

Gràfics de funcions de dues variables

f.u. et al.

12/8/2020

Obre Rstudio per seguir aquesta activitat

Si no recordes bé com treballar amb Rstudio, ves ara a: [link](#). En tot cas, obre Rstudio per seguir aquesta activitat, huaies de copiar/enganxar les instruccions R que hi apareixen i comprovar que entens el que fan i que saps fer-ho.

Ordres bàsiques de R per graficar funcions

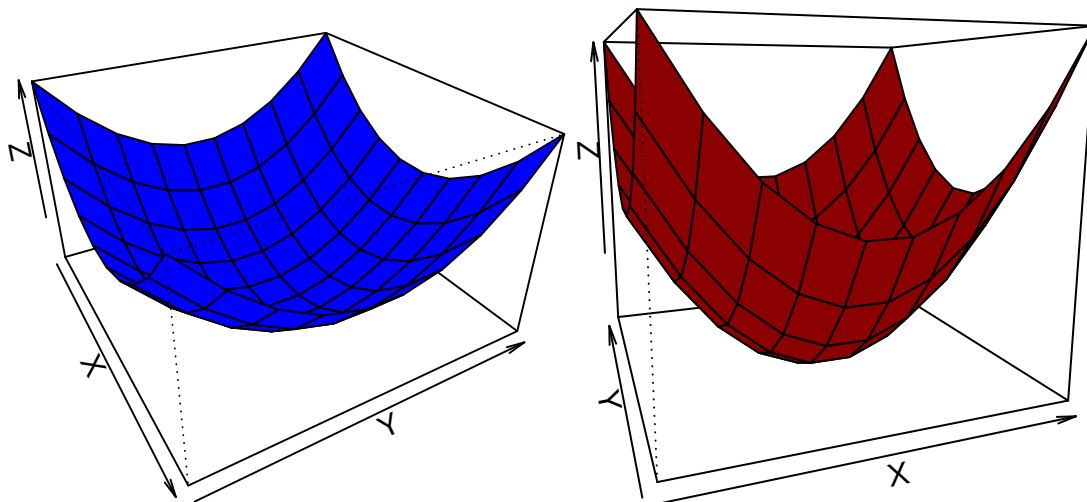
Quan fem la gràfica d'una funció $y = f(x)$ necessitem dues dimensions i el graf és una línia unidimensional. Ara volem utilitzar funcions de dues variables com ara $z = f(x, y) = x^2 + y^2$. Necessitarem tres dimensions, una per cada variable i una tercera pels valors de la funció z i ara el graf de la funció serà una superfície.

Fem-ho per la funció anterior:

```
# primer definim la funció
f <- function(x,y) return(x^2+y^2)

# definim els valors de x i de y on volem avaluar la funció
X <- seq(-1, 1, length = 10)
Y <- seq(-1, 1, length = 10)
# i ara fabriquem una matriu de valors z per cada parella x, y
Z <- outer(X,Y,f)

# ja podem graficar:
persp(X,Y,Z, theta = 60, phi = 30, expand = 0.6, col="blue")
persp(X,Y,Z, theta = -20, phi = 10, expand = 0.9, col="darkred")
```

