Ejemplos de uso de R con knitr

Victoria y JA

March 14, 2012

Contents

1 Introducción

Este documento, `eje.Rnw` recoje las potencia del uso de `knitr` con distintos ejemplos de ayuda del propio R. Para compilarlo bastará con ...

```
Rscript -e "library(knitr); knit2pdf('pretty.Rnw')"
```

2 Operadores aritméticos

```
x <- -1:12
x + 1
```

```
##  [1]  0  1  2  3  4  5  6  7  8  9 10 11 12 13
```

```
2 * x + 3
```

```
##  [1]  1  3  5  7  9 11 13 15 17 19 21 23 25 27
```

```
x%%2  #-- is periodic
```

```
##  [1] 1 0 1 0 1 0 1 0 1 0 1 0 1 0
```

```
x%/%5
```

```
##  [1] -1  0  0  0  0  0  1  1  1  1  1  2  2  2
```

3 Función summary

```
summary(attenu, digits = 4)  #-> summary.data.frame(...), default
precision
```

```
##      event            mag           station         dist
##  Min.   : 1.00   Min.   :5.000   117    :  5   Min.   :  0.50
##  1st Qu.: 9.00   1st Qu.:5.300   1028   :  4   1st Qu.: 11.32
##  Median :18.00   Median :6.100   113    :  4   Median : 23.40
##  Mean   :14.74   Mean   :6.084   112    :  3   Mean   : 45.60
##  3rd Qu.:20.00   3rd Qu.:6.600   135    :  3   3rd Qu.: 47.55
##  Max.   :23.00   Max.   :7.700   (Other):147   Max.   :370.00
##                                  NA's   : 16
##      accel
##  Min.   :0.00300
##  1st Qu.:0.04425
##  Median :0.11300
##  Mean   :0.15422
##  3rd Qu.:0.21925
##  Max.   :0.81000
##
```

```
summary(attenu$station, maxsum = 20)  #-> summary.factor(...)
```

```
##     117    1028     113     112     135     475    1030    1083    1093
##       5       4       4       3       3       3       2       2       2
##    1095     111     116    1219    1299     130    1308    1377    1383
##       2       2       2       2       2       2       2       2       2
## (Other)    NA's
##     120      16
```

```
lst <- unclass(attenu$station) > 20  # logical with NAs
## summary.default() for logicals -- different from
#   *.factor:
summary(lst)
```

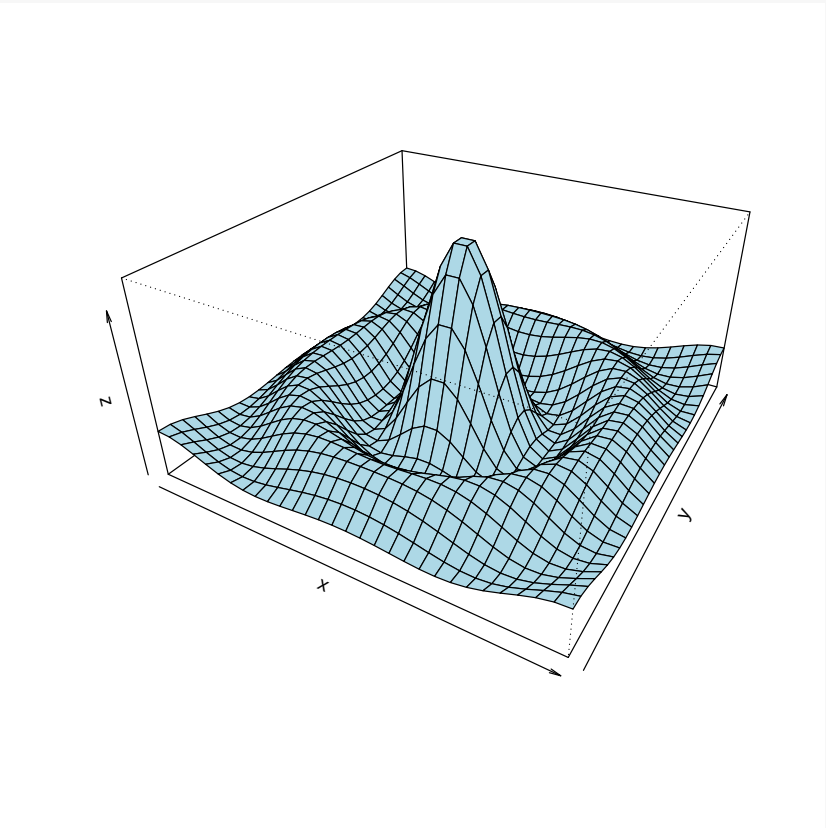
```
##    Mode   FALSE    TRUE    NA's
## logical      28     138      16
```

```
summary(as.factor(lst))
```

```
## FALSE  TRUE  NA's
##    28   138    16
```

