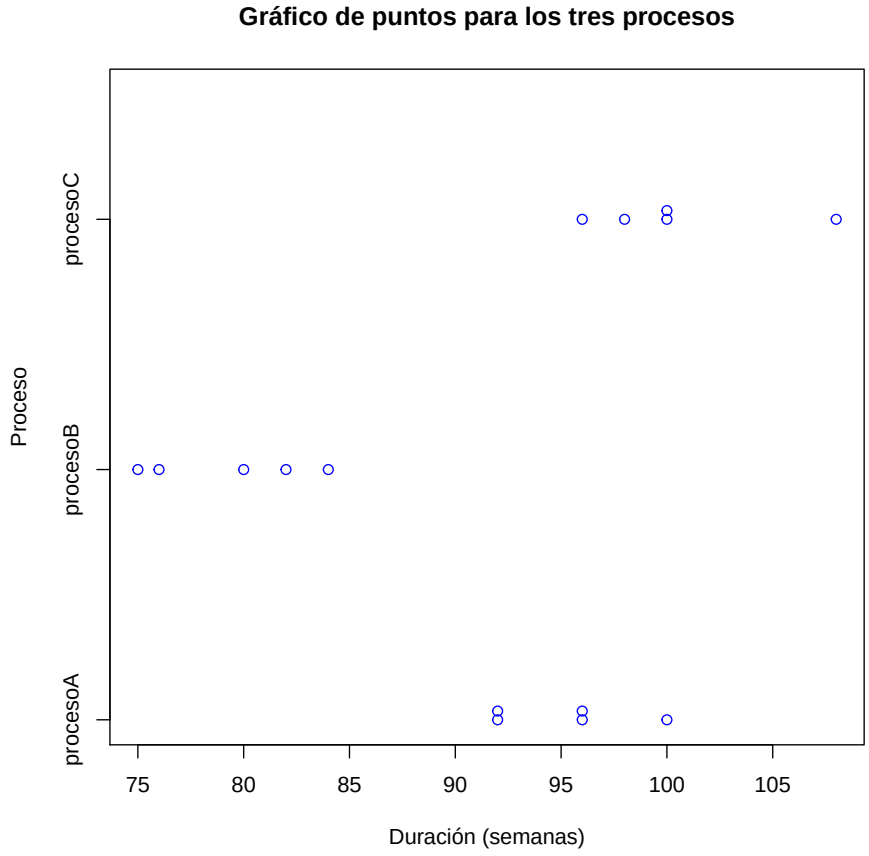
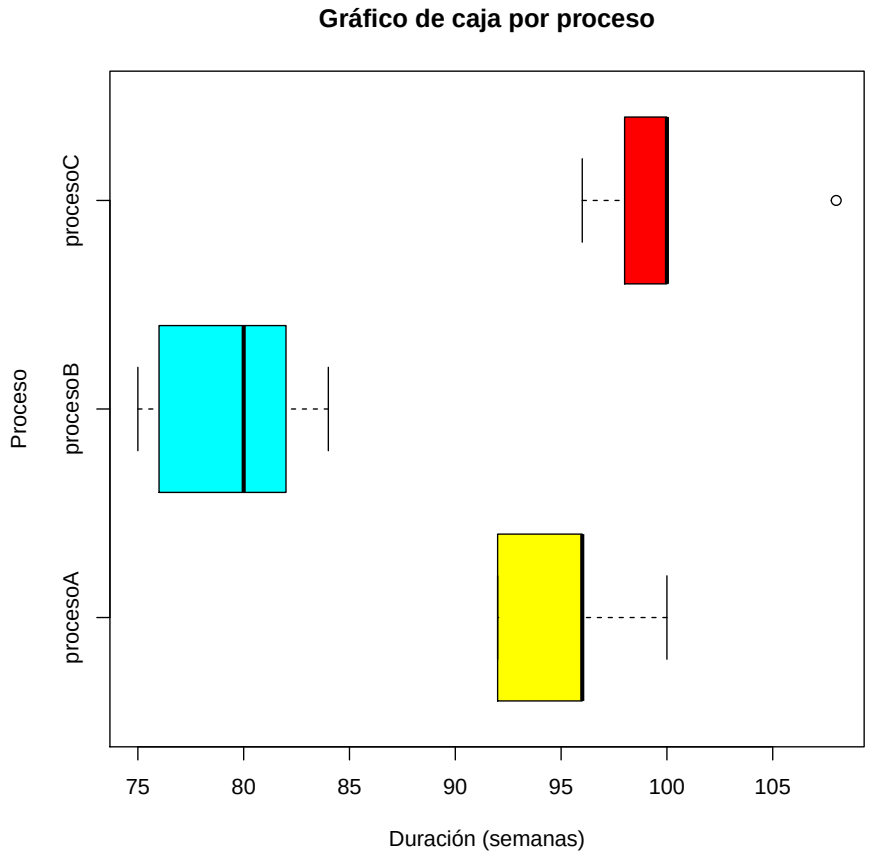
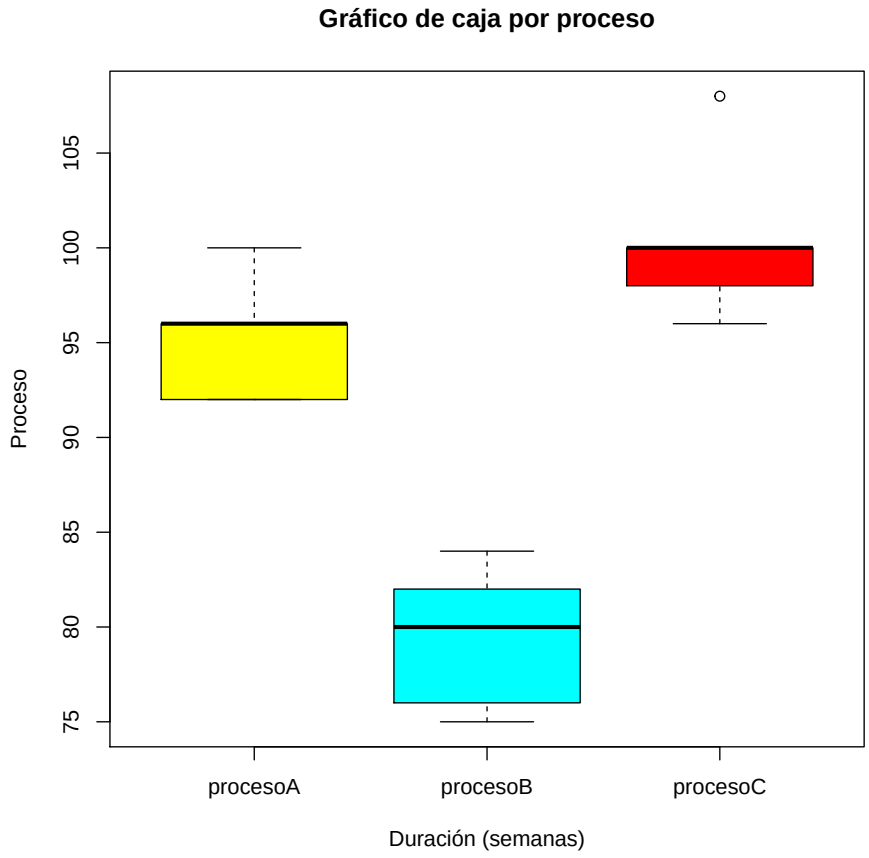
```r
# GUIA 10

A <- c(100,96,92,96,92); A

## [1] 100  96  92  96  92

B <- c(76,80,75,84,82); B

## [1] 76 80 75 84 82

C <- c(108,100,96,98,100); C

## [1] 108 100  96  98 100

Baterias <- data.frame(procesoA=A, procesoB=B, procesoC=C); Baterias

##   procesoA procesoB procesoC
## 1      100       76      108
## 2       96       80      100
## 3       92       75       96
## 4       96       84       98
## 5       92       82      100

fix(Baterias)
write.table(Baterias, file="Baterias.txt", append=FALSE, quote=TRUE, sep=" ", na="NA",col.na
ls(); rm(list=ls(all=TRUE)); ls()

## [1] "A"        "B"        "Baterias" "C"
## character(0)

Baterias <- read.table("Baterias.txt", header=TRUE); Baterias

##   procesoA procesoB procesoC
## 1      100       76      108
## 2       96       80      100
## 3       92       75       96
## 4       96       84       98
## 5       92       82      100

attach(Baterias, pos=2)
search()

##  [1] ".GlobalEnv"        "Baterias"          "package:knitr"
##  [4] "package:stats"     "package:graphics"  "package:grDevices"
##  [7] "package:utils"     "package:datasets"  "package:methods"
## [10] "Autoloads"         "package:base"

stripchart(Baterias, main="Grfico de puntos para los tres procesos", method = "stack", verti
FALSE, col="blue", pch=1, xlab="Duracin (semanas)", ylab="Proceso")
```




```
summary(Baterias)

