

# R / Bioconductor: Curso Intensivo

List

Factor

Apply

Datos  
Univariados

Categoricos

Discretos

Continuos

Gráficas

PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

Leonardo Collado Torres

Licenciatura en Ciencias Genómicas, UNAM

[www.lcg.unam.mx/~lcollado/index.php](http://www.lcg.unam.mx/~lcollado/index.php)

Cuernavaca, México

Oct-Nov, 2008

# Datos Univariados y Gráficas

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Categoricos

Discretos

Contínuos

Gráficas

PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

- 1 List
- 2 Factor
- 3 Apply
- 4 Datos Univariados
- 5 Categóricos
- 6 Discretos
- 7 Contínuos
- 8 Gráficas
- 9 PNG, PDF
- 10 Gráficas de Barras
- 11 Histogramas
- 12 Aleatorios
- 13 Dotchart
- 14 Resumiendo

# Que son

- R ofrece diferentes objetos como son los vectores atómicos<sup>1</sup>, matrices y data frames. Otro de estos son los `list`.
- Las `list` en realidad consisten de una colección de objetos conocidos como sus componentes. Estos pueden ser de cualquier tipo como ven aquí:

```
> lista <- list(nombre = "Leo", hermano = "Alex",  
+             edad = 21, calif.alumnos = c(6,  
+             9, 10, 8, 7))
```

```
> lista$nombre == lista[[1]]
```

```
[1] TRUE
```

```
> lista$calif.alumnos[1] == lista[[4]][1]
```

```
[1] TRUE
```

# Que son

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Categoricos

Discretos

Continuos

Gráficas

PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

```
> var <- "hermano"  
> lista[["hermano"]] == lista[[var]]  
  
[1] TRUE
```

---

<sup>1</sup>Donde todos los elementos son del mismo tipo

# Accesando una lista

- Como se habrán dado cuenta, hay diferentes formas de acceder a una lista. En general, puedes acceder a cada elemento usando `lst[[i]]` donde `i` va desde 1 hasta `length(lst)`.
  - ▶ `$` es útil por si no se acuerdan de que posición corresponde al elemento que quieren recuperar.
  - ▶ `lista[[var]]` es bastante útil si el nombre del elemento que quieren acceder está en una variable.
  - ