Análisis caso ABB

Jordi López Sintas

3 de septiembre de 2014

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```r
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```r
require(dplyr)
```

```
## Loading required package: dplyr
## 
## Attaching package: 'dplyr'
## 
## The following objects are masked from 'package:stats':
## 
##     filter, lag
## 
## The following objects are masked from 'package:base':
## 
##     intersect, setdiff, setequal, union
```

```r
#la opción file.chose() en la función read.table nos permite escoger un fichero de datos guardado en el
#leer el fichero de datos abb-R.txt, el cual contiene los datos de la elección de las empresas eléctric
abb<-read.table("abb-r.txt", header=T)
#la función head() nos permite visualizar las primer seis líneas de un objeto de datos.
#He traspuesto el resupado con la función t() con el objeto de facilitar la lectura. Así las líneas rep
t(head(abb))
```

```
##                 1     2    3              4        5     6   
## id              "1"   "1"  "1"            "1"      "2"   "2" 
## Alternatives    "ABB" "GE" "Westinghouse" "Edison" "ABB" "GE"
## choice          "0"   "1"  "0"            "0"      "0"   "0" 
## price           "6"   "6"  "6"            "5"      "3"   "3" 
## energy_loss     "6"   "6"  "5"            "5"      "4"   "4" 
## maintenance     "7"   "6"  "7"            "6"      "5"   "5" 
## warranty        "6"   "7"  "5"            "7"      "4"   "4" 
## spare_parts     "6"   "9"  "3"            "8"      "4"   "7" 
## ease_install    "5"   "9"  "4"            "2"      "5"   "3" 
## problem_solving "7"   "7"  "7"            "6"      "6"   "5" 
## quality         "5"   "5"  "6"            "5"      "4"   "5" 
## DA              "1"   "0"  "0"            "0"      "1"   "0" 
## DB              "0"   "1"  "0"            "0"      "0"   "1" 
## DC              "0"   "0"  "1"            "0"      "0"   "0" 
## DD              "0"   "0"   "0"            "1"      "0"   "0"
## volume          "761" "761" "761"          "761"    "627" "627"
## district        "1"   "1"   "1"            "1"      "1"   "1"
```

```r
#la funcion names() muestra los nombres de las variables

names(abb)
```

```
##  [1] "id"              "Alternatives"    "choice"
##  [4] "price"           "energy_loss"     "maintenance"
##  [7] "warranty"        "spare_parts"     "ease_install"
## [10] "problem_solving" "quality"         "DA"
## [13] "DB"              "DC"              "DD"
## [16] "volume"          "district"
```

