

**UNIVERSIDAD DE EL SALVADOR
FACULTAD MULTIDISCIPLINARIA DE OCCIDENTE
DEPARTAMENTO DE MATEMÁTICA**



Licenciatura en Estadística

Control Estadístico del Paquete R

”UNIDAD DOS”

**Alumna:
Erika Beatríz Guillén Pineda**

**Fecha de elaboración
Santa Ana - 27 de noviembre de 2015**

ANÁLISIS DE UNA VARIABLE BIDIMENSIONAL (CATEGÓRICA, CONTINUA)

Ejemplo 1: Se están estudiando tres procesos (A, B, C) para fabricar pilas o baterías. Se sospecha que el proceso incide en la duración (en semanas) de las baterías, es decir, que la duración (en semanas) de los procesos es diferente. Se seleccionan aleatoriamente cinco baterías de cada proceso y al medirles aleatoriamente su duración los datos que se obtienen

1) Activa tu directorio de trabajo

```
getwd()

## [1] "C:/Users/User/Documents/TODAS_PRACTICAS"

setwd("C:/Users/User/Documents/TODAS_PRACTICAS")
```

2) Crea un nuevo script y llamarle "Script10-DatosBivariados2"

3) Crea un vector de datos para cada proceso descrito en el problema.

```
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
```

4) Crea una hoja de datos teniendo como componentes (columnas) los tres vectores (se puede hacer pues el número de datos en cada proceso es igual, de lo contrario se debería de crear dos variables una para la duración de cada proceso y otra para identificar a qué proceso corresponde).

```
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
```

```
# Para editar los datos puede utilizar la función fix() fix(Baterias)
fix(Baterias)
```

5) Guarda la hoja de datos en un archivo.

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

6) Elimina todos objetos que existen en el espacio de trabajo (Workspace)

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

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

7) Recupera la hoja de datos, para probar si fue guardada.

```
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
```

8) Conecta o adjunta la hoja de datos a la segunda ruta o lista de búsqueda.

