

XI JORNADAS DE USUARIOS



Madrid, 14-16 de noviembre de 2019
Auditorio Repsol, 14 de noviembre
UNED Campus Moncloa, 15-16 de noviembre

Herramientas de R para la investigación de mercados

5. Conjoint Analysis



Conjoint Analysis (CA)

1. Qué es y para qué sirve
2. Métodos
3. Fases
4. Utilidad
5. Confección de SETS con R
6. Choice-Based Conjoint (CBC) en R con {mlogit}



1. Qué es y para qué sirve

- Técnica que permite estimar en qué medida los consumidores valoran las características (**ATRIBUTOS**) que componen un producto o servicio y cada una de las alternativas que puede tener cada atributo (**NIVELES**)
- Se solicita a un conjunto de respondientes que indique sus preferencias respecto a los productos, cada uno de ellos con sus definido por unas características determinadas (**PERFILES/Alternatives**)
- De este modo se calcula la importancia para los respondientes de cada **ATRIBUTO**, y dentro de ellos los **NIVELES**, a la hora de decantarse por cada **PERFIL/Alternative** de producto
- También permite estimar **nuevos escenarios, combinaciones de NIVELES** de los distintos atributos y **cuotas de mercado** de cada **PERFIL/Alternative** de producto



2.Métodos

- **Conjoint Value Analysis (CVA)**: método tradicional de perfil completo (full profile), actualmente casi en desuso. Los respondientes clasifican o puntúan todos los **PERFILES**
- **Adaptative Conjoint Analysis** (Análisis Conjunto Adaptativo - ACA): los **PERFILES** no recogen todos los **ATRIBUTOS** (diseño factorial fraccionado), sino una parte de ellos, seleccionados a partir de las preferencias expresadas por los respondientes
- **Choice-based Conjoint** (CBCA o CBC): se muestra un grupo de posibles productos o **PERFILES** (diseño factorial fraccionado) y se pide al entrevistado que elija cuál prefiere. Se utilizan modelos no lineales (modelos de elección discreta). Es el método más utilizado en la actualidad
- **Maxdiff**: el entrevistado elige el mejor y el peor. No es propiamente CA, pero es bueno para medir preferencias (marcas, envases, etc.)



2.Métodos

- Métodos más utilizados en 2019, según el ***Sawtooth Software User Survey***:

Porcentaje de usuarios que han utilizado cada método	
CBC (Choice-Based Conjoint)	72%
ACBC (Adaptive Choice)	37%
ACA (Adaptive Conjoint Analysis)	7%
CVA (traditional conjoint analysis)	5%
MBC (Menu-Based Choice)	10%

3.Fases

- Diseño experimental:
 - elección de **ATRIBUTOS** y **NIVELES** de Atributos del producto o servicio
 - confección de **PERFILES** a partir de la combinación de **NIVELES**
 - confección de **SETS** de **PERFILES**
- Toma de datos:
 - recogida de las **preferencias** de los entrevistados
 - re-**elaboración** de los datos para el análisis
- Análisis:
 - agregado o individualizado de las **UTILIDADES** de **ATRIBUTOS** y **NIVELES**
 - **segmentación** del mercado
 - proyección de **Perfiles** testados o nuevos



3.Métodos de diseño experimental

- **Diseño balanceado:** los diferentes Niveles de cada Atributo se muestran el mismo número de veces
- **Diseño ortogonal:** la cantidad de veces que un Nivel de un Atributo es comparado con los todos los Niveles del resto de Atributos es igual o proporcional
- **Equilibrio posicional:** evitar el “sesgo de orden”, tanto entre los Perfiles de un Set, como entre los Atributos de un Perfil



4.Utilidad (1)

- **Utilidad:** valor cuantificable que un consumidor atribuye a un bien
- No podemos conocer la utilidad objetiva que percibe una persona, sino sus preferencias **en comparación** con otras alternativas
- **Probabilidades de elección:** probabilidad de que un individuo tome una decisión concreta, teniendo en cuenta la utilidad medida para cada opción y la posibilidad de error debido a factores no observados

4.Utilidad (2)