4 Graficos con persp

```
require(grDevices)  # for trans3d
## More examples in  demo(persp) !!
##                   -----------

# (1) The Obligatory Mathematical surface.
#     Rotated sinc function.

x <- seq(-10, 10, length = 30)
y <- x
f <- function(x, y) {
    r <- sqrt(x^2 + y^2)
    10 * sin(r)/r
}
z <- outer(x, y, f)
z[is.na(z)] <- 1
op <- par(bg = "white")
persp(x, y, z, theta = 30, phi = 30, expand = 0.5,
    col = "lightblue")
```

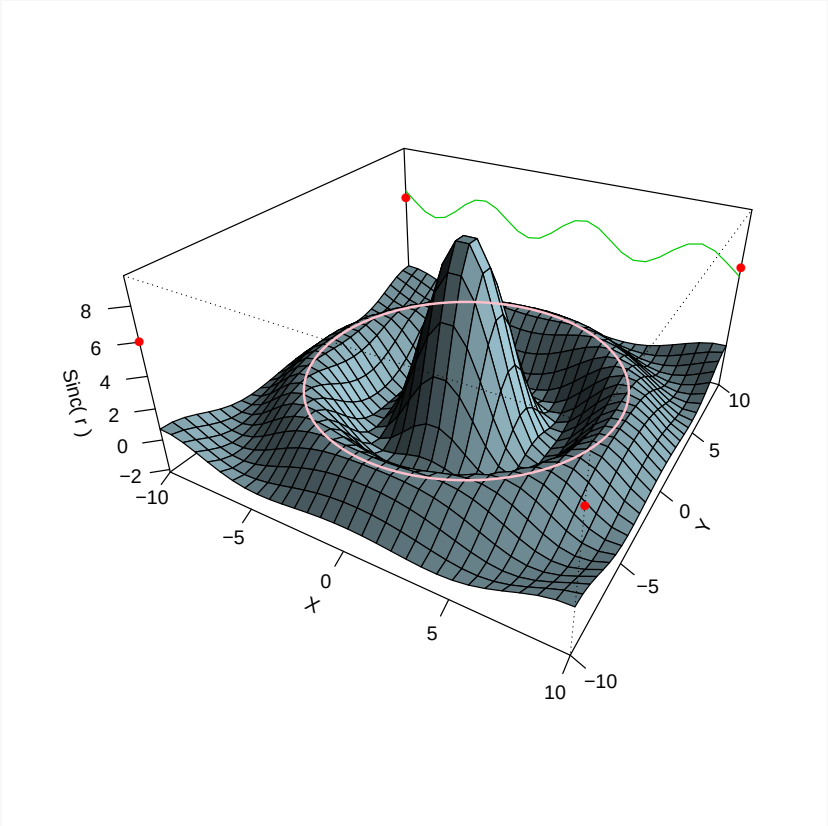



```
res <- persp(x, y, z, theta = 30, phi = 30, expand = 0.5,
    col = "lightblue", ltheta = 120, shade = 0.75, ticktype =
"detailed",
    xlab = "X", ylab = "Y", zlab = "Sinc( r )")
round(res, 3)
```

```
##       [,1]   [,2]   [,3]   [,4]
## [1,] 0.087 -0.025  0.043 -0.043
## [2,] 0.050  0.043 -0.075  0.075
## [3,] 0.000  0.074  0.042 -0.042
## [4,] 0.000 -0.273 -2.890  3.890
```