##     procesoA        procesoB       procesoC
##  Min.   : 92.0   Min.   :75.0   Min.   : 96.0
##  1st Qu.: 92.0   1st Qu.:76.0   1st Qu.: 98.0
##  Median : 96.0   Median :80.0   Median :100.0
##  Mean   : 95.2   Mean   :79.4   Mean   :100.4
##  3rd Qu.: 96.0   3rd Qu.:82.0   3rd Qu.:100.0
##  Max.   :100.0   Max.   :84.0   Max.   :108.0

# Horizontal
boxplot(Baterias, width=NULL, varwidth=TRUE, names, add= FALSE, horizontal = TRUE,
main="Grfico de caja por proceso", border=par("fg"), col=c("yellow", "cyan", "red"), xlab =
"Duracin (semanas)", ylab="Proceso")
```




```
# Vertical
boxplot(Baterias, width=NULL, varwidth=TRUE, names, add= FALSE, horizontal = FALSE,
main="Grfico de caja por proceso", border=par("fg"), col=c("yellow", "cyan", "red"), xlab =
"Duracin (semanas)", ylab="Proceso")
```




```
#Presenta la matriz de covarianzas muestral.
options(digits=3) # slo imprime 3 lugares decimales
S <- var(Baterias); S
```

```
##          procesoA procesoB procesoC
## procesoA     11.2     -1.6     12.4
## procesoB     -1.6     14.8     -4.7
## procesoC     12.4     -4.7     20.8
```

```
Baterias <- stack(Baterias); Baterias
```

```
##    values      ind
## 1     100 procesoA
## 2      96 procesoA
## 3      92 procesoA
## 4      96 procesoA
```

```
## 5      92 procesoA
## 6      76 procesoB
## 7      80 procesoB
## 8      75 procesoB
## 9      84 procesoB
## 10     82 procesoB
## 11    108 procesoC
## 12    100 procesoC
## 13     96 procesoC
## 14     98 procesoC
## 15    100 procesoC
```