- ¿Cómo determinamos la relación entre utilidad y probabilidad de elección?
- **Modelo de comportamiento:** modelo estadístico que determina la relación entre las utilidades y las probabilidades de elección. El más usado en **CA** es el **Logit Multinomial (MNL)**. Si los individuos eligen diferente, se debe a atributos no observados, que pueden ser diferentes para diferentes personas, y que generan error de medición. Pero si observásemos todos los atributos de los productos, ese error sería nulo y podríamos predecir perfectamente las decisiones de las personas



4.Utilidad (3)

- En la práctica no es posible observar todos los factores, por lo tanto siempre tendremos un cierto error. **MNL** modela este error mediante una distribución de probabilidad conocida como **distribución de valor extremo** (*extreme value*). Es una distribución parecida a la normal: tiene mayor densidad en su zona central (el valor 0 es el más probable) y un rápido decaimiento a medida que nos alejamos del valor central. A diferencia de la distribución normal, la **distribución de valor extremo** es integrable y nos permite relacionar utilidades y probabilidades de elección
- El **CA** parte de las opciones que prefieren los individuos y a partir de ahí calcula las utilidades de las opciones



5.Confección de SETS con R

- Descripción del ejemplo
- Creación de SETS de PERFILES
- Creación de un diseño factorial completo
- Creación de un diseño factorial fraccional
- Creación de M copias
- Creación de SETS
- Confección de los sets
- Ejemplo de set y pregunta

5.Descripción del ejemplo

Se analizan las preferencias de los consumidores respecto a algunos atributos de tipos de automóvil

Cada tipo de automóvil tiene tres atributos:

- AUTO**: el automóvil tiene cambio automático.
Niveles: 1 = NO; 2 = SI
- ELE**: el automóvil cuenta con motor eléctrico.
Niveles: 1 = NO; 2 = SI
- PRE**: precio del automóvil en miles de euros.
Niveles: 1 = 20; 2 = 30; 3 = 40; 4 = 50

El tipo de automóvil se supone igual para los demás atributos no reseñados



5.Creación de SETS de PERFILES

La función `gen.factorial()` de `{AlgDesign}` crea un diseño factorial completo

La función `optFederov()` de `{AlgDesign}` genera un diseño factorial fraccional a partir del diseño factorial completo

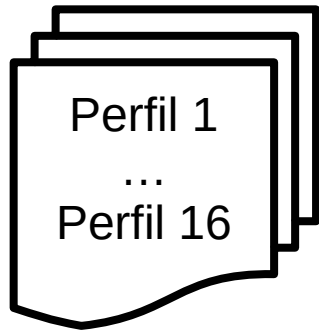
Creamos M copias del diseño factorial fraccional

Para crear S SETS de M alternativas, seleccionamos aleatoriamente una de las alternativas (rows) a partir de cada SET del diseño factorial fraccional sin remplazamiento. Repetimos este paso hasta que todas las alternativas de cada una de las M copias del diseño factorial fraccional están asignadas a los S SETS de elección

Se vuelcan los códigos de los perfiles a los NIVELES correspondientes para incluirlos en los SETS a presentar a los entrevistados



5.Creación de un diseño factorial completo

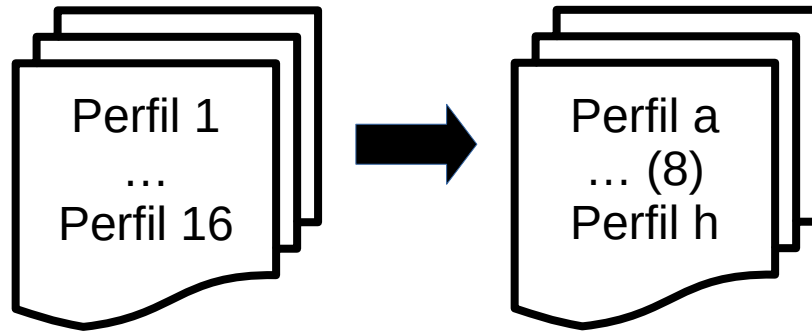


```
library (AlgDesign)
```

```
# Genera diseño factorial completo de 2x2x4=16  
perfiles:
```

