAN+ AOCPCR:VIN0
           + AOCREG:VAR+ AOCVIL:VAR+ AOCPCR:VAR
           | ID | 0 | ID, data= DatPool)
maa <- felm(WTP~ AOC+ AOCREG:MEAN+ AOCVIL:MEAN+ AOCPCR:VIN0
           + AOCREG:VAR + AOCVIL:VAR+ AOCPCR:VAR
           | 0 | 0 | ID, data= DatPool)
mbb <- felm(WTP~ AOC+ AOCREG:MEAN+ AOCVIL:MEAN+ AOCPCR:VIN0
           + AOCREG:VAR + AOCVIL:VAR+ AOCPCR:VAR
           | ID | 0 | ID, data= DatPool)
## htmlreg(list(m5a, m5b, m6a, m6b, maa, mbb), file= "Tables/Reg3A.xls",
##         inline.css= F, doctype= T, html.tag= T, head.tag= T, body.tag= T)
screenreg(list(m5a, m5b, m6a, m6b, maa, mbb))
```

```
====================================================================================================
                       Model 1       Model 2       Model 3       Model 4       Model 5       Model 6
----------------------------------------------------------------------------------------------------
(Intercept)            6.42 ***                    6.45 ***                    6.19 ***
                      (0.47)                      (0.47)                      (0.45)
AOCVIL                 1.74 ***      1.74 ***      1.56 ***      1.65 ***      1.88 ***      1.66 ***
                      (0.35)        (0.37)        (0.33)        (0.36)        (0.43)        (0.43)
AOCPCR                 1.43          1.57          4.24 **       3.94 ***      6.17 ***      6.04 ***
                      (1.02)        (1.00)        (1.42)        (0.64)        (0.80)        (0.60)
AOCREG:MEAN            0.31          0.46 **                                  4.08          0.06
                      (0.24)        (0.16)                                    (3.89)        (2.36)
AOCVIL:MEAN            0.26 **       0.29 ***                                  0.17          0.23 ***
                      (0.09)        (0.06)                                    (0.15)        (0.05)
AOCPCR:MEAN            0.51 ***      0.51 ***                                  0.48 ***      0.47 ***
                      (0.10)        (0.11)                                    (0.11)        (0.10)
AOCPCR:VIN0            1.32          1.35 *        1.65          1.63 **       1.84          1.87 **
                      (1.27)        (0.61)        (1.25)        (0.61)        (1.26)        (0.61)
MEAN                                               0.19          0.24 ***
                                                  (0.15)        (0.05)
AOCREG:VAR                                         0.09          0.18         -3.29          0.34
                                                  (0.12)        (0.14)        (3.31)        (2.01)
AOCVIL:VAR                                         0.21          0.14 *        0.22          0.14 *
                                                  (0.20)        (0.06)        (0.20)        (0.06)
AOCPCR:VAR                                        -0.26         -0.21 *       -0.41 ***     -0.41 ***
                                                  (0.14)        (0.09)        (0.08)        (0.09)
----------------------------------------------------------------------------------------------------
Number  obs.           1815          1815          1815          1815          1815          1815
R^2 (full model)       0.17          0.90          0.17          0.90          0.17          0.90
R^2 (proj model)       0.17          0.61          0.17          0.61          0.17          0.61
Adj. R^2 (full model)  0.16          0.89          0.16          0.89          0.16          0.89
Adj. R^2 (proj model)  0.16          0.58          0.16          0.58          0.16          0.58