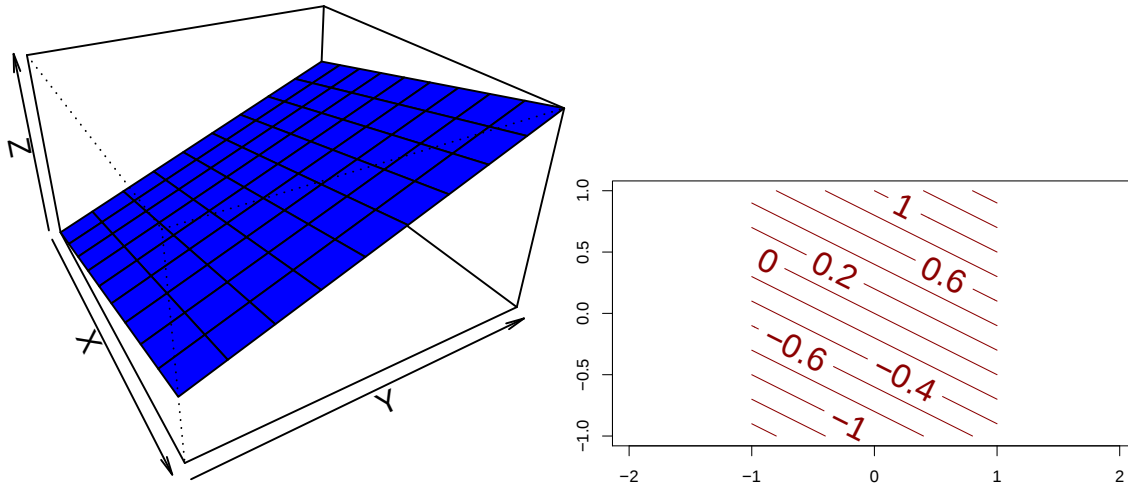


A la funció **persp** li podem dir la posició de la “càmera” des de la que veiem la perspectiva. Són els angles **theta** (esquerra/dreta) i **phi** (dalt/baix). També podem dir-li el color i altres coses...

Les funcions més simples són les lineals. Grafiquem $z = g(x, y) = \frac{1}{2}x + y$ juntament amb les seves línies de nivell.

```
# primer definim la funció
g <- function(x,y) return(x/2+y)
X <- seq(-1, 1, length = 10)
Y <- seq(-1, 1, length = 10)
# i ara fabriquem una matriu de valors z per cada parella x, y
Z <- outer(X,Y,g)

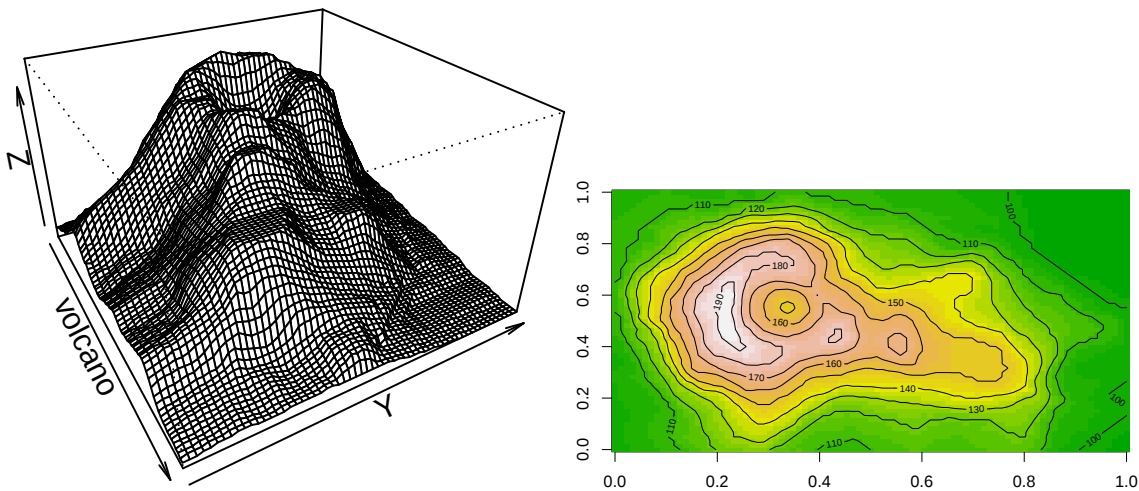
# ja podem graficar:
persp(X,Y,Z, theta = 60, phi = 30, expand = 0.6, col="blue")
# i les línies de nivell
contour(X,Y,Z, col="darkred", nlevels = 12, labcex=2, asp=1)
```



Per entendre millor les corbes de nivell pot anar bé mirar un mapa topogràfic.

```
persp(volcano, theta = 60, phi = 30, expand = 0.6)

image(volcano, col = terrain.colors(25))
contour(volcano, add = TRUE)
```



Observa bé aquestes figures. A l'esquerra la superfície en 3D del volcà et dona la idea de la forma de la muntanya, però és difícil veure clarament la posició dels elements interessants. En canvi, a la dreta, les corbes de nivell permeten situar per exemple el punt més alt. Fixa't que cada corba de nivell porta escrita l'alçada (són xifres una mica petites, amplia la imatge si no ho veus). En aquest mapa, s'ha dibuixat una corba de nivell per cada 10 metres d'alçada. Pots veure aproximadament les coordenades del punt més alt del volcà? (Diríem que són 0.22 i 0.5, aproximadament).