```r
#La función str() nos proporciona una descripción de la base de datos
str(abb)
```

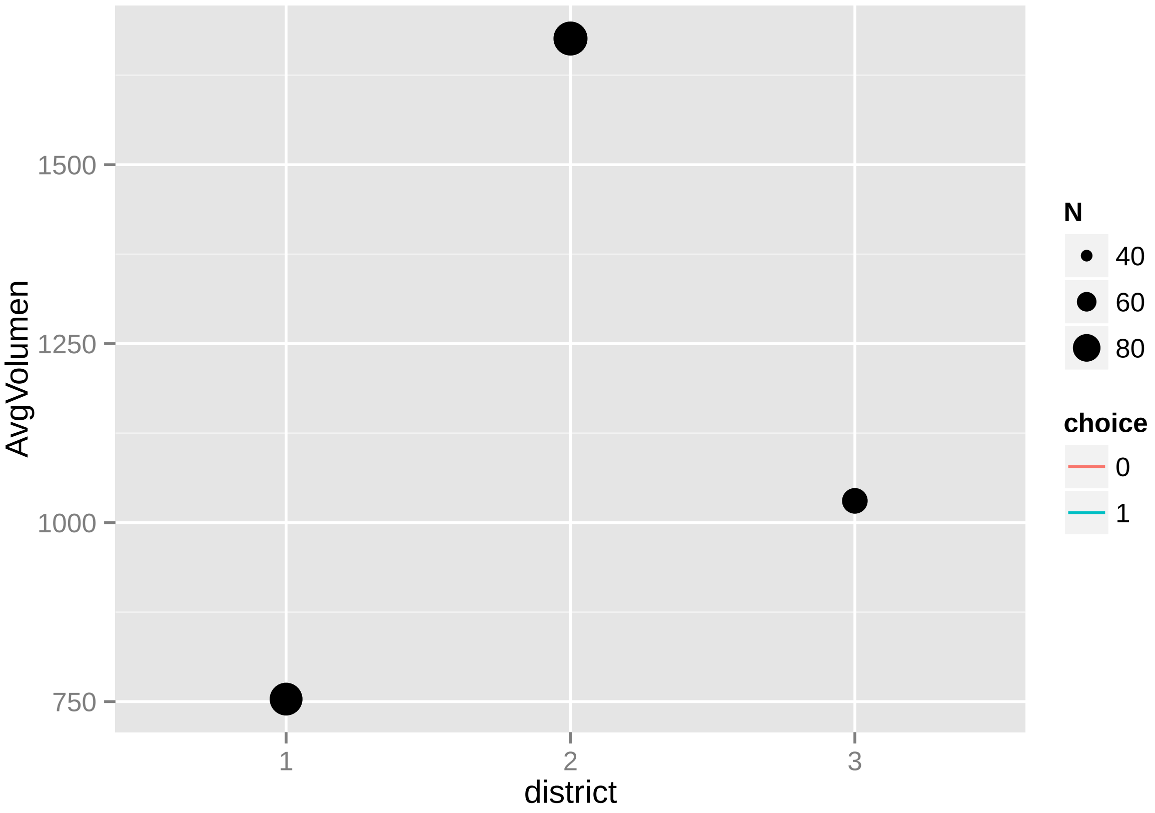
```
## 'data.frame':    352 obs. of  17 variables:
##  $ id             : int  1 1 1 1 2 2 2 2 3 3 ...
##  $ Alternatives   : Factor w/ 4 levels "ABB","Edison",..: 1 3 4 2 1 3 4 2 1 3 ...
##  $ choice         : num  0 1 0 0 0 0 0 1 1 0 ...
##  $ price          : num  6 6 6 5 3 3 4 4 6 5 ...
##  $ energy_loss    : num  6 6 5 5 4 4 5 5 6 6 ...
##  $ maintenance    : num  7 6 7 6 5 5 5 6 7 7 ...
##  $ warranty       : num  6 7 5 7 4 4 5 5 7 7 ...
##  $ spare_parts    : num  6 9 3 8 4 7 5 4 6 5 ...
##  $ ease_install   : num  5 9 4 2 5 3 7 5 7 6 ...
##  $ problem_solving: num  7 7 7 6 6 6 5 6 5 7 8 ...
##  $ quality        : num  5 5 6 5 4 5 4 6 6 6 ...
##  $ DA             : num  1 0 0 0 1 0 0 0 1 0 ...
##  $ DB             : num  0 1 0 0 0 1 0 0 0 1 ...
##  $ DC             : num  0 0 1 0 0 0 1 0 0 0 ...
##  $ DD             : int  0 0 0 1 0 0 0 1 0 0 ...
##  $ volume         : int  761 761 761 761 627 627 627 627 643 643 ...
##  $ district       : int  1 1 1 1 1 1 1 1 2 2 ...
```

