GRÁFICOS Y DOCUMENTACIÓN EN

FUNCIONES EXPLORATORIAS

- `hist()`: visualización de distribución empírica de los datos
- `boxplot()`: representación visual de mínimo, mediana, máximo... diagrama de caja
- `plot()`: representación de nubes de puntos
- `barplot()`: visualización de datos categóricos

el operador "tilde" ~ representa dependencia izquierda derecha y se usa en las funciones gráficas para observar la dependencia de una variable frente a otra:

```
> plot(y ~ x)
```

EJERCICIO

datos de ejemplo:

```
> notas <- read.table('http://verso.mat.uam.es/~joser.berrendero/datos/notas.txt', sep
```

1. representar histograma de las notas de 2009
2. comparar notas de 2009 con notas de 2010 con un gráfico de caja
3. comparar notas de 2010 para cada tipo de colegio con un gráfico de caja
4. estudiar si hay relación entre notas de 2009 y 2010 con una nube de puntos
5. usar `table()` y un gráfico de barras para representar cuántos alumnos hay por tipo de centro

usar el operador ~ para 3 y 4

BIBLIOTECA LATTICE

```
> library("lattice")
```

los gráficos se crean con una única llamada (ej: `xyplot`,`bwplot`)

útiles para visualizar datos multidimensionales

```
> plot(nota09 ~ nota10 | tipo, data = notas)
```

BIBLIOTECA GGPLOT2

```
> install.packages("ggplot2")
> library("ggplot2")
```

creado por Hadley Wickham: más intuitivo y fácil de usar que `lattice`

plot con tendencia:

```
> qplot(nota09, nota10, data = notas, geom = c("point", "smooth"))
```

histograma:

```
> qplot(nota09, fill=tipo, data = notas)
```

diagrama de cajas:

```
> qplot(tipo, nota10, data = notas, geom = "boxplot")
```

BIBLIOTECA GGPLOT2: FACETAS

dividen el gráfico en paneles

```
> qplot(nota09, data = notas, facets = tipo ~ . )
```

- la variable a la izquierda de ~ indica cómo se dividen las filas
- la variable a la derecha de ~ indica cómo se dividen las columnas (**.** significa que ninguna)

PROGRAMACIÓN ESTADÍSTICA DOCUMENTADA

pros:

- texto y código están en un único lugar y con el orden lógico dictado por el flujo del análisis
- los resultados se actualizan automáticamente para reflejar cambios en datos, código, etc.
- el código está vivo

contras:

- a veces texto y código en un único lugar hacen que sea difícil de leer
- el procesado de documentos puede ser lento si el documento es muy largo

DOCUMENTACIÓN EN R: MARKDOWN

Es una herramienta de conversión de texto a HTML para escritores en la web. Permite escribir formato en texto plano fácil de leer y escribir con el que se puede crear una página web.

John Gruber

DOCUMENTACIÓN EN R: MARKDOWN

los signos de puntuación significan lo que parecen

Ejemplo: **esta presentación**

para generar un PDF maquetado con latex a partir de markdown:

pandoc

DOCUMENTACIÓN EN R: SINTAXIS MARKDOWN

cabeceras

```
# cabecera de primer nivel
## cabecera de segundo nivel
### cabecera de tercer nivel
```

formatos *cursiva* y **negrita** listas no ordenadas:

```
- elemento 1
- elemento 2
- elemento 3
```

listas ordenadas:

```
1. elemento 1
2. elemento 2
3. elemento 3
```

enlaces: `[Título del enlace](http://www.icane.es)`

KNITR

paquete diseñado por Yihui Xie para generar informes en R de forma elegante, dinámica, flexible y rápida

buena idea para:

- manuales y tutoriales
- documentos técnicos de extensión corta o media
- informes periódicos

no tan buena idea para:

- artículos de investigación muy largos
- documentación de cálculos muy complejos
- documentos que requieren formatos específicos complicados

KNITR

```
> install.packages("knitr")
> library("knitr")
```

fragmentos o bloques de código en R:

```
```{r} # comienzo de bloque de código en R
``` # fin de bloque de código en R
```

ocultar resultados:

```
```{r results="hide"}
```

hacer tablas:

```
library(xtable)
xt <- xtable(summary(cars))
print(xt, type = "html")
```
```

Acceso al repositorio con la presentación

Acceso a la presentación

Fco. Javier Parra Rodríguez,Jefe de Servicio de Estadísticas Económicas y Socio-demográficas