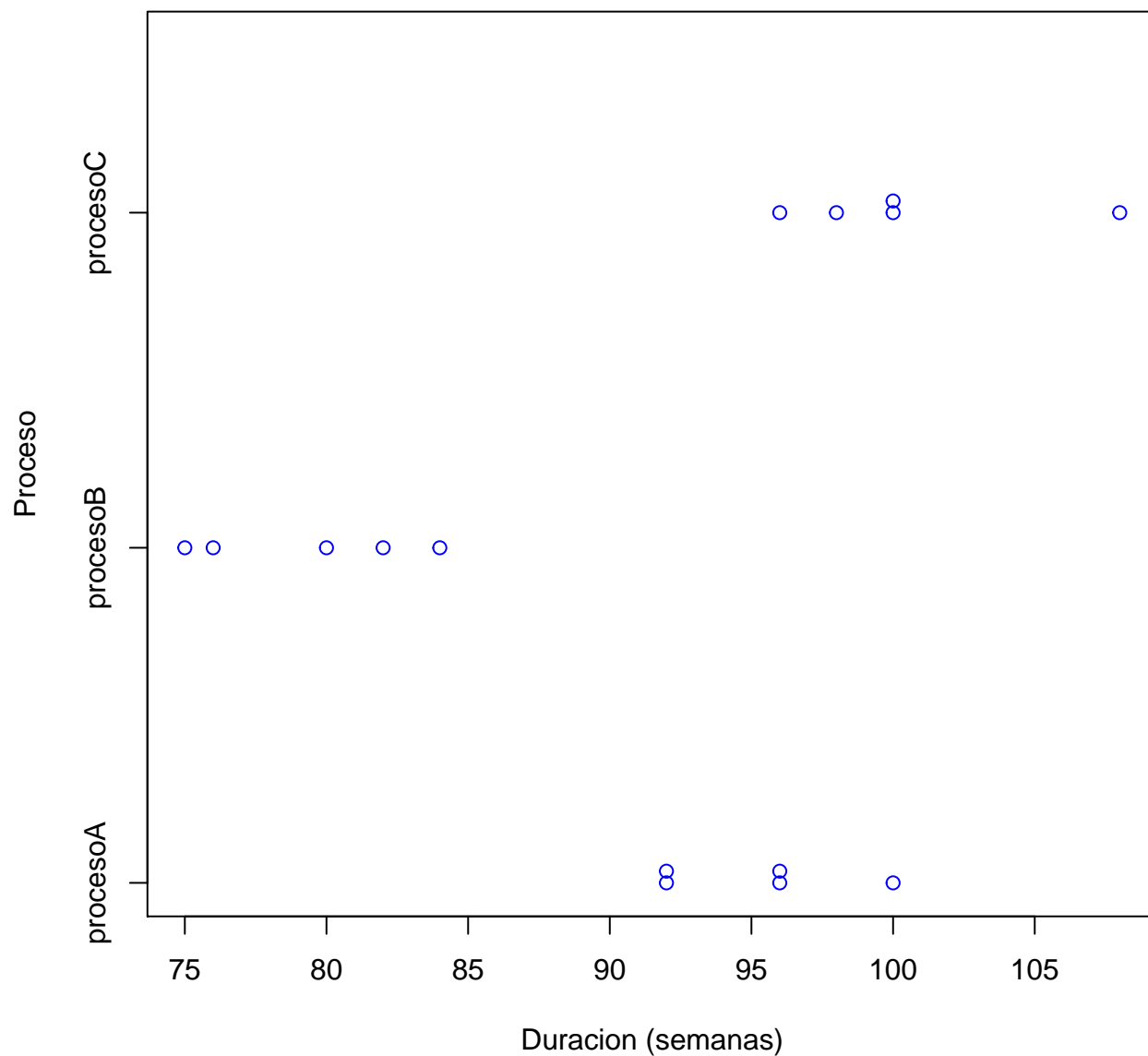
```
attach(Baterias, pos=2)
search()

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

9) Dibuja un gráfico horizontal de puntos para los tres procesos.

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

Grafico de puntos para los tres procesos



Note que con ayuda de este gráfico podemos observar si los tres procesos se comportan

10) Muestra un resumen estadístico para los tres procesos.

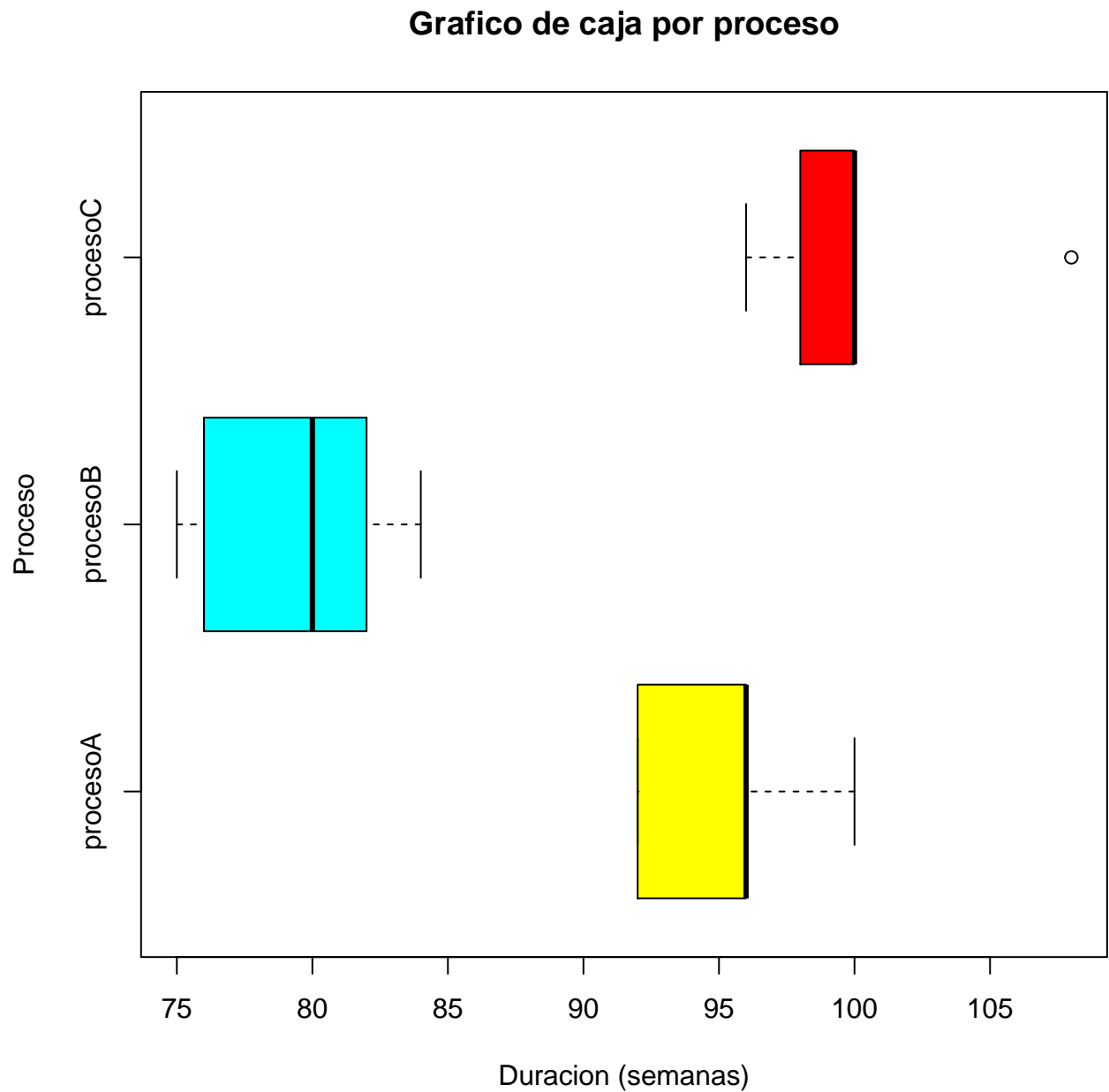