```
names(Baterias) # Muestra los encabezados de los vectores
```

```
## [1] "values" "ind"
```

```
detach(Baterias, pos=2); search()
```

```
##  [1] ".GlobalEnv"        "package:knitr"     "package:stats"
##  [4] "package:graphics"  "package:grDevices" "package:utils"
##  [7] "package:datasets"  "package:methods"   "Autoloads"
## [10] "package:base"
```

```
#Anlisis de una variable bidimensional
```

```
Fuma = c("Si","No","No","Si","No","Si","Si","Si","No","Si"); Fuma
```

```
##  [1] "Si" "No" "No" "Si" "No" "Si" "Si" "Si" "No" "Si"
```

```
Cantidad = c(1,2,2,3,3,1,2,1,3,2); Cantidad
```

```
##  [1] 1 2 2 3 3 1 2 1 3 2
```

```
Estudia <- data.frame(Fuma=Fuma, Cantidad=Cantidad); Estudia
```

```
##    Fuma Cantidad
## 1    Si        1
## 2    No        2
## 3    No        2
## 4    Si        3
## 5    No        3
## 6    Si        1
## 7    Si        2
## 8    Si        1
## 9    No        3
## 10   Si        2
```

```r
fix(Estudia)
write.table(Estudia, file="Estudia.txt", append=FALSE, quote=TRUE, sep=" ", na="NA",
col.names=TRUE)
write.table
```

```
## function (x, file = "", append = FALSE, quote = TRUE, sep = " ",
##     eol = "\n", na = "NA", dec = ".", row.names = TRUE, col.names = TRUE,
##     qmethod = c("escape", "double"), fileEncoding = "")
## {
##     qmethod <- match.arg(qmethod)
##     if (is.logical(quote) && (length(quote) != 1L || is.na(quote)))
##         stop("'quote' must be 'TRUE', 'FALSE' or numeric")
##     quoteC <- if (is.logical(quote))
##         quote
##     else TRUE
##     qset <- is.logical(quote) && quote
##     if (!is.data.frame(x) && !is.matrix(x))
##         x <- data.frame(x)
##     makeRownames <- isTRUE(row.names)
##     makeColnames <- is.logical(col.names) && !identical(FALSE,
##         col.names)
##     if (is.matrix(x)) {
##         p <- ncol(x)
##         d <- dimnames(x)
##         if (is.null(d))
##             d <- list(NULL, NULL)
##         if (is.null(d[[1L]]) && makeRownames)
##             d[[1L]] <- seq_len(nrow(x))
##         if (is.null(d[[2L]]) && makeColnames && p > 0L)
##             d[[2L]] <- paste0("V", 1L:p)
##         if (qset)
##             quote <- if (is.character(x))
##                 seq_len(p)
##             else numeric()
##     }
##     else {
##         if (qset)
##             quote <- if (length(x))
##                 which(unlist(lapply(x, function(x) is.character(x) ||
##                   is.factor(x))))
##             else numeric()
##         if (any(sapply(x, function(z) length(dim(z)) == 2 &&
##             dim(z)[2L] > 1))) {
##             c1 <- names(x)
##             x <- as.matrix(x, rownames.force = makeRownames)
##             d <- dimnames(x)
```

```
##             if (qset) {
##                 ord <- match(c1, d[[2L]], 0L)
##                 quote <- ord[quote]
##                 quote <- quote[quote > 0L]
##             }
##         }
##         else d <- list(if (makeRownames) row.names(x), if (makeColnames) names(x))
