

```

library(support.CEs)
library(survival)
library(gridExtra)
library(grid)
library(ggplot2)
library(lattice)
library(readr)
library(ggplot2)

# diseño
des1 <- rotation.design(
  attribute.names = list(
    cantidad = c("250", "600", "litro"),
    agua = c("pura", "gas", "funcional"),
    botella = c("convencional", "recta", "lágrima")),
  nalternatives = 3,
  nblocks = 1,
  row.renames = TRUE,
  randomize = TRUE,
  seed = 987)

# cuestionario
questionnaire(choice.experiment.design = des1)

# nombrar el diseño
choice.experiment.design = des1

# matriz de diseno
desmat1 <- make.design.matrix(
  choice.experiment.design = des1,
  optout = TRUE, #la opción de exclusión SI esta incluida
  categorical.attributes = c("cantidad", "agua", "botella"),
  continuous.attributes = NULL,
  unlabeled = TRUE,
  common = NULL,
  binary = FALSE)

# subir el archivo de respuestas COMPLETO
RESPUESTAS <- read_delim("~/ANA/Clases/tesis/RESPUESTAS.csv",
  ";", escape_double = FALSE, trim_ws = TRUE)

## modelo completo
# haciendo el conjunto de datos adecuado para el modelo
dataset1 <- make.dataset(
  respondent.dataset = RESPUESTAS,
  choice.indicators =
    c("q1", "q2", "q3", "q4", "q5", "q6", "q7", "q8", "q9"),
  design.matrix = desmat1)

# MODELO LOGIT general
clogout1 <- clogit(RES ~ ASC + X600 + litro + gas + funcional + recta + lágrima
  + strata(STR), data = dataset1)

summary(clogout1)
gofm(clogout1)

# BASE DE DATOS DE LOS CLIENTES
RESPUESTAS_CLIENTES = RESPUESTAS[1:14,]

# haciendo el conjunto de datos adecuado para el modelo
dataset1_CLIENTES <- make.dataset(
  respondent.dataset = RESPUESTAS_CLIENTES,
  choice.indicators =
    c("q1", "q2", "q3", "q4", "q5", "q6", "q7", "q8", "q9"),
  design.matrix = desmat1)

# MODELO LOGIT PARA CLIENTES
clogout1_CLIENTES <- clogit(RES ~ ASC + X600 + litro + gas + funcional + recta + lágrima
  + strata(STR), data = dataset1_CLIENTES)

summary(clogout1_CLIENTES)
gofm(clogout1_CLIENTES)

# BASE DE DATOS PARA CONSUMIDOR
RESPUESTAS_CONS = RESPUESTAS[15:90,]

# haciendo el conjunto de datos adecuado para el modelo
dataset1_CONS <- make.dataset(
  respondent.dataset = RESPUESTAS_CONS,
  choice.indicators =
    c("q1", "q2", "q3", "q4", "q5", "q6", "q7", "q8", "q9"),
  design.matrix = desmat1)

```

```

# MODELO LOGIT PARA CONSUMIDOR FINAL
clogout1_CONS <- clogit(RES ~ ASC + X600 + litro + gas + funcional + recta + lágrima
+ strata(STR), data = dataset1_CONS)

summary(clogout1_CONS)
gofm(clogout1_CONS)

## ANALISIS DESCRIPTIVOS

descriptivos <- read.csv("~/ANA/Clases/tesis/ARCHIVOS TESIS/descriptivos.csv", sep=";")

# tipo de cliente

cliente <- ggplot(data = descriptivos, aes(x = factor(TIPO_CLIENTE))) +
  theme (text = element_text(size = 12)) +
  geom_bar(stat = "count", fill = "#207EE9", colour = "#207EE9", position = position_dodge()) +
  ggtitle("Tipo de Cliente") +
  theme(legend.position = "bottom") +
  theme (plot.title = element_text(size=rel(2),
                                hjust=0.5,
                                face="plain",
                                lineheight=1.5)) +