Per això ppreferim la representació en corbes de nivell (o isonivells) més que no la de superfície.

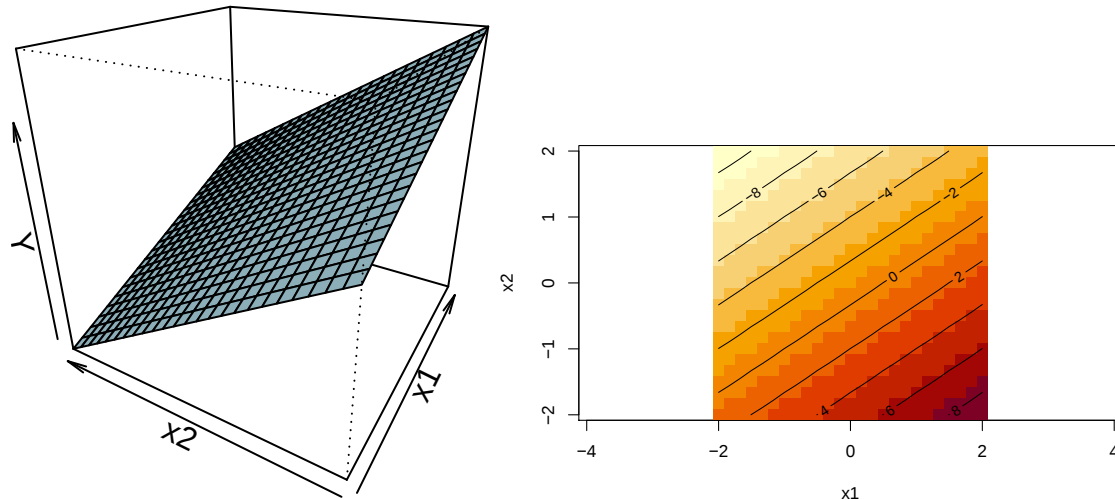
Alguns exemples de funcions senzilles, o no tant

Vegem altres exemples de funcions i les seves representacions gràfiques.

Una funció lineal $f(x, y) = 2x - 3y - 1$

```
f <- function(x1, x2) return(2*x1-3*x2-1)
x1 <- seq(-2, 2, length = 25)
x2 <- seq(-2, 2, length = 25)
Y <- outer(x1, x2, f)

persp(x1, x2, Y, theta = -60, phi = 25, shade = 0.9, col = "lightblue")
image(x1, x2, Y, asp=1)
contour(x1, x2, Y, add = TRUE, lwd = 0.7, labcex = 0.8)
```