```
summary(Baterias)
```

##	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

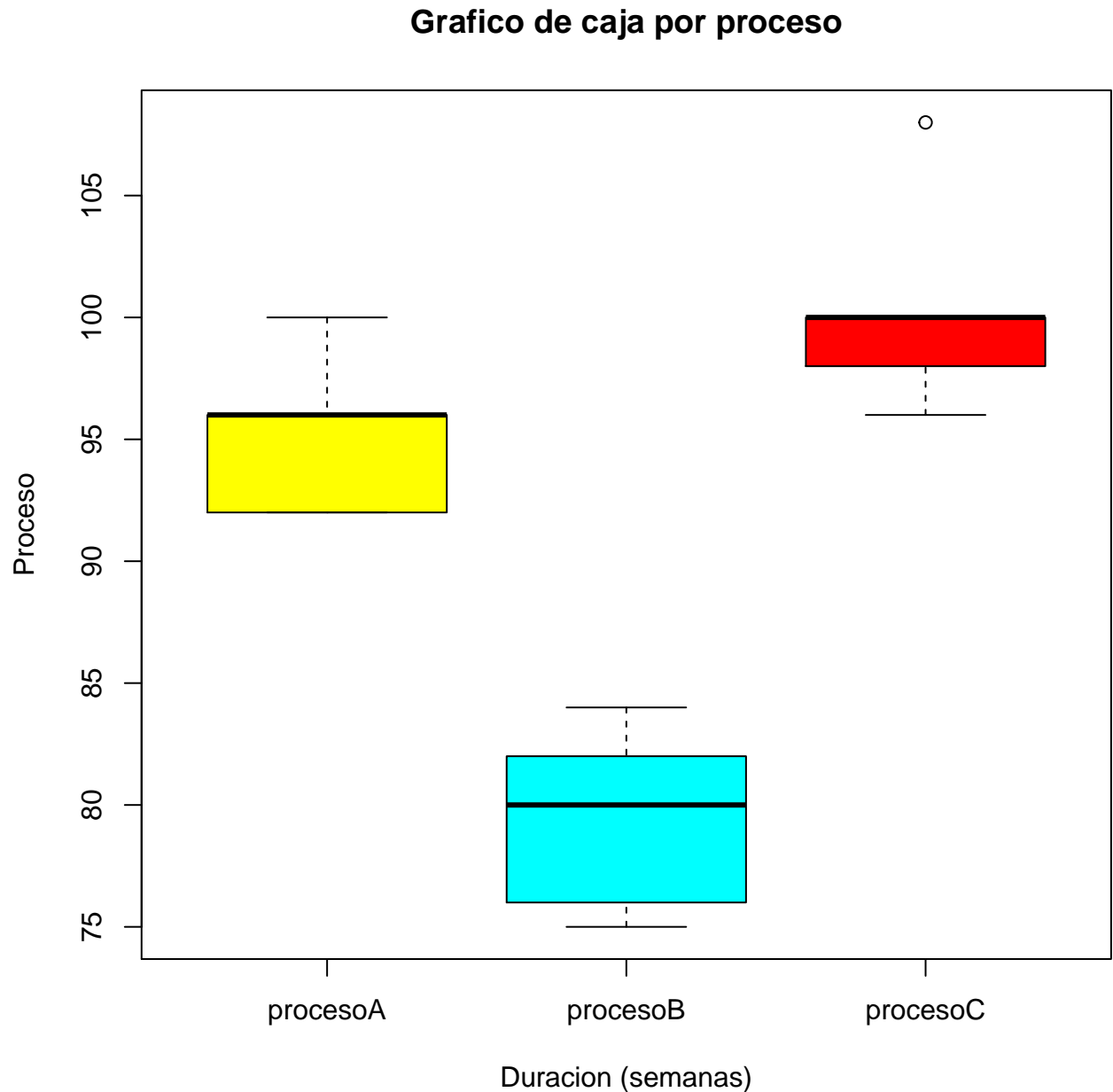
```
## 3rd Qu.: 96.0    3rd Qu.:82.0    3rd Qu.:100.0
## Max.      :100.0    Max.      :84.0    Max.      :108.0
```

11) Dibuja un gráfico de cajas (box-plot) para los tres procesos.

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



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



12) Presenta la matriz de covarianzas muestral.