##         p <- ncol(x)
##     }
##     nocols <- p == 0L
##     if (is.logical(quote))
##         quote <- NULL
##     else if (is.numeric(quote)) {
##         if (any(quote < 1L | quote > p))
##             stop("invalid numbers in 'quote'")
##     }
##     else stop("invalid 'quote' specification")
##     rn <- FALSE
##     rnames <- NULL
##     if (is.logical(row.names)) {
##         if (row.names) {
##             rnames <- as.character(d[[1L]])
##             rn <- TRUE
##         }
##     }
##     else {
##         rnames <- as.character(row.names)
##         rn <- TRUE
##         if (length(rnames) != nrow(x))
##             stop("invalid 'row.names' specification")
##     }
##     if (!is.null(quote) && rn)
##         quote <- c(0, quote)
##     if (is.logical(col.names)) {
##         if (!rn && is.na(col.names))
##             stop("'col.names = NA' makes no sense when 'row.names = FALSE'")
##         col.names <- if (is.na(col.names) && rn)
##             c("", d[[2L]])
##         else if (col.names)
##             d[[2L]]
##         else NULL
##     }
##     else {
##         col.names <- as.character(col.names)
##         if (length(col.names) != p)
```

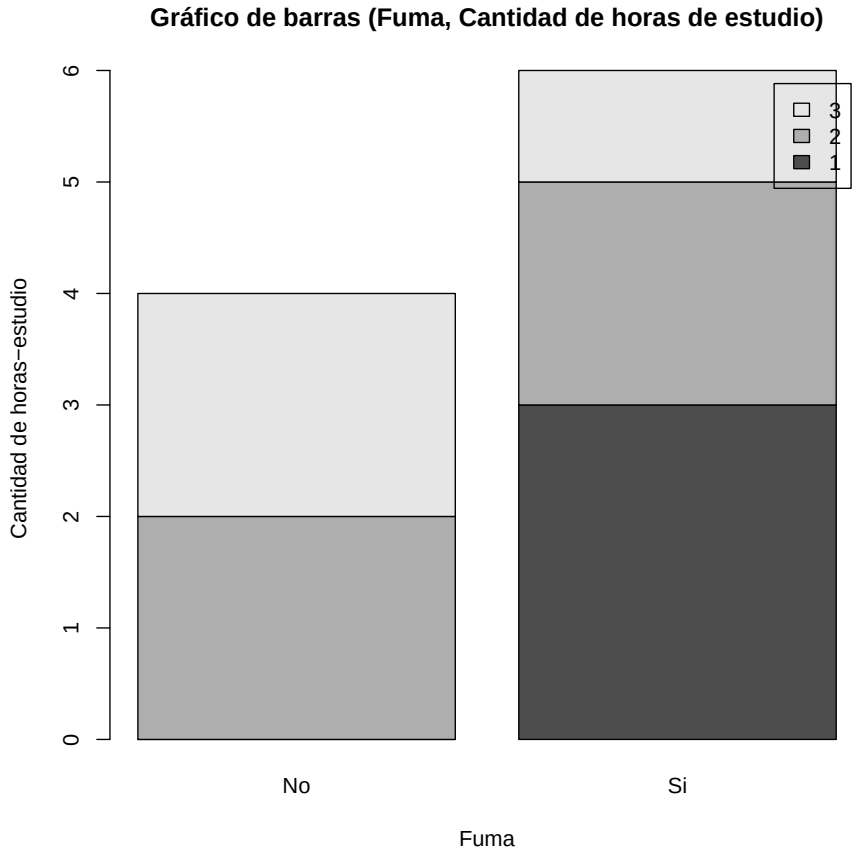
```
##             stop("invalid 'col.names' specification")
##     }
##     if (file == "")
##         file <- stdout()
##     else if (is.character(file)) {
##         file <- if (nzchar(fileEncoding))
##             file(file, ifelse(append, "a", "w"), encoding = fileEncoding)
##         else file(file, ifelse(append, "a", "w"))
##         on.exit(close(file))
##     }
##     else if (!isOpen(file, "w")) {
##         open(file, "w")
##         on.exit(close(file))
##     }
##     if (!inherits(file, "connection"))
##         stop("'file' must be a character string or connection")
##     qstring <- switch(qmethod, escape = "\\\\\"", double = "\"\"")
##     if (!is.null(col.names)) {
##         if (append)
##             warning("appending column names to file")
##         if (quoteC)
##             col.names <- paste("\"", gsub("\"", qstring, col.names),
##                 "\"", sep = "")
##         writeLines(paste(col.names, collapse = sep), file, sep = eol)
##     }
##     if (nrow(x) == 0L)
##         return(invisible())
##     if (nocols && !rn)
##         return(cat(rep.int(eol, NROW(x)), file = file, sep = ""))
##     if (is.matrix(x) && !is.atomic(x))
##         mode(x) <- "character"
##     if (is.data.frame(x)) {
##         x[] <- lapply(x, function(z) {
##             if (is.object(z) && !is.factor(z))
##                 as.character(z)
##             else z
##         })
##     }
##     invisible(.External2(C_writetable, x, file, nrow(x), p, rnames,
##         sep, eol, na, dec, as.integer(quote), qmethod != "double"))
## }
## <bytecode: 0x000000000b141328>
## <environment: namespace:utils>


ls()
```

```
## [1] "Baterias" "Cantidad" "Estudia"  "Fuma"     "S"

rm(list=ls(all=TRUE))
ls()