====================================================================================================
*** p < 0.001, ** p < 0.01, * p < 0.05
```

2.4 Table 3 in the paper

```r
## htmlreg(list(m1, m5, m1a, m4a, m5a, maa), omit= "VIN", file= "Tables/Fin3.xls",
##         inline.css= F, doctype= T, html.tag= T, head.tag= T, body.tag= T)
screenreg(list(m1, m5, m1a, m4a, m5a, maa), omit= "VIN")
```

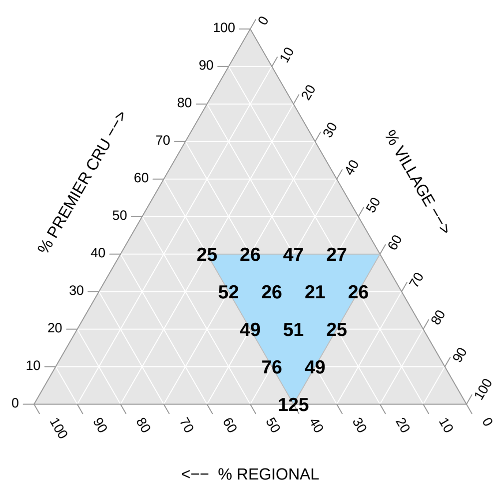
```
=================================================================================================
                        Model 1       Model 2       Model 3       Model 4       Model 5       Model 6
-------------------------------------------------------------------------------------------------
(Intercept)                6.77 ***      6.63 ***      6.17 ***      6.38 ***      6.42 ***      6.19 ***
                          (0.41)        (0.43)        (0.41)        (0.41)        (0.47)        (0.45)
AOCVIL                     2.71 ***      2.80 ***                    1.42 ***      1.74 ***      1.88 ***
                          (0.20)        (0.25)                      (0.37)        (0.35)        (0.43)
AOCPCR                     6.25 ***      5.43 ***                    3.02 **       1.43          6.17 ***
                          (0.40)        (0.78)                      (1.03)        (1.02)        (0.80)
MEAN                                                   0.79 ***      0.32 ***
                                                      (0.05)        (0.09)
VAR                                                   -0.17 ***      0.03
                                                      (0.05)        (0.06)
AOCREG:MEAN                                                                        0.31          4.08
                                                                                  (0.24)        (3.89)
MEAN:AOCVIL                                                                        0.26 **       0.17
                                                                                  (0.09)        (0.15)
MEAN:AOCPCR                                                                        0.51 ***
                                                                                  (0.10)
AOCREG:VAR                                                                                      -3.29
                                                                                                (3.31)
AOCVIL:VAR                                                                                       0.22
                                                                                                (0.20)
AOCPCR:VAR                                                                                      -0.41 ***
                                                                                                (0.08)
-------------------------------------------------------------------------------------------------
obs.                 1815          1815          1815          1815          1815          1815
R^2 (full model)           0.16          0.16          0.16          0.17          0.17          0.17
R^2 (proj model)           0.16          0.16          0.16          0.17          0.17          0.17
R^2 (full model)     0.16          0.16          0.16          0.16          0.16          0.16
R^2 (proj model)     0.16          0.16          0.16          0.16          0.16          0.16
=================================================================================================
*** p < 0.001, ** p < 0.01, * p < 0.05
```

3 Figures

3.1 Figure 1

Using the `ternary` package.