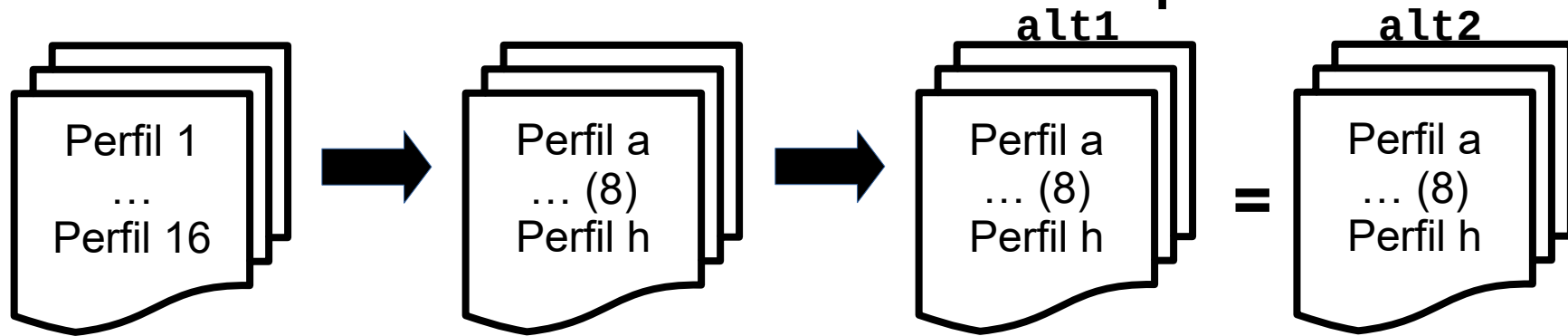
```
ffd <- gen.factorial (  
  c (2, 2, 4), # Núm. niveles de cada atributo  
  varNames = c ("AUT", "ELE", "PRE"), # Nombres  
  de los atributos  
  factors = "all") # Todos los atributos "factor"
```

5. Creación de un diseño factorial fraccional



```
set.seed (54321) # Generación aleatoria
des <- optFederov ( # Genera un diseño
  factorial fraccional a partir de uno completo
  ~., # Todas las variables de datos se usan
  linealmente y se usan sus nombres
  ffd, # El diseño factorial completo de
  partida
  8) # Número de perfiles a generar
```

5. Creación de M^1 copias



(1) M: número de PERFILES a incluir en cada set

```
alt1 <- des$design # Asignamos el diseño  
factorial fraccional a un primer objeto  
"alt1"
```

```
alt2 <- alt1 # Creamos una nueva copia igual
```


5.Creación de SETS

Seleccionamos aleatoriamente filas de los diseños factoriales fraccionales sin reemplazamiento añadiendo a cada diseño una nueva variable aleatoria:

```
alt1 <- transform (alt1, r1 = runif (8)) # Incorporamos a  
"alt1" una nueva variable aleatoria con ocho valores  
(tantas como filas tiene "alt1")
```

```
alt2 <- transform (alt2, r2 = runif (8)) # Idem
```

Ordenamos ambos diseños según la nueva variable aleatoria:

```
alt1_sort <- alt1 [order (alt1$r1), ]  
alt2_sort <- alt2 [order (alt2$r2), ]
```

Creamos las parejas de alternativas que presentaremos a los encuestados, ligando por su orden "alt1_sort" y "alt2_sort":

```
cbind (alt1_sort, alt2_sort) # Vemos cómo se emparejan
```

Si en algún caso aparece la misma alternativa dos veces en una pareja, se repite todo el proceso



5.Confección de los sets

```
> alt1_sort
```

	AUT	ELE	PRE	r1
1	1	1	1	0.05374091
10	2	1	3	0.111170085
8	2	2	2	0.12866907
5	1	1	2	0.15476521
11	1	2	3	0.23430698
4	2	2	1	0.75569340
15	1	2	4	0.78156383
14	2	1	4	0.91222889