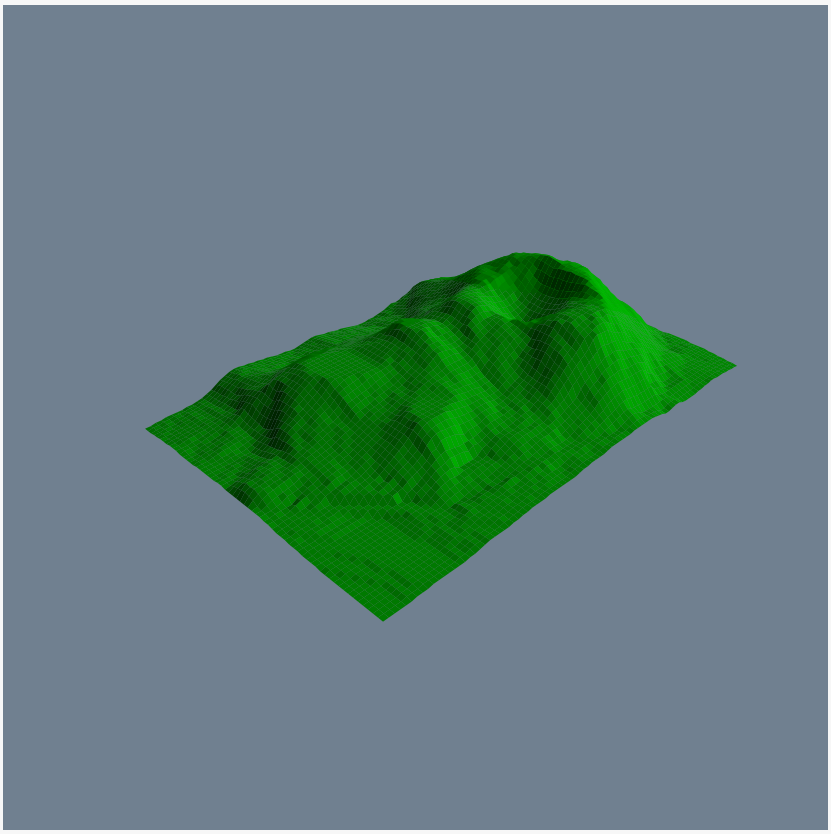
```
# (2) Add to existing persp plot - using trans3d() :

xE <- c(-10, 10)
xy <- expand.grid(xE, xE)
points(trans3d(xy[, 1], xy[, 2], 6, pmat = res),
    col = 2, pch = 16)
```

```r
lines(trans3d(x, y = 10, z = 6 + sin(x), pmat = res),
    col = 3)

phi <- seq(0, 2 * pi, len = 201)
r1 <- 7.725  # radius of 2nd maximum
xr <- r1 * cos(phi)
yr <- r1 * sin(phi)
lines(trans3d(xr, yr, f(xr, yr), res), col = "pink",
    lwd = 2)
```



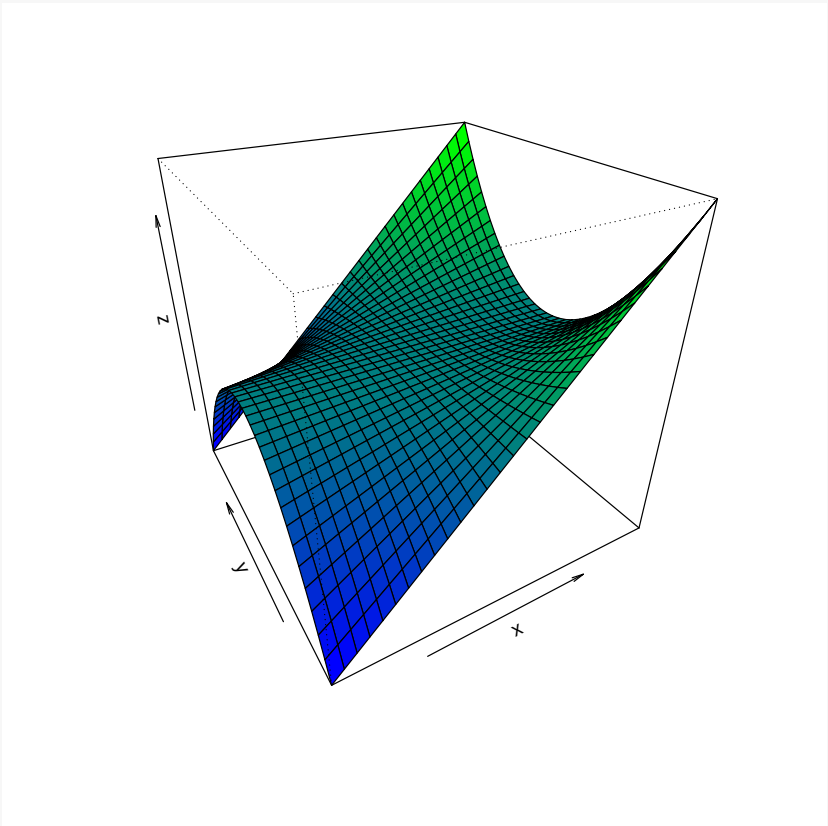

```r
## (no hidden lines)

# (3) Visualizing a simple DEM model

z <- 2 * volcano  # Exaggerate the relief
x <- 10 * (1:nrow(z))  # 10 meter spacing (S to N)
y <- 10 * (1:ncol(z))  # 10 meter spacing (E to W)
## Don't draw the grid lines :  border = NA
par(bg = "slategray")
```

```r
persp(x, y, z, theta = 135, phi = 30, col = "green3",
    scale = FALSE, ltheta = -120, shade = 0.75, border = NA,
    box = FALSE)
```



```r
# (4) Surface colours corresponding to z-values

par(bg = "white")
x <- seq(-1.95, 1.95, length = 30)
y <- seq(-1.95, 1.95, length = 35)
z <- outer(x, y, function(a, b) a * b^2)
nrz <- nrow(z)
ncz <- ncol(z)
# Create a function interpolating colors in the range
#   of specified colors
jet.colors <- colorRampPalette(c("blue", "green"))
# Generate the desired number of colors from this
#   palette
nbcol <- 100
```

```
color <- jet.colors(nbcol)
# Compute the z-value at the facet centres
zfacet <- z[-1, -1] + z[-1, -ncz] + z[-nrz, -1] +
    z[-nrz, -ncz]
# Recode facet z-values into color indices
facetcol <- cut(zfacet, nbcol)
persp(x, y, z, col = color[facetcol], phi = 30,
    theta = -30)
```




```
par(op)
```