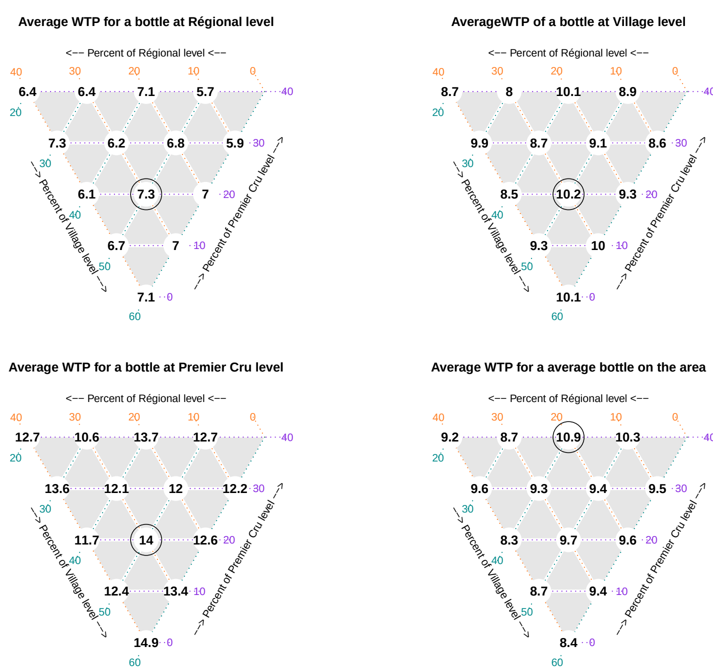
```
library(Ternary)
DatPool$SR <- ifelse(DatPool$FIXIN!= 1, DatPool$STRUCT,
                     paste0(as.numeric(substr(DatPool$STRUCT, 1, 1))- 1,
                            substr(DatPool$STRUCT, 2, 5)))
gg <- data.frame(SR= DatPool$SR,
                 model.matrix(~0+ DatPool$SR)/ 3)
hh <- aggregate(rep(1/ 3, nrow(gg)), by= list(gg$SR), sum)
hh <- data.frame(hh, as.numeric(substr(hh$Group.1, 1, 1)),
                 as.numeric(substr(hh$Group.1, 3, 3)),
                 as.numeric(substr(hh$Group.1, 5, 5)))
dpt <- list(as.numeric(hh[1, 3: 5]))
for (i in 2: nrow(hh)) dpt <- c(dpt, list(as.numeric(hh[i, 3: 5])))
par(mar= c(0, 0, 2, 0))
TernaryPlot(alab= '% PREMIER CRU -->', isometric= T,
            blab= '% VILLAGE -->', clab= '<--  % REGIONAL',
            grid.lty='solid', grid.col='white',
            axis.col=rgb(0.6, 0.6, 0.6), ticks.col=rgb(0.6, 0.6, 0.6),
            main= "", col= "grey90",
            grid.minor.lines= 0, padding= .075)
Interest <- matrix(c( 40, 20, 40,
                      40, 60, 00,
                      00, 60, 40), ncol= 3, byrow= TRUE)
TernaryPolygon(Interest, col='grey80', border='grey')
AddToTernary(text, dpt, hh$x, cex=1.2, font=2)
```



3.2 Figure 2 color

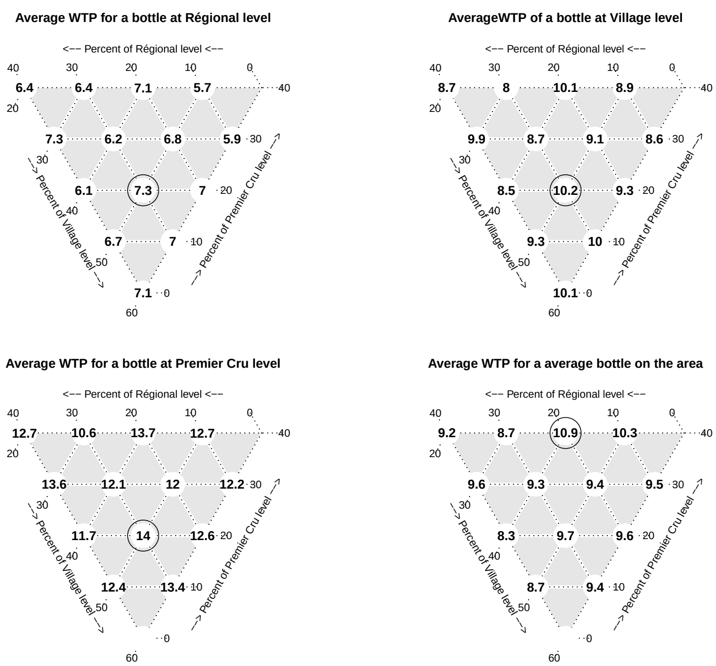
See the Appendix for the function `TernZoom`.

