



**Universidad  
Zaragoza**

# **[ Aplicación del lenguaje “R”: El modelo de conducta del consumidor] (Anexo)**

**Directores: Joaquín Andaluz y Gloria Jarne.**

**Autor: Pablo Puertas Miramón**

**Grado de Economía**

**Línea de trabajo: “Herramientas informáticas en la optimización y análisis de datos”.**

## I. ANEXOS

### ANEXO 1: Representación gráfica de una función.

$y=f(x)$

```
eq = function(x){FUNCIÓN}  
curve(eq, from=1, to=50, xlab="x", ylab="y")
```

### ANEXO 2: Representación gráfica de curvas de nivel.

```
fun1<-function(x1) (FUNCIÓN 1)  
fun2<-function(x1) (FUNCIÓN 2)  
fun3<-function(x1) (FUNCIÓN 3)  
fun4<-function(x1) (FUNCIÓN 4)  
fun5<-function(x1) (FUNCIÓN 5)
```

```
x1<-seq(0,2*pi,0.2)  
x2<-seq(0,2*pi,0.2)
```

```
matplot(x1,xlab="NOMBRE EJE X",ylab="NOMBRE EJE Y",  
cbind(fun1(x1),fun2(x1),fun3(x1),fun4(x1),fun5(x1)),type="l",col=c("blue","red"  
))
```

### ANEXO 3: Representación gráfica de una función de dos variables. $z = f(x, y)$

```
x <- seq(10, 30, length = 30)  
y <- x  
f <- function(x, y) {FUNCIÓN}  
z <- outer(x, y, f)
```

```
persp(x, y, z, theta = 100, phi = 30, expand = 0.5, col = "lightblue",
xlab="NOMBRE EJE X", ylab="NOMBRE EJE Y", zlab="NOMBRE EJE
Z")
```

#### **ANEXO 4: Cálculo de derivadas.**

```
D(expression((FUNCIÓN), 'x')
```

#### **ANEXO 5: Cálculo de la matriz Hessiana**

```
func <- expression(FUNCIÓN)
```

```
vars <- c("x", "y")
```

```
funcD <- sapply(vars, function(v) D(func, v))
```

```
funcDD <- matrix(list(), 2,2)
```

```
for (i in 1:length(vars))
```

```
    funcDD[,i] <- sapply(vars, function(v) D(funcD[[i]], v))
```

```
funcD <- sapply(vars, function(v) D(func, v))
```

```
funcDD <- matrix(list(), 2,2)
```

```
for (i in 1:length(vars))
```

```
    funcDD[,i] <- sapply(vars, function(v) D(funcD[[i]], v))
```

```
matrix(sapply(funcDD, eval, env=list(x=0, y=pi)), length(vars))
```

## ANEXO 6: Cálculo de las funciones de demanda marshallianas.

Para calcular las funciones de demanda marshallianas, introduciremos  $x_1$  despejado de la restricción presupuestaria y la introduciremos en la función de utilidad para despejar  $x_2$ .

Derivaremos el resultado igualándolo a 0.

```
ECUACION <- function(z)
```

```
{z<- D(expression(FUNCIÓN), "(variable despejada)")
```

```
  return(variable despejada)}
```

```
ECUACION(0)
```