```r
#cambiar la clase de las variables según sea apropiado.
#las variables choice y district deberían se factores.
abb$district <- as.factor(abb$district)
abb$choice <- as.factor(abb$choice)
#Ahora con la ayuda de la función select() del packete dplyr y del operador tubería (pipeline) %>% calc
A= select(abb, choice, volume, district, price) %>%
  group_by(district, choice) %>%
  summarize(AvgPrice = mean(price), AvgVolumen = mean(volume), N = length(price))
A
```

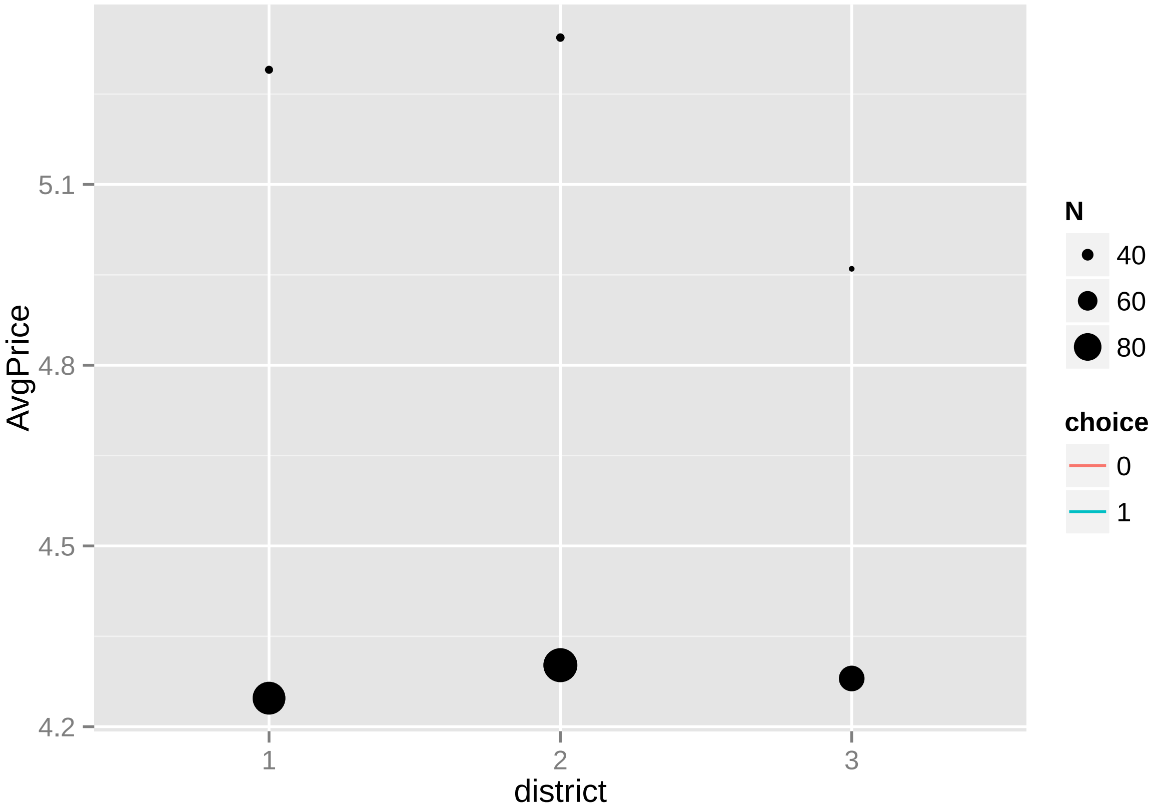
```
## Source: local data frame [6 x 5]
## Groups: district
##
##   district choice AvgPrice AvgVolumen  N
## 1        1      0    4.247      753.5 93
## 2        1      1    5.290      753.5 31
## 3        2      0    4.302     1676.2 96
## 4        2      1    5.344     1676.2 32
## 5        3      0    4.280     1030.4 75
## 6        3      1    4.960     1030.4 25
```

You can also embed plots, for example:

```
## geom_path: Each group consist of only one observation. Do you need to adjust the group aesthetic?
```



```
## geom_path: Each group consist of only one observation. Do you need to adjust the group aesthetic?
```



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
## Loading required package: splines
```

```
## Call:
## coxph(formula = Surv(rep(1, 352L), choice) ~ price + energy_loss +
##     maintenance + warranty + spare_parts + ease_install + problem_solving +
##     quality + DA + DB + DC + strata(id), data = abb, method = "exact")
##
##   n= 352, number of events= 88
##
##                   coef exp(coef) se(coef)     z Pr(>|z|)
## price            2.181     8.851    0.587  3.72  0.00020 ***
## energy_loss      2.656    14.234    0.674  3.94  8.1e-05 ***
## maintenance      0.594     1.811    0.437  1.36  0.17431
## warranty         1.141     3.129    0.331  3.45  0.00057 ***
## spare_parts     -0.133     0.876    0.218 -0.61  0.54216
## ease_install     0.520     1.682    0.173  3.01  0.00263 **
## problem_solving  2.032     7.631    0.550  3.70  0.00022 ***
## quality          2.639    14.005    0.688  3.84  0.00012 ***
## DA              -0.124     0.884    0.679 -0.18  0.85524
## DB              -0.671     0.511    0.719 -0.93  0.35081
## DC              -0.687     0.503    0.715 -0.96  0.33650
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                 exp(coef) exp(-coef) lower .95 upper .95
## price               8.851     0.1130     2.804     27.95
## energy_loss        14.234     0.0703     3.801     53.31
## maintenance         1.811     0.5523     0.769      4.26
## warranty            3.129     0.3196     1.636      5.99
## spare_parts         0.876     1.1418     0.572      1.34
## ease_install        1.682     0.5945     1.199      2.36
## problem_solving     7.631     0.1310     2.598     22.41
## quality            14.005     0.0714     3.638     53.91
## DA                  0.884     1.1318     0.234      3.34
## DB                  0.511     1.9566     0.125      2.09
## DC                  0.503     1.9882     0.124      2.04
##
## Rsquare= 0.411   (max possible= 0.5 )
## Likelihood ratio test= 186  on 11 df,   p=0
## Wald test            = 23.7  on 11 df,   p=0.0142
## Score (logrank) test = 103  on 11 df,   p=0
```

Ahora calculamos la utilidad de cada elección

```
u <- predict(abb.clogit)
head(u)
```

```
##       1       2       3       4       5       6
##  2.0459  3.7277  0.2034 -5.9770 -3.7209 -5.0991
```