```
options(digits=3) # s\ 'olo 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
```

13) Presenta la desviación estándar de cada proceso.

```
desv <- sd(procesoA); desv
## [1] 3.35

desv <- sd(procesoB); desv
## [1] 3.85

desv <- sd(procesoC); desv
## [1] 4.56
```

14) Realiza un análisis de varianza de una vía, para probar la hipótesis nula de que el proceso no influye en la duración de las baterías, es decir, que no hay diferencias entre los tres procesos.

```
# Concatena los tres vectores dentro de un vector simple, junto con un vector
# factor indicador de la categor\ 'ia o tratamiento (A, B, C) que origina cada
# observacion. El resultado es un data.frame que tiene como componentes los
# dos vectores anteriores.
```

```
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"
```

```
# Prueba de igualdad de medias por descomposicion de la varianza en dos
# fuentes de variacion: la variabilidad que hay entre los grupos (debida a
# la variable independiente o los tratamientos), y la variabilidad que
# existe dentro de cada grupo (variabilidad no explicada por los tratamientos).

aov.Baterias <- aov(values~ind, data=Baterias)
# values~ind relaciona los valores muestrales con los respectivos grupos

summary(aov.Baterias)

##              Df Sum Sq Mean Sq F value    Pr(>F)
## ind              2    1196      598    38.3 6.1e-06 ***
## Residuals      12     187        16
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Note que es necesario la instruccion anterior para poder visualizar la
# tabla ANOVA
```

Decisión: ya que $\alpha = 0.05 > p\text{-value}$ obtenido, entonces se rechaza H_0

```
# Prueba de igualdad de medias en un diseño de una vía (o unifactorial)
# asumiendo que las varianzas de los grupos son iguales
oneway.test(values~ind, data=Baterias, var.equal = TRUE)

##
## One-way analysis of means
##
## data:  values and ind
## F = 40, num df = 2, denom df = 10, p-value = 6e-06
```

15) Deshace la concatenación del vector de valores y el vector indicador de categoría

```
Baterias = unstack(Baterias);Baterias

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

16) Desconecta la hoja de datos de la segunda ruta o lista de búsqueda


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

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

Ejemplo 2:

Suponga que un estudiante hace una encuesta para evaluar si los estudiantes que fuman estudian menos que los que no fuman.

REALICE UN ESTUDIO ESTADÍSTICO DE LOS DATOS.

1) Activa tu directorio de trabajo.

```
getwd()

## [1] "C:/Users/User/Documents/TODAS_PRACTICAS"

setwd("C:/Users/User/Documents/TODAS_PRACTICAS")
```

2) Crea un nuevo script y llamarle "Script11-DatosBivariados3".

3) Crea dos vectores con los datos.

```
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
```

4) Crea una hoja de datos que tenga como componentes o columnas los dos vectores.

```
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

# Puedes editar los datos utilizando
fix(Estudia)
```

5) Guarda la hoja de datos en un archivo.

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

6) Elimina los objetos almacenados en el área de trabajo (Workspace)

```
ls()

## [1] "aov.Baterias" "Baterias"      "Cantidad"      "desv"
## [5] "Estudia"      "Fuma"          "S"

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

## character(0)
```

7) Recupera desde el archivo la hoja de datos.

```
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
```

8) Conecta la hoja de datos a la segunda ruta o lista de búsqueda

```
attach(Estudia, pos=2)
search()

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

9) Crea una tabla de contingencia o de doble entrada.