## character(0)

Estudia <- read.table("Estudia.txt", header=TRUE)
Estudia

##    Fuma Cantidad
## 1    Si        1
## 2    No        2
## 3    No        2
## 4    Si        3
## 5    No        3
## 6    Si        1
## 7    Si        2
## 8    Si        1
## 9    No        3
## 10   Si        2

tablaCont <- table(Estudia)
tablaCont

##     Cantidad
## Fuma 1 2 3
##   No 0 2 2
##   Si 3 2 1

options(digits=3) # slo imprime 3 lugares decimales
propTotal <- prop.table(tablaCont); propTotal

##     Cantidad
## Fuma   1   2   3
##   No 0.0 0.2 0.2
##   Si 0.3 0.2 0.1

propFila <- prop.table(tablaCont, 1)
propFila

##     Cantidad
## Fuma     1     2     3
##   No 0.000 0.500 0.500
##   Si 0.500 0.333 0.167

propCol <- prop.table(tablaCont, 2)
propCol
```

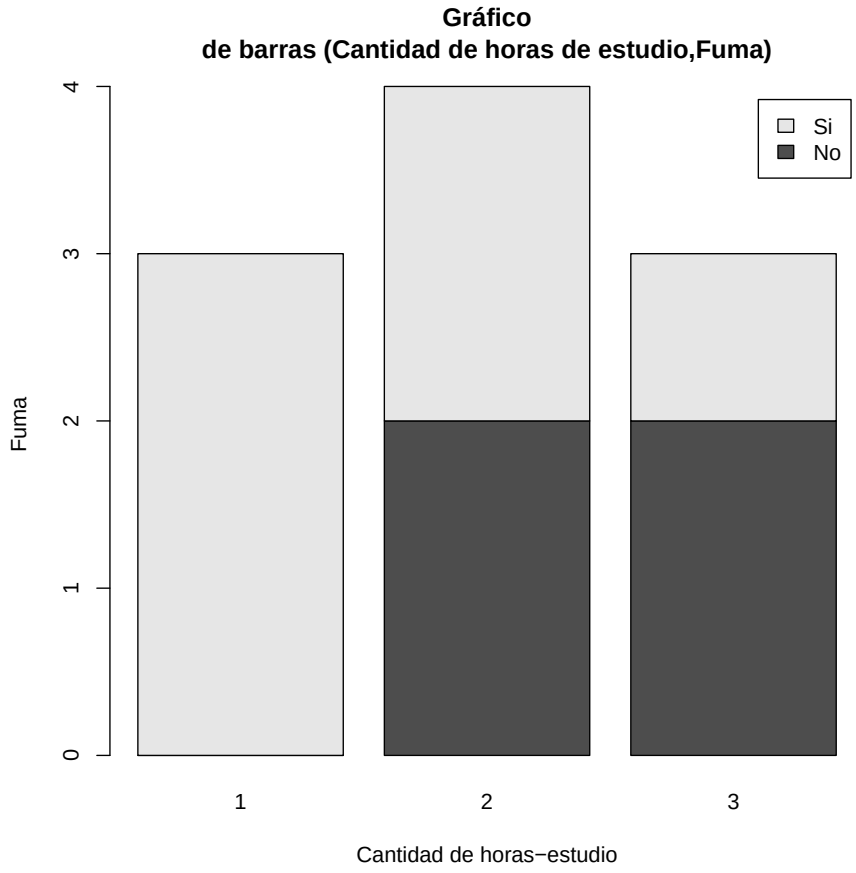
```
##     Cantidad
## Fuma     1     2     3
##   No 0.000 0.500 0.667
##   Si 1.000 0.500 0.333
```

```
barplot(table(Estudia$Cantidad, Estudia$Fuma), beside = FALSE, horizontal=FALSE, main="Grfic
```

```
## Warning in plot.window(xlim, ylim, log = log, ...):  "horizontal"
is not a graphical parameter
## Warning in axis(if (horiz) 2 else 1, at = at.l, labels = names.arg,
lty = axis.lty, :  "horizontal" is not a graphical parameter
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab,
...):  "horizontal" is not a graphical parameter
## Warning in axis(if (horiz) 1 else 2, cex.axis = cex.axis, ...):
"horizontal" is not a graphical parameter
```

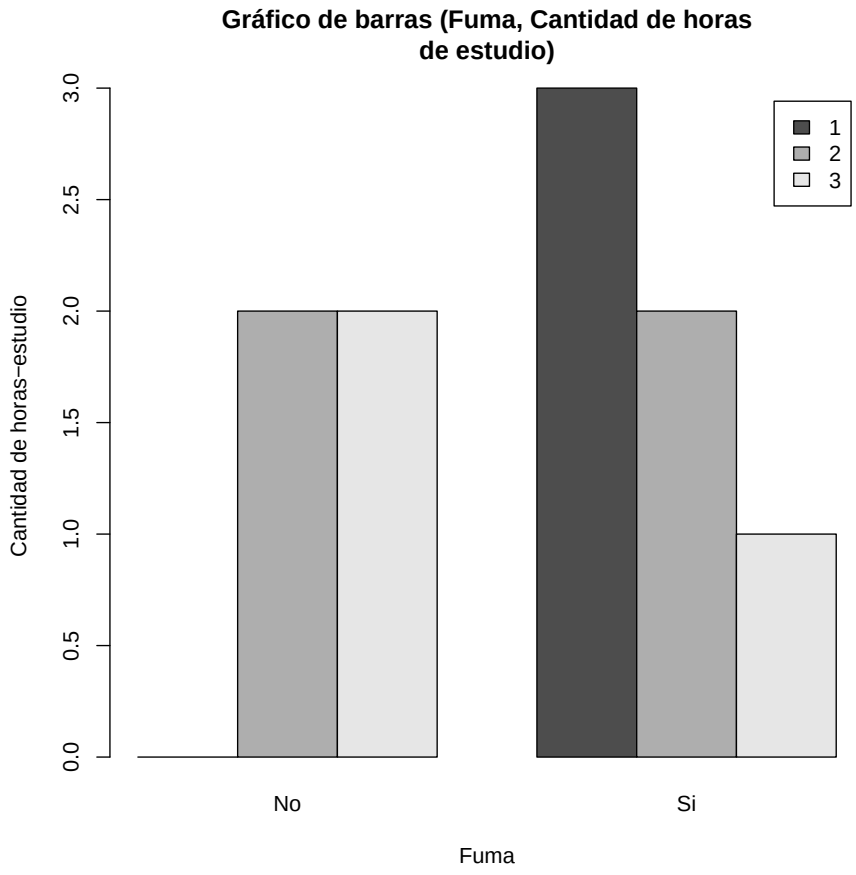


```