← P.1 →

← P.2 →

← P.3 →

← P.4 →

← P.5 →

← P.6 →

← P.7 →

← P.8 →

```
> alt2_sort
```

	AUT	ELE	PRE	r2
14	2	1	4	0.02218425
4	2	2	1	0.17772093
15	1	2	4	0.49822342
1	1	1	1	0.55358492
8	2	2	2	0.68168019
10	2	1	3	0.72350066
11	1	2	3	0.81742403
5	1	1	2	0.97593965



P.	AUT	ELE	PRE
1	NO	NO	20
2	SI	NO	40
3	SI	SI	30
4	NO	NO	30
5	NO	SI	40
6	SI	SI	20
7	NO	SI	50
8	SI	NO	50

← P.1 →

← P.2 →

← P.3 →

← P.4 →

← P.5 →

← P.6 →

← P.7 →

← P.8 →



P.	AUT	ELE	PRE
1	SI	NO	50
2	SI	SI	20
3	NO	SI	50
4	NO	NO	20
5	SI	SI	30
6	SI	NO	40
7	NO	SI	40
8	NO	NO	30



5. Ejemplo de set y pregunta

P1. Por favor, rodee con un círculo el automóvil de la lista inferior que es más probable que usted adquiera:

- 1. Automóvil A*
- 2. Automóvil B*
- 3. Ninguno de los dos*

	Auto. A	Auto. B
Cambio automático	NO	SI
Motor eléctrico	NO	NO
Precio (miles euros)	20	50

Choice-Based Conjoint (CBC) en R con {mlogit} (1)

- Captura de los datos:

```
cbc.df <- read.csv  
("http://goo.gl/5xQ0bB", colClasses =  
c(seat = "factor", price =  
"factor")) # Utilizamos un ejemplo ya  
existente, convirtiendo dos variables en factor  
  
library(mlogit) # Paquete que estima el  
modelo de elección más habitual, el modelo  
multinomial logit o conditional logit  
  
str(cbc.df) # Vemos lo que tenemos
```



Choice-Based Conjoint (CBC) en R con {mlogit} (2)

- Transformamos los datos a un formato especial con la función `mlogit.data()`:

```
cbc.mlogit <- mlogit.data ( # Función utilizada
data = cbc.df, # Datos de partida
choice = "choice", # Columna con los datos de respuesta
shape = "long", # Datos presentados en formato "long" en
vez de "wide"
varying = 5:8, # Columnas con los atributos
alt.levels = paste ("pos", 1:3), # Nombres de las
alternativas
id.var = "resp.id") # Columna de los IDs de los
entrevistados
```

Obtenemos un dataset tipo `data.frame` y tipo `"mlogit.data"`



Choice-Based Conjoint (CBC) en R con {mlogit} (3)

- Estimamos el modelo con `mlogit()`:

```
m1 <- mlogit (choice ~ # Varial  
0 +  
seat + cargo + eng + price,  
data = cbc.mlogit)
```

La inclusión de "0 +" indica que no queremos que aparezca los (dos) "intercept", que expresan la posición que ocupan en la presentación del "choice"

```
summary (m1) # Inspeccionamos los resultados
```



Choice-Based Conjoint (CBC) en R con {mlogit} (4)

Call:

```
mlogit(formula = choice ~ 0 + seat + cargo + eng + price, data = cbc.mlogit, method =  
"nr", print.level = 0)
```

Frequencies of alternatives:

```
  pos 1    pos 2    pos 3  
0.32700 0.33467 0.33833
```

nr method

5 iterations, 0h:0m:0s

$g'(-H)^{-1}g = 7.84E-05$

successive function values within tolerance limits

Coefficients :

	Estimate	Std. Error	z-value	Pr(> z)	
seat7	-0.535280	0.062360	-8.5837	< 2.2e-16	***
seat8	-0.305840	0.061129	-5.0032	5.638e-07	***
cargo3ft	0.477449	0.050888	9.3824	< 2.2e-16	***
enggas	1.530762	0.067456	22.6926	< 2.2e-16	***
enghyb	0.719479	0.065529	10.9796	< 2.2e-16	***
price35	-0.913656	0.060601	-15.0765	< 2.2e-16	***
price40	-1.725851	0.069631	-24.7856	< 2.2e-16	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Log-Likelihood: -2581.6