Después obtenemos exp(u) y sumamos exp(u) para cada individuo

```
eu <- exp(u)
sumaeu <- by(eu, abb$id, sum)
head(sumaeu)
```

```
## abb$id
##       1       2       3       4       5       6
##   50.55  516.23  248.18  164.16 2069.30  153.61
```

Ahora calculamos la probabilidad de elección de cada marca. Para ello definimos una función que llamaremos prob()

```
prob<-function(suma, eutil, indiv){
#suma, eutil, inviv son los argumentos de la función
n<-0
#Crea un vector con tantos elementos como el producto entre
#lis individuos y las marcas
p<-1:indiv*4
#Para cada individuo
for (i in 1:indiv) {
#para cada marca
for (j in 1:4) {
#construye un índice
n<-n+1
#calcula la probabilidad de que el individuo i compre la #marca j
p[n]<-eutil[n]/suma[i]
}
}
#Devuelve el vector de probabilidades
return(p)
}
```

Y después la utilizamos con los datos calculados previamente

```
pchoice <- prob(sumaeu, eu, 88)
head(pchoice)
```

```
## [1] 1.530e-01 8.227e-01 2.425e-02 5.018e-05 4.690e-05 1.182e-05
```

```
abb$pchoice <- pchoice
t(head(abb))
```

```
##                 1           2           3              4
## id              "1"         "1"         "1"            "1"
## Alternatives    "ABB"       "GE"        "Westinghouse" "Edison"
## choice          "1"         "2"         "1"            "1"
## price           "6"         "6"         "6"            "5"
## energy_loss     "6"         "6"         "5"            "5"
## maintenance     "7"         "6"         "7"            "6"
## warranty        "6"         "7"         "5"            "7"
## spare_parts     "6"         "9"         "3"            "8"
## ease_install    "5"         "9"         "4"            "2"
## problem_solving "7"         "7"         "7"            "6"
## quality         "5"         "5"         "6"            "5"
## DA              "1"         "0"         "0"            "0"
## DB              "0"         "1"         "0"            "0"
## DC              "0"         "0"         "1"            "0"
## DD              "0"         "0"         "0"            "1"
## volume          "761"       "761"       "761"          "761"
## district        "1"         "1"         "1"            "1"
## pchoice         "1.530e-01" "8.227e-01" "2.425e-02"    "5.018e-05"
##                 5           6
## id              "2"         "2"
## Alternatives    "ABB"       "GE"
## choice          "1"         "1"
## price           "3"         "3"
## energy_loss     "4"         "4"
## maintenance     "5"         "5"
## warranty        "4"         "4"
## spare_parts     "4"         "7"
## ease_install    "5"         "3"
## problem_solving "6"         "5"
## quality         "4"         "5"
## DA              "1"         "0"
## DB              "0"         "1"
## DC              "0"         "0"
## DD              "0"         "0"
## volume          "627"       "627"
## district        "1"         "1"
## pchoice         "4.690e-05" "1.182e-05"
```

Ahora creamos una función para clasificar a los clientes en función de su probabilidad de compra

```
msegment<-function(p, indiv){
# p es el vector de probabilidades
# in es el número de individuos
s<-1:indiv*4
j<-0
for (i in 1:indiv) {
#para cada individuo
j=j+4
#Leales
if (p[j-3]>0.8) {s[j-3]<-"L"; s[j-2]<-"L"; s[j-1]<-"L"; s[j]<-"L"}
#Competitivos
if (p[j-3]<=0.8 & p[j-3]>0.5) {s[j-3]<-"C"; s[j-2]<-"C"; s[j-1]<-"C"; s[j]<-"C"}
#Apropiables
if (p[j-3]<=0.5 & p[j-3]>0.15) {s[j-3]<-"A"; s[j-2]<-"A"; s[j-1]<-"A"; s[j]<-"A"}
#Perdidos
if (p[j-3]<=0.15) {s[j-3]<-"P"; s[j-2]<-"P"; s[j-1]<-"P"; s[j]<-"P"}
}
#Devuelve el resultado de la función
return(s)
}
```

Ahora utilizamos la nueva función para clasificar la base de datos

```
seg <- msegment(pchoice, 88)
abb$seg <- seg

abb.select.ord <- select(abb, volume, pchoice, seg) %>%
  arrange(-volume)
head(abb.select.ord)
```

```
##   volume   pchoice seg
## 1  14798 4.989e-04   P
## 2  14798 6.260e-08   P
## 3  14798 8.012e-07   P
## 4  14798 9.995e-01   P
## 5  12514 7.867e-03   P
## 6  12514 3.195e-04   P
```

primero seleccionamos las variables que queremos ordenar, después