5 Regresión lineal

```
require(graphics)
```

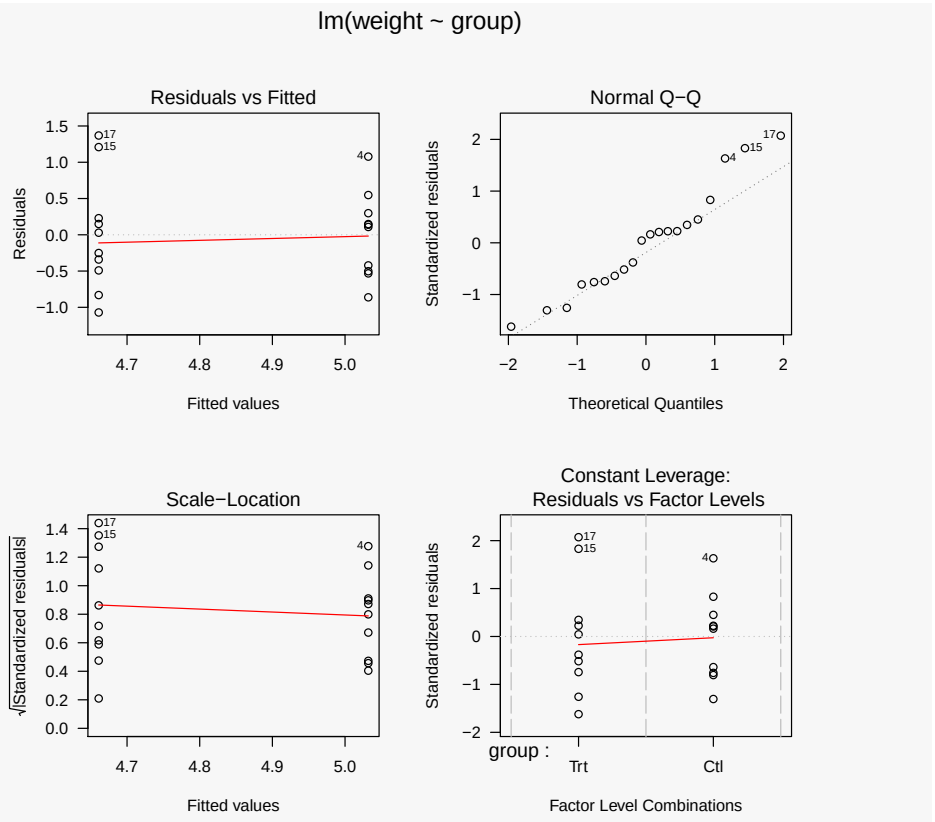
```
## Annette Dobson (1990) 'An Introduction to
#   Generalized Linear Models'.
## Page 9: Plant Weight Data.
ctl <- c(4.17, 5.58, 5.18, 6.11, 4.5, 4.61, 5.17,
    4.53, 5.33, 5.14)
trt <- c(4.81, 4.17, 4.41, 3.59, 5.87, 3.83, 6.03,
    4.89, 4.32, 4.69)
group <- gl(2, 10, 20, labels = c("Ctl", "Trt"))
weight <- c(ctl, trt)
lm.D9 <- lm(weight ~ group)
lm.D90 <- lm(weight ~ group - 1)  # omitting intercept

anova(lm.D9)
```

```
## Analysis of Variance Table
##
## Response: weight
##           Df Sum Sq Mean Sq F value Pr(>F)
## group      1   0.69   0.688    1.42   0.25
## Residuals 18   8.73   0.485
```

```
summary(lm.D90)
```

```
##
## Call:
## lm(formula = weight ~ group - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0710 -0.4938  0.0685  0.2462  1.3690
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## groupCtl     5.03       0.22    22.9  9.5e-15 ***
## groupTrt     4.66       0.22    21.2  3.6e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.696 on 18 degrees of freedom
## Multiple R-squared: 0.982,Adjusted R-squared: 0.98
## F-statistic:  485 on 2 and 18 DF,  p-value: <2e-16
##
```

```
opar <- par(mfrow = c(2, 2), oma = c(0, 0, 1.1,
    0))
plot(lm.D9, las = 1)  # Residuals, Fitted, ...
```




```
par(opar)

### less simple examples in 'See Also' above
```