```
tablaCont <- table(Estudia)
tablaCont

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

10) Calcula las tablas de proporciones o de probabilidades.

```
options(digits=3) # solo imprime 3 lugares decimales

# Proporciones basadas en el total de la muestra, la suma de filas y
# columnas suman 1
propTotal <- prop.table(tablaCont); propTotal

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

# Proporciones basadas en el total por fila, cada fila suma 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

# Proporciones basadas en el total por columna, cada columna suma 1
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
```

11) Construya los gráficos de barras de la variable bidimensional.

```
# Grafico de barras apiladas con la frecuencia de Cantidad como altura

barplot(table(Estudia$Cantidad, Estudia$Fuma), beside = FALSE,
horizontal=FALSE, main="Grafico de barras (Fuma, Cantidad de
horas de estudio)", legend.text =T, xlab="Fuma", ylab="Cantidad de
horas-estudio",col=c("yellow", "white", "cyan"))

## 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
```

Grafico de barras (Fuma, Cantidad de horas de estudio)

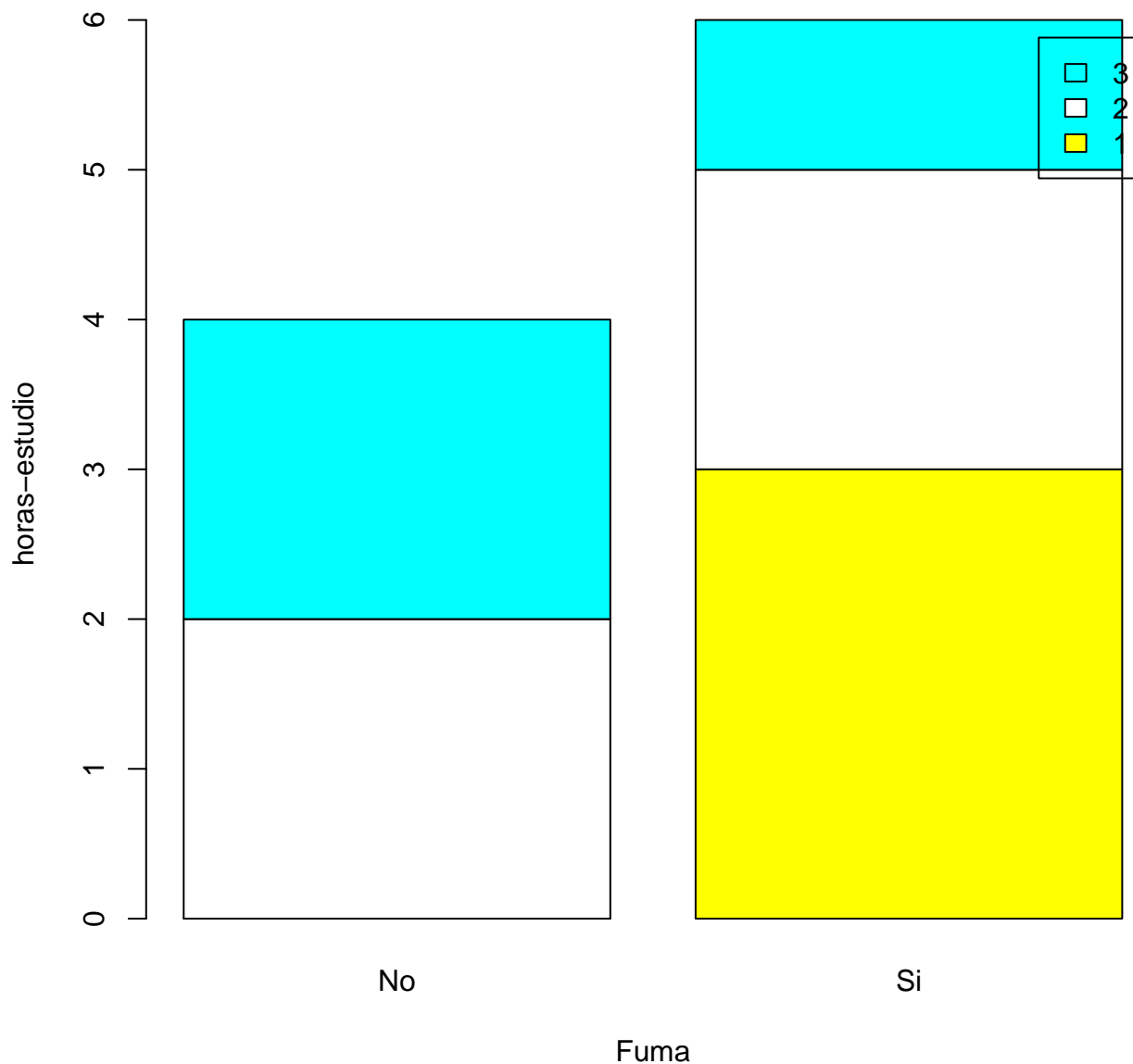


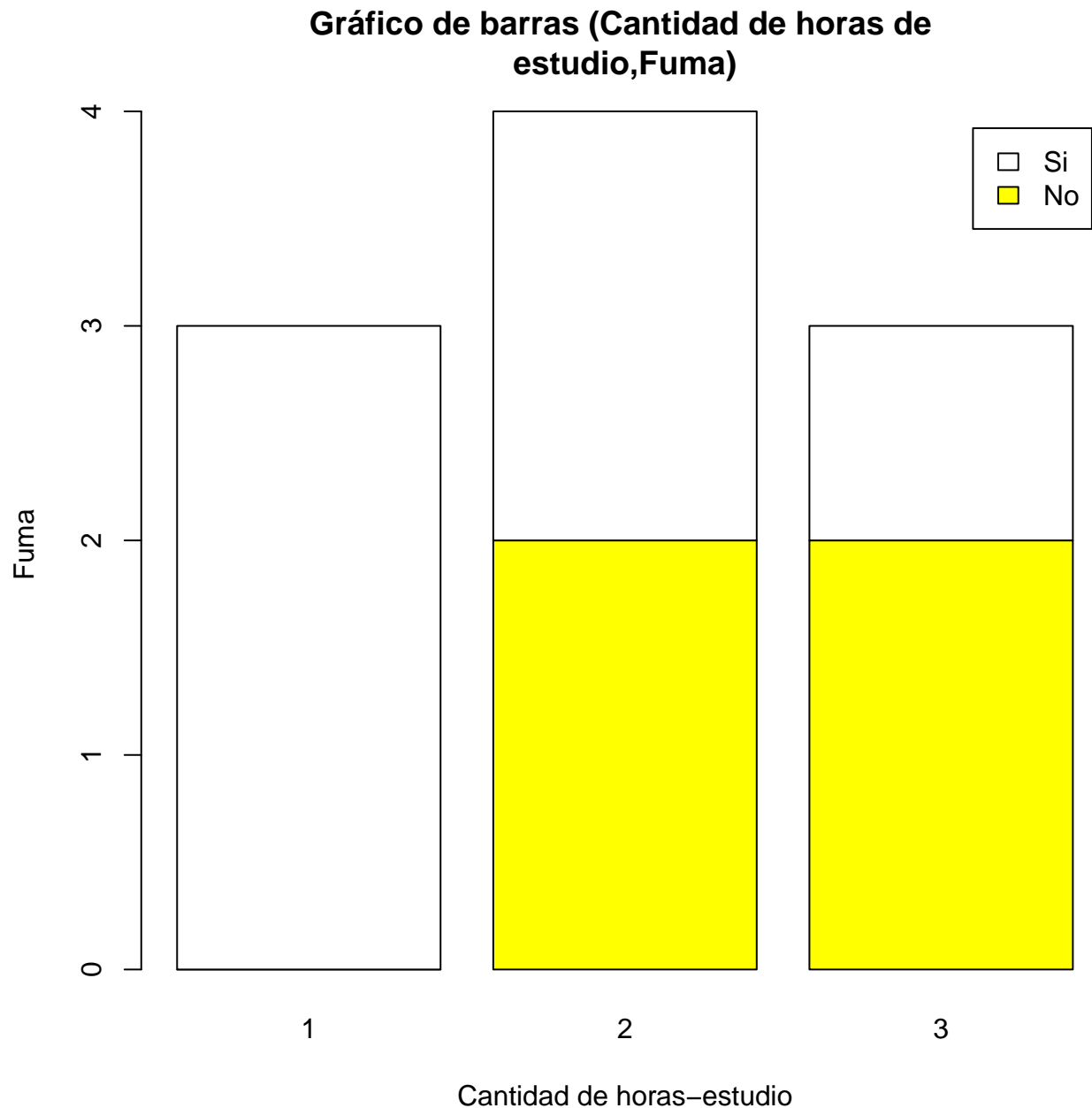
Grafico de barras apiladas con la frecuencia de Fuma como altura

```
barplot(table(Estudia$Fuma, Estudia$Cantidad), beside = FALSE,
horizontal=FALSE,main="Gráfico de barras (Cantidad de horas de
estudio,Fuma)", legend.text =T, xlab="Cantidad de horas-estudio",
ylab="Fuma",col=c("yellow", "white", "cyan"))
```

```
## Warning in plot.window(xlim, ylim, log = log, ...): "horizontal" is not a graphical
parameter
```

```
## Warning in axis(if (horiz) 2 else 1, at = at.1, 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
```