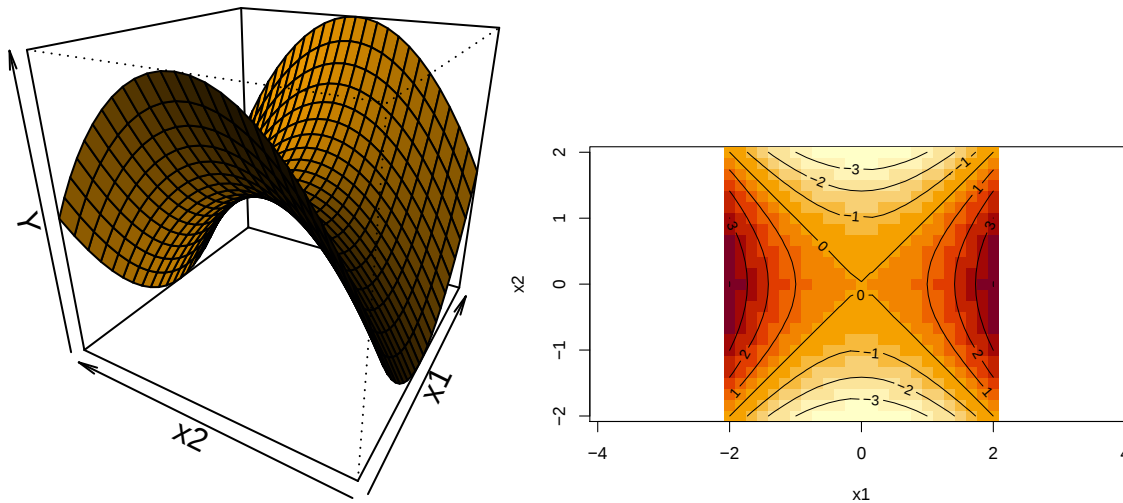


La funció **image** ens dona el gràfic de colors que visualitza l'alçada de la superfície. La funció **contour** dibuixa els isonivells amb els rètols de l'alçada corresponent.

Un funció polinòmica de grau 2: $f(x, y) = x^2 - y^2$:

```
f <- function(x1, x2) return(x1^2-x2^2)
x1 <- seq(-2, 2, length = 25)
x2 <- seq(-2, 2, length = 25)
Y <- outer(x1, x2, f)

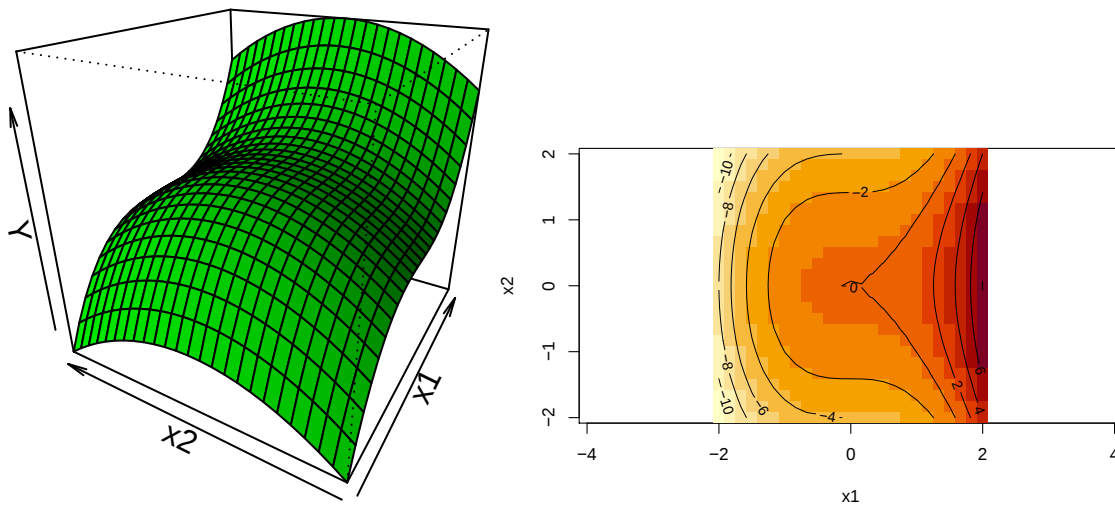
persp(x1, x2, Y, theta = -60, phi = 25, shade = 0.9, col = "orange")
image(x1, x2, Y, asp=1)
contour(x1, x2, Y, add = TRUE, lwd = 0.7, labcex = 0.8)
```



Un funció polinòmica de grau 3: $f(x, y) = x^3 - x^2$:

```
f <- function(x1, x2) return(x1^3-x2^2)
x1 <- seq(-2, 2, length = 25)
x2 <- seq(-2, 2, length = 25)
Y <- outer(x1, x2, f)

persp(x1, x2, Y, theta = -60, phi = 25, shade = 0.9, col = "green")
image(x1, x2, Y, asp=1)
contour(x1, x2, Y, add = TRUE, lwd = 0.7, labcex = 0.8)
```