```
yop <- aggregate(DatPool$WTP,
                 by= list(DatPool$AOC, DatPool$SR), mean)
names(yop) <- c("VIN", "SR", "ValP")
yap1 <- merge(yop[yop$VIN== "PCR", c("SR", "ValP")],
              yop[yop$VIN== "VIL", c("SR", "ValP")], by= "SR")
yap2 <- merge(yap1, yop[yop$VIN== "REG", c("SR", "ValP")], by= "SR")
yap2$PCR <- as.numeric(substr(yap2$SR, 1, 1))
yap2$VIL <- as.numeric(substr(yap2$SR, 3, 3))
yap2$REG <- as.numeric(substr(yap2$SR, 5, 5))
yup <- yap2[order(yap2$REG, yap2$VIL, yap2$PCR), ]
yup$ValT <- (yup$PCR* yup$ValP.x+
             yup$VIL* yup$ValP.y+ yup$REG* yap2$ValP)/ 10
# png(filename= "Figures/TriangleF2.png",
#    units="in", width= 11, height= 9, pointsize= 12, res=300)
par(mfrow= c(2, 2), mar= c(0, 0, 3, 0))
TernZoom(yup$ValP, "Average WTP for a bottle at Régional level")
AddToTernary(points, c(50, 25, 25), pch=21, cex=6.5)
TernZoom(yup$ValP.y, "AverageWTP of a bottle at Village level")
AddToTernary(points, c(50, 25, 25), pch=21, cex=6.5)
TernZoom(yup$ValP.x, "Average WTP for a bottle at Premier Cru level")
AddToTernary(points, c(50, 25, 25), pch=21, cex=6.5)
TernZoom(yup$ValT, "Average WTP for a average bottle on the area")
AddToTernary(points, c(0, 50, 50), pch=21, cex=6.5)
# dev.off()
```



3.3 Figure 2 black and white

```
# png(filename= "Figures/TriangleF3.png",
#     units="in", width= 11, height= 9, pointsize= 12, res=300)
par(mfrow= c(2, 2), mar= c(0, 0, 3, 0))
TernZoomBW(yup$ValP, "Average WTP for a bottle at Régional level")
AddToTernary(points, c(50, 25, 25), pch=21, cex=6.5)
TernZoomBW(yup$ValP.y, "AverageWTP of a bottle at Village level")
AddToTernary(points, c(50, 25, 25), pch=21, cex=6.5)
TernZoomBW(yup$ValP.x, "Average WTP for a bottle at Premier Cru level")
AddToTernary(points, c(50, 25, 25), pch=21, cex=6.5)
TernZoomBW(yup$ValT, "Average WTP for a average bottle on the area")
AddToTernary(points, c(0, 50, 50), pch=21, cex=6.5)
# dev.off()
```



4 Appendix

4.1 Function for ternary plots

```
TernZoom <- function(vecteur, lbl= ""){
    dpt2 <- list(c(0  , 2.5, 7.5), c(2.5, 0  , 7.5),  c(0  , 5  , 5  ),
                 c(2.5, 2.5, 5  ), c(5  , 0  , 5  ),  c(0  , 7.5, 2.5),
                 c(2.5, 5  , 2.5), c(5  , 2.5, 2.5),  c(7.5, 0  , 2.5),
                 c(0  , 10 , 0  ), c(2.5, 7.5, 0  ),  c(5  , 5  , 0  ),
                 c(7.5, 2.5, 0  ), c(10 , 0 , 0  ))
    TernaryPlot(alab= ' --> Percent of Premier Cru level --> ',
                blab= ' --> Percent of Village level --> ', col.lab= "red",
                clab= ' <-- Percent of Régional level <-- ', grid.lwd= 4,
                grid.lty='solid', col=rgb(.9, .9, .9), grid.col='white',
                axis.col="white", ticks.col= "white", isometric= T,
                padding= 0.1, main= lbl, grid.minor.lines= 0,
                grid.line= 4,  axis.labels= F, point= 'down')