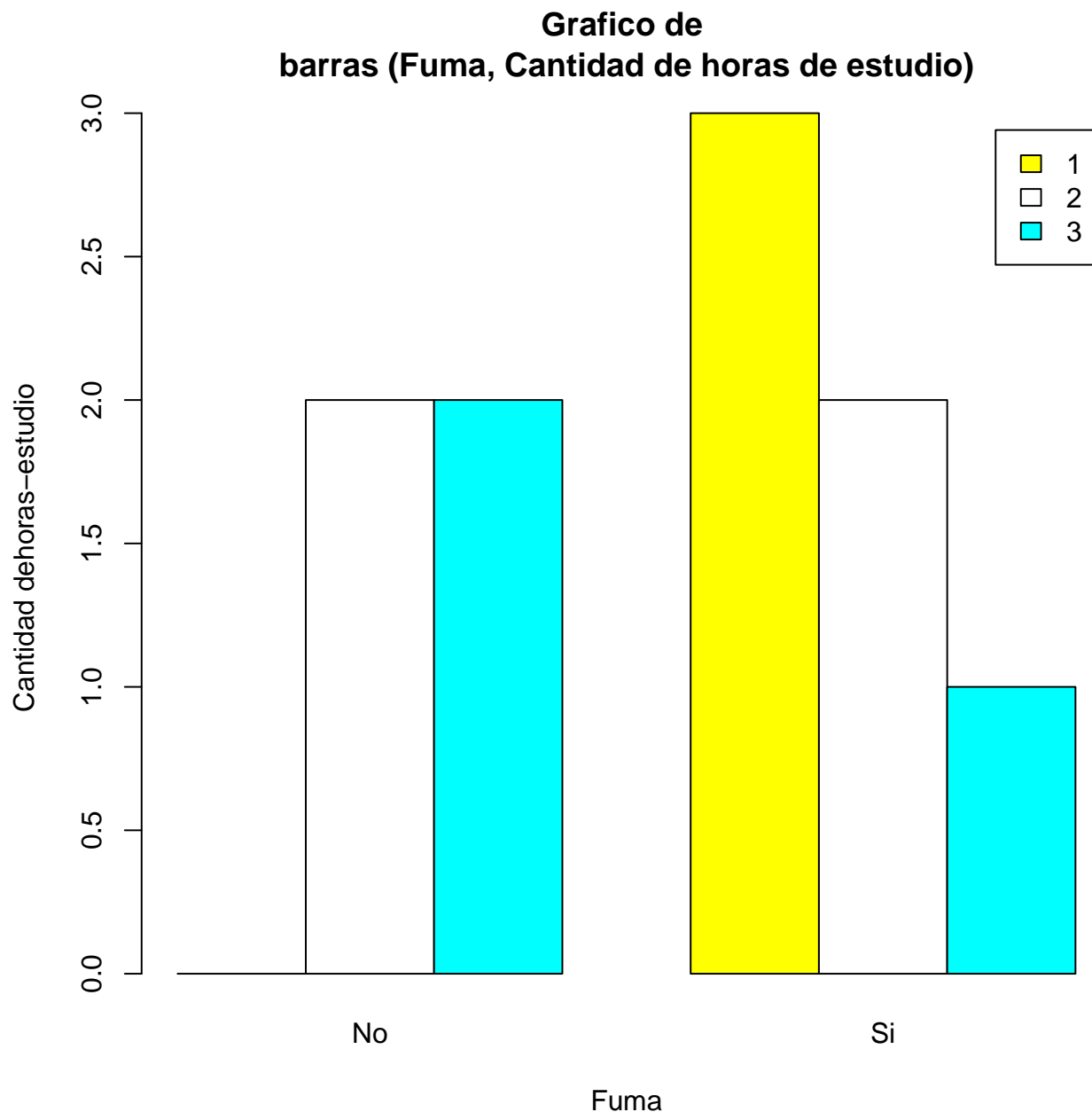
```
# Grafico de barras no apiladas y colocacion de leyenda

# Crear un factor para los nombres en la leyenda

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

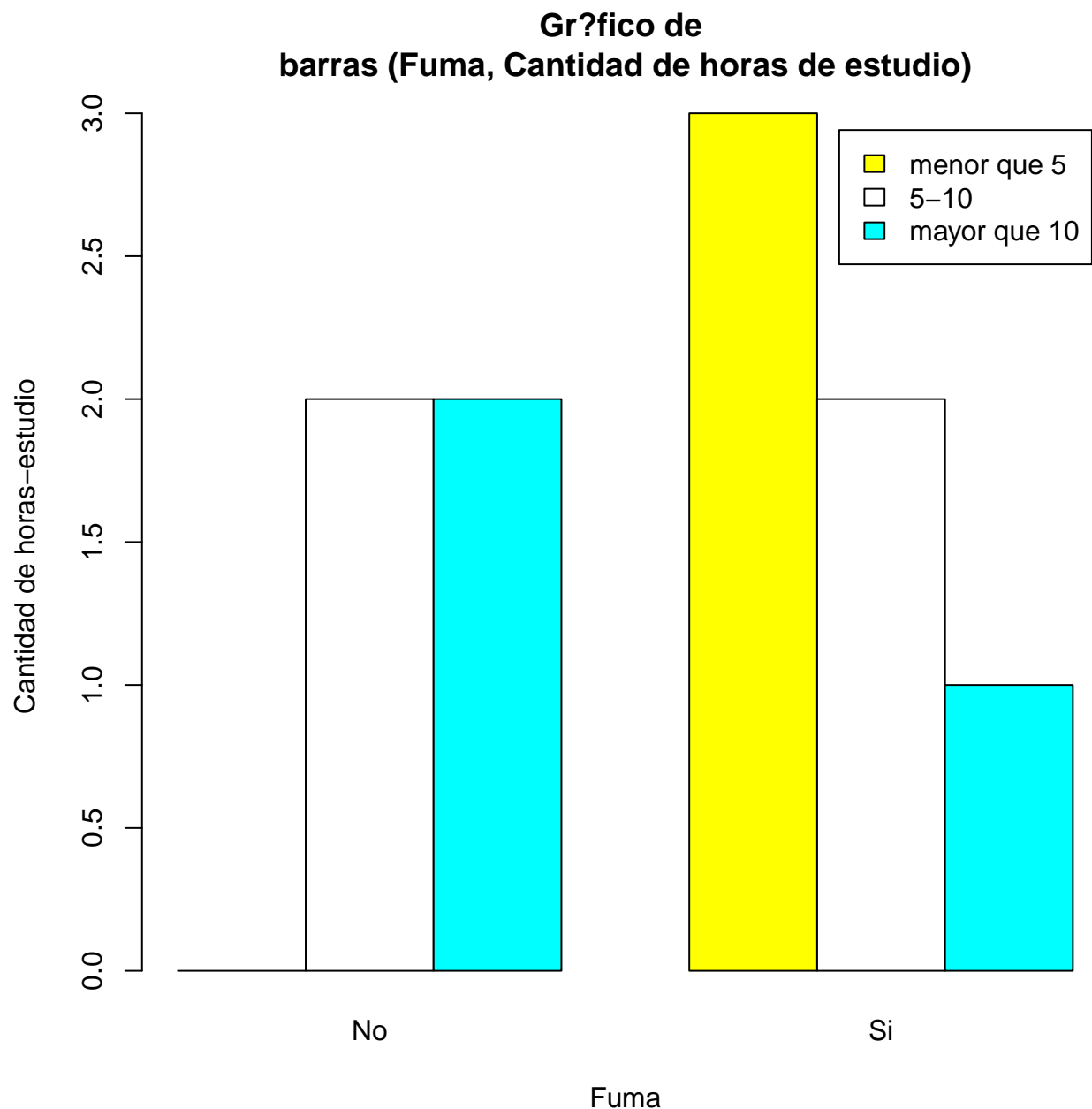
```
## [1] Si No No Si No Si Si Si No Si
## Levels: No Si
```

```
barplot(table(Estudia$Cantidad, Estudia$Fuma), main="Grafico de
barras (Fuma, Cantidad de horas de estudio)", xlab="Fuma",
ylab="Cantidad de horas-estudio", beside=TRUE,
col=c("yellow", "white", "cyan"), legend.text=T)
```



```
barplot(table(Estudia$Cantidad, Estudia$Fuma), main="Gr?fico de
barras (Fuma, Cantidad de horas de estudio)", xlab="Fuma",
ylab="Cantidad de horas-estudio", beside=TRUE, col=c("yellow",
```

```
"white", "cyan"), legend.text=c("menor que 5", "5-10", "mayor que 10"))
```



12) Realiza la prueba o contraste Chi-cuadrado para las probabilidades dadas
`chisq.test(tablaCont)`

```
# Si p-value > a aceptar Ho: Las variables son independientes
# Recuerde que las frecuencias esperadas deben ser mayores a 5 para poder
# utilizarlas.
# Probabilidades esperadas para la prueba Chi-cuadrada

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


##		Cantidad			
##	Fuma	1	2	3	
##	No	1.2	1.6	1.2	
##	Si	1.8	2.4	1.8	