  labs(x = "Tipo de cliente", y = "Personas encuestadas")

cliente

# edad por tipo de cliente y sexo
desedadygen <- ggplot(descriptivos, aes(x=factor(SEXO), y=EDAD, fill=SEXO)) +
  geom_boxplot() +
  facet_grid(. ~ TIPO_CLIENTE) +
  ggtitle("Edad y género por tipo de cliente") +
  labs(x = "Género", y = "Edad") +
  theme (plot.title = element_text(size=rel(1.5),
                                hjust=0.5,
                                face="plain",
                                lineheight=1.5)) +
  theme(legend.position="none")

desedadygen

# sexo por tipo de cliente
sexocliente <- ggplot(descriptivos, aes(x=SEXO, fill=TIPO_CLIENTE)) +
  geom_bar(stat="count", position=position_dodge()) +
  ggtitle("Género por tipo de cliente") +
  scale_fill_manual(values=c("#59B4F7", "#BBF759")) +
  theme (text = element_text(size=10), plot.title = element_text(hjust=2))

sexocliente

# estrato por tipo de cliente
estratocliente <- ggplot(descriptivos, aes(x=ESTRATO, fill=TIPO_CLIENTE)) +
  geom_bar(stat="count", position=position_dodge()) +
  ggtitle("Estrato por tipo de cliente") +
  scale_fill_manual(values=c("#E2FD47", "#47FDC1")) +
  theme (text = element_text(size=10), plot.title = element_text(hjust=2))

estratocliente

# nivel educativo por tipo de cliente
nivelcliente <- ggplot(descriptivos, aes(x=NIVEL_EDUCATIVO, fill=TIPO_CLIENTE)) +
  geom_bar(stat="count", position=position_dodge()) +
  ggtitle("Nivel educativo por tipo de cliente")+
  scale_fill_manual(values=c("#6AF677", "#16B624")) +
  theme (text = element_text(size=10), plot.title = element_text(hjust=5))

nivelcliente

# ocupación por tipo de cliente
ocucliente <- ggplot(descriptivos, aes(x=PROFESION, fill=TIPO_CLIENTE)) +
  geom_bar(stat="count", position=position_dodge()) +
  ggtitle("Ocupación por tipo de cliente")+
  scale_fill_manual(values=c("#6A99F6", "#4433BB")) +
  theme (text = element_text(size=10), plot.title = element_text(hjust=3))

ocucliente

grid.arrange(sexocliente, estratocliente, nivelcliente, ocucliente, ncol=2)

# edad promedio por tipo de cliente
d <- data.frame(descriptivos$TIPO_CLIENTE, descriptivos$EDAD, descriptivos$SEXO)

with(d, aggregate(d$descriptivos.EDAD, list(d$descriptivos.SEXO, d$descriptivos.TIPO_CLIENTE), mean))
with(d, tapply(d$descriptivos.EDAD, list( d$descriptivos.TIPO_CLIENTE), mean))

# análisis univariados

```

```
# Un primer análisis univariado:
# Cómo la escogencia se afecta por el nivel del precio CLIENTES
pander(xtabs(RES ~ X600, data=dataset1_CLIENTES))
pander(xtabs(RES ~ litro, data=dataset1_CLIENTES))
pander(xtabs(RES ~ gas, data=dataset1_CLIENTES))
pander(xtabs(RES ~ funcional, data=dataset1_CLIENTES))
pander(xtabs(RES ~ recta, data=dataset1_CLIENTES))
pander(xtabs(RES ~ lágrima, data=dataset1_CLIENTES))

# Cómo la escogencia se afecta por el nivel del precio CFINALES
pander(xtabs(RES ~ X600, data=dataset1_CONS))
pander(xtabs(RES ~ litro, data=dataset1_CONS))
pander(xtabs(RES ~ gas, data=dataset1_CONS))
pander(xtabs(RES ~ funcional, data=dataset1_CONS))
pander(xtabs(RES ~ recta, data=dataset1_CONS))
pander(xtabs(RES ~ lágrima, data=dataset1_CONS))
```