    TernaryLines(list(c(100,   0,   0), c(-10, 115, 0)),
                 lty= 3, lwd= 1.4, col= "chocolate1")
    TernaryLines(list(c( 75,   0,  25), c(-10, 85, 25)),
                 lty= 3, lwd= 1.4, col= "chocolate1")
    TernaryLines(list(c( 50,   0,  50), c(-10, 60, 50)),
                 lty= 3, lwd= 1.4, col= "chocolate1")
    TernaryLines(list(c( 25,   0,  75), c(-10, 35, 75)),
                 lty= 3, lwd= 1.4, col= "chocolate1")
    TernaryLines(list(c(  0,   0, 100), c(-10, 10,100)),
                 lty= 3, lwd= 1.4, col= "chocolate1")
    AddToTernary(text, c(-10, 114, 0), 40, col= "chocolate1")
    AddToTernary(text, c(-10, 85, 25), 30, col= "chocolate1")
    AddToTernary(text, c(-10, 60, 50), 20, col= "chocolate1")
    AddToTernary(text, c(-10, 35, 75), 10, col= "chocolate1")
    AddToTernary(text, c(-10, 10,100),  0, col= "chocolate1")
    TernaryLines(list(c(  0,  75,  25), c( 35, 75 , -10)),
                 lty= 3, lwd= 1.4, col= "darkcyan")
    TernaryLines(list(c( 0,  50,  50), c( 60, 50, -10)),
                 lty= 3, lwd= 1.4, col= "darkcyan")
    TernaryLines(list(c( 0,   25,  75), c(85, 25, -10)),
                 lty= 3, lwd= 1.4, col= "darkcyan")
    TernaryLines(list(c( 0,   0, 100), c(115, 0, -10)),
                 lty= 3, lwd= 1.4, col= "darkcyan")
    TernaryLines(list(c( 0, 100,  0),  c( 10, 100, -10)),
                 lty= 3, lwd= 1.4, col= "darkcyan")
    AddToTernary(text, c(10,100, -10), 20, col= "darkcyan")
    AddToTernary(text, c(35, 75, -10), 30, col= "darkcyan")
    AddToTernary(text, c(60, 50, -10), 40, col= "darkcyan")
    AddToTernary(text, c(85, 25, -10), 50, col= "darkcyan")
    AddToTernary(text, c(115, 0, -10), 60, col= "darkcyan")
    TernaryLines(list(c(  0, 100,  0), c( 0, -10, 115)),
                 lty= 3, lwd= 1.4, col= "blueviolet")
    TernaryLines(list(c( 25,  75, 0), c( 25, -10, 85)),
                 lty= 3, lwd= 1.4, col= "blueviolet")
    TernaryLines(list(c( 50,  50,  0), c(50, -10, 60)),
                 lty= 3, lwd= 1.4, col= "blueviolet")
    TernaryLines(list(c( 75,  25, 0), c(75, -10, 35)),
                 lty= 3, lwd= 1.4, col= "blueviolet")
    TernaryLines(list(c(100,   0, 0), c(100, -10, 9.99)),
```

```
                lty= 3, lwd= 1.4, col= "blueviolet")
    AddToTernary(text, c( 0,-10, 115), 40, col= "blueviolet")
    AddToTernary(text, c(25,-10,  85), 30, col= "blueviolet")
    AddToTernary(text, c(50,-10,  60), 20, col= "blueviolet")
    AddToTernary(text, c(75,-10,  35), 10, col= "blueviolet")
    AddToTernary(text, c(100,-10, 9.99), 0, col= "blueviolet")
    AddToTernary(points, dpt2, pch= 21, col= 'white', bg= "white", cex=5)
    AddToTernary(text, dpt2, round(vecteur, 1), cex=1.25, font=2)
}
```

4.2 Function for ternary plots black and white