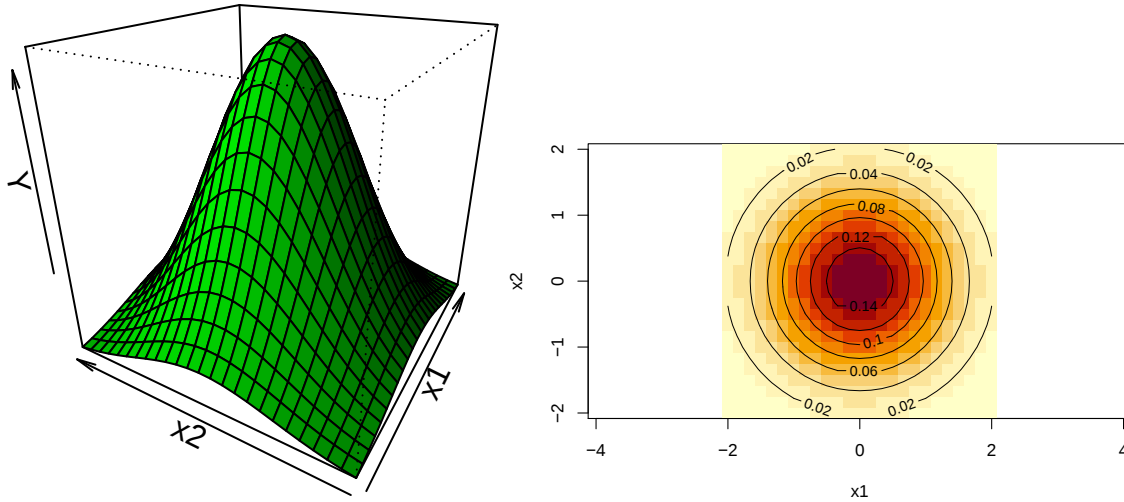


Una funció exponencial:

$$f(x_1, x_2) = \frac{1}{2\pi} e^{-\frac{1}{2}(x_1^2 + x_2^2)}$$

```
f <- function(x1, x2) return(1/(2*pi)*exp(-(x1^2+x2^2)/2))
x1 <- seq(-2, 2, length = 25)
x2 <- seq(-2, 2, length = 25)
Y <- outer(x1, x2, f)

persp(x1, x2, Y, theta = -60, phi = 25, shade = 0.9, col = "green")
image(x1, x2, Y, asp=1)
contour(x1, x2, Y, add = TRUE, lwd = 0.7, labcex = 0.8)
```



Potser has reconegut la *campana de Gauss* bivariant, la funció de densitat de probabilitat normal bivariant.

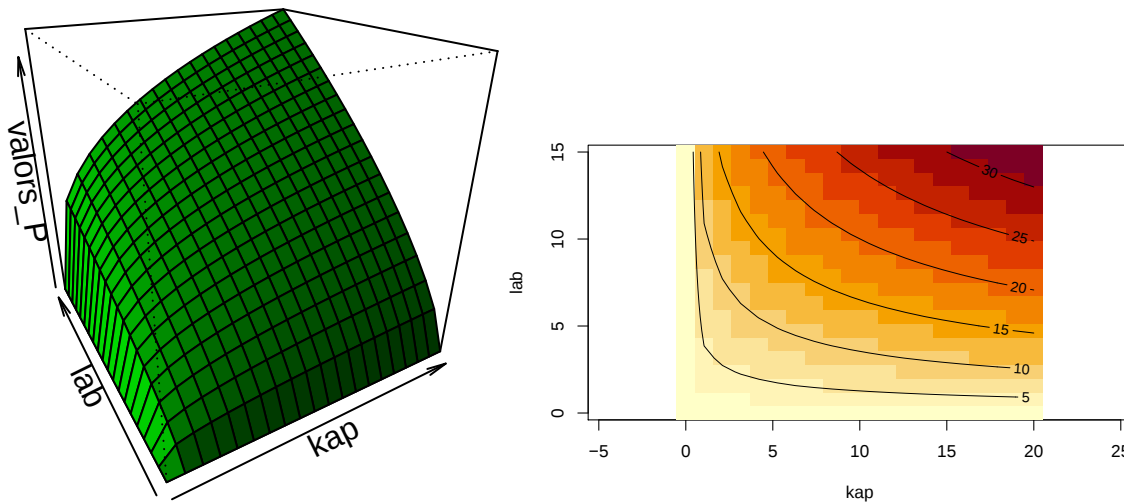
Una funció de producció Cobb-Douglas

Ja has trobat funcions com aquesta en alguns models econòmics la producció en funció del capital i el treball:

$$P(k, l) = K k^a l^b, \quad \text{amb constants } a, b, K > 0, \quad a + b = 1$$

```
a <- 1/3
b <- 1 - a
K <- 2
P <- function(k,l) return(K * k^a * l^b)
kap <- seq(0, 20, length = 20)
lab <- seq(0, 15, length = 20)
valors_P <- outer(kap, lab, P)

persp(kap, lab, valors_P, theta = -30, phi = 25, shade = 0.9, col = "green")
image(kap, lab, valors_P, asp=1)
contour(kap, lab, valors_P, add = TRUE, lwd = 0.7, labcex = 0.8)
```



Efecte 3D amb la llibreria rgl

Una mica més complicat tècnicament és crear figures 3D que puguem bellugar amb el ratolí. Haureu de tenir instal·lada la llibreria **rgl**. Si no la teniu, cal que feu primer `install.packages("rgl")`.

```
library(rgl)

f = function(x,y) return(x^2+y^2)

x <- seq(-1, 1, length = 100)
y <- seq(-1, 1, length = 100)
z <- outer(x,y,f)

## surface3d(x, y, z)
axes3d(edges=c("x-", "y-", "z-"), labels = c("x","y", "z"))
surface3d(x, y, z, col="blue", back="lines", aspect="iso")
```

(Has de introduir el codi a R per veure en una finestra a banda el gràfic que podras girar amb el ratolí)

Aquí podem visualitzar millor el tall de la superfície amb un pla de nivell:

```
library(rgl)

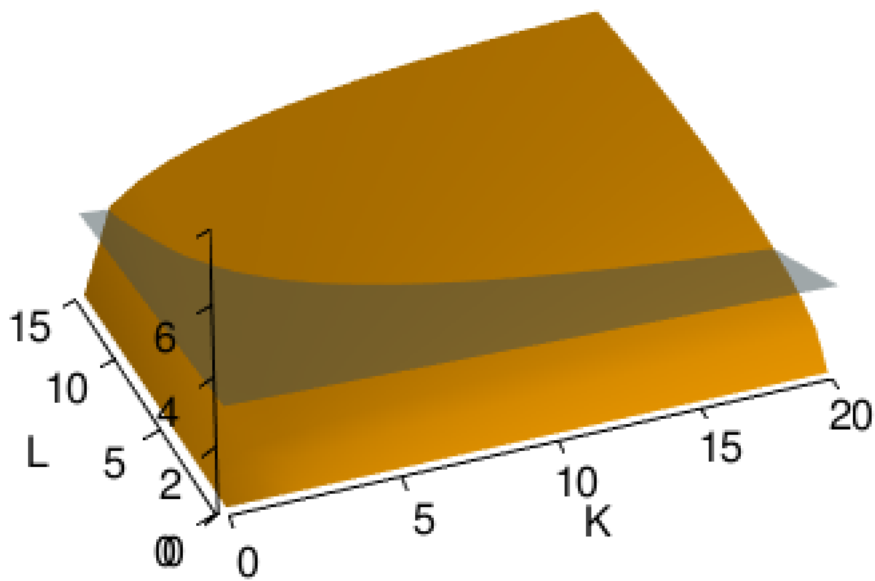
a <- 1/3
b <- 1 - a
K <- 0.5
P <- function(k,l) return(K * k^a * l^b)
npts = 20
kap <- seq(0, 20, length = npts)
lab <- seq(0, 15, length = npts)
valors_P <- outer(kap, lab, P)
nivell = 3

open3d() # nova finestra

glX
1

surface3d(kap, lab, valors_P, col="orange")
axes3d(edges=c("x-", "y-", "z-"))
title3d(xlab="K", ylab="L", zlab="P")
# sobreimposem el pla de nivell constant 3
surface3d(kap, lab, matrix(rep(nivell, npts*npts), nrow=npts),
          alpha=0.5, col="lightblue", add=TRUE)
```

Hauries d'obtenir alguna cosa com això, ànims!



Compara-ho amb l'isonivell 3 de la figura següent.

```
contour(kap, lab, valors_P, levels=seq(1,9,by=2), asp=1)
```