barplot(table(Estudia$Fuma, Estudia$Cantidad), beside = FALSE, horizontal=FALSE,main="Grfico
de barras (Cantidad de horas de estudio,Fuma)", legend.text =T, xlab="Cantidad de horas-estu
ylab="Fuma")

## Warning in plot.window(xlim, ylim, log = log, ...):  "horizontal"
is not a graphical parameter
## Warning in axis(if (horiz) 2 else 1, at = at.l, labels = names.arg,
lty = axis.lty, :  "horizontal" is not a graphical parameter
## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab,
...):  "horizontal" is not a graphical parameter
## Warning in axis(if (horiz) 1 else 2, cex.axis = cex.axis, ...):
"horizontal" is not a graphical parameter
```

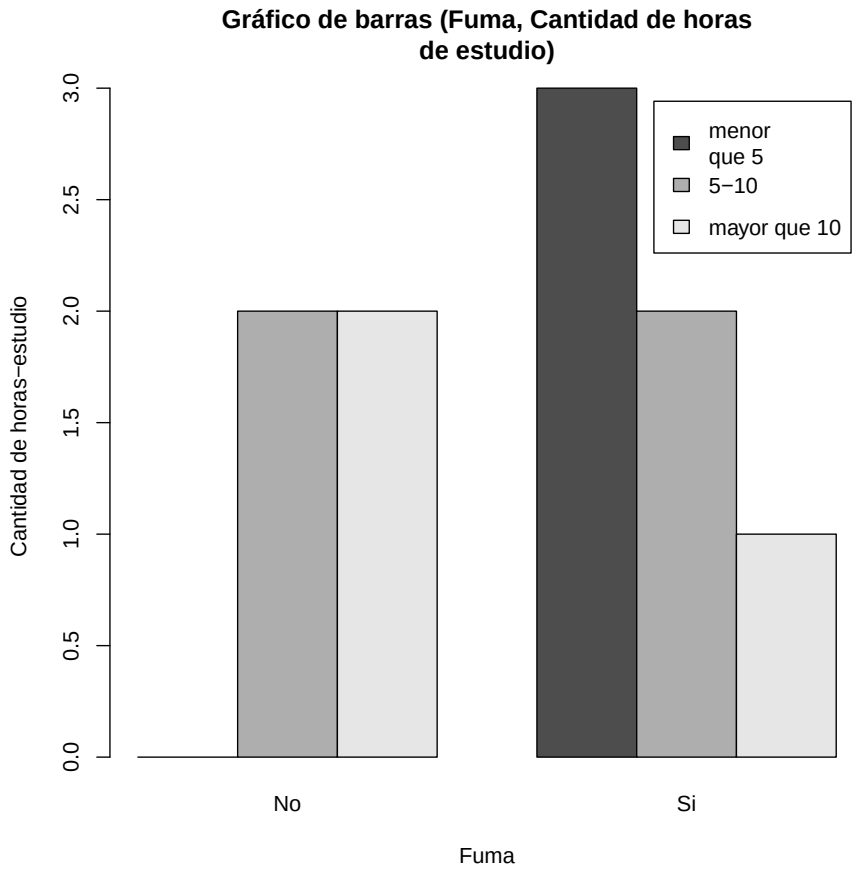



```
Fuma=factor(Estudia$Fuma); Fuma

##  [1] Si No No Si No Si Si Si No Si
```

```
## Levels: No Si

barplot(table(Estudia$Cantidad, Estudia$Fuma), main="Grfico de barras (Fuma, Cantidad de hor
de estudio)", xlab="Fuma", ylab="Cantidad de horas-estudio", beside=TRUE, legend.text=T)
```




```
barplot(table(Estudia$Cantidad, Estudia$Fuma), main="Grfico de barras (Fuma, Cantidad de hor
de estudio)", xlab="Fuma", ylab="Cantidad de horas-estudio", beside=TRUE, legend.text=c("men
que 5", "5-10", "mayor que 10"))
```




```
chisq.test(tablaCont)

## Warning in chisq.test(tablaCont):  Chi-squared approximation may
be incorrect

##
##  Pearson's Chi-squared test
##
## data:  tablaCont
## X-squared = 3, df = 2, p-value = 0.2

#Como p-valor es mayor que 0.05, se concluye
#que las variables con independientes
```