```
TernZoomBW <- function(vecteur, lbl= ""){
    dpt2 <- list(c(0  , 2.5, 7.5), c(2.5, 0  , 7.5),  c(0  , 5  , 5  ),
                 c(2.5, 2.5, 5  ), c(5  , 0  , 5  ),  c(0  , 7.5, 2.5),
                 c(2.5, 5  , 2.5), c(5  , 2.5, 2.5),  c(7.5, 0  , 2.5),
                 c(0  , 10 , 0  ), c(2.5, 7.5, 0  ),  c(5  , 5  , 0  ),
                 c(7.5, 2.5, 0  ), c(10 , 0 , 0  ))
    TernaryPlot(alab= ' --> Percent of Premier Cru level --> ',
                blab= ' --> Percent of Village level --> ', col.lab= "red",
                clab= ' <-- Percent of Régional level <-- ', grid.lwd= 4,
                grid.lty='solid', col=rgb(.9, .9, .9), grid.col='white',
                axis.col="white", ticks.col= "white", isometric= T,
                padding= 0.1, main= lbl, grid.minor.lines= 0,
                grid.line= 4,  axis.labels= F, point= 'down')
    TernaryLines(list(c(100,   0,   0), c(-10, 115, 0)), lty= 3, lwd= 1.4)
    TernaryLines(list(c( 75,   0,  25), c(-10, 85, 25)), lty= 3, lwd= 1.4)
    TernaryLines(list(c( 50,   0,  50), c(-10, 60, 50)), lty= 3, lwd= 1.4)
    TernaryLines(list(c( 25,   0,  75), c(-10, 35, 75)), lty= 3, lwd= 1.4)
    TernaryLines(list(c(  0,   0, 100), c(-10, 10,100)), lty= 3, lwd= 1.4)
    AddToTernary(text, c(-10, 114, 0), 40)
    AddToTernary(text, c(-10, 85, 25), 30)
    AddToTernary(text, c(-10, 60, 50), 20)
    AddToTernary(text, c(-10, 35, 75), 10)
    AddToTernary(text, c(-10, 10,100),  0)
    TernaryLines(list(c(  0,  75,  25), c( 35, 75 , -10)),lty= 3, lwd= 1.4)
    TernaryLines(list(c( 0,  50,  50), c( 60, 50, -10)), lty= 3, lwd= 1.4)
    TernaryLines(list(c( 0,   25,  75), c(85, 25, -10)), lty= 3, lwd= 1.4)
    TernaryLines(list(c( 0,   0, 100), c(115, 0, -10)), lty= 3, lwd= 1.4)
    TernaryLines(list(c( 0, 100,  0),  c( 10, 100, -10)), lty= 3, lwd= 1.4)
    AddToTernary(text, c(10,100, -10), 20)
    AddToTernary(text, c(35, 75, -10), 30)
    AddToTernary(text, c(60, 50, -10), 40)
    AddToTernary(text, c(85, 25, -10), 50)
    AddToTernary(text, c(115, 0, -10), 60)
    TernaryLines(list(c(  0, 100,  0), c( 0, -10, 115)), lty= 3, lwd= 1.4)
    TernaryLines(list(c( 25,  75, 0), c( 25, -10, 85)), lty= 3, lwd= 1.4)
    TernaryLines(list(c( 50,  50,  0), c(50, -10, 60)), lty= 3, lwd= 1.4)
    TernaryLines(list(c( 75,  25, 0), c(75, -10, 35)), lty= 3, lwd= 1.4)
    TernaryLines(list(c(100,   0, 0), c(100, -10, 9.99)), lty= 3, lwd= 1.4)
    AddToTernary(text, c( 0,-10, 115), 40)
    AddToTernary(text, c(25,-10,  85), 30)
    AddToTernary(text, c(50,-10,  60), 20)
    AddToTernary(text, c(75,-10,  35), 10)
    AddToTernary(text, c(100,-10, 9.99), 0)
```

```
    AddToTernary(points, dpt2, pch= 21, col= 'white', bg= "white", cex=5)
    AddToTernary(text, dpt2, round(vecteur, 1), cex=1.25, font=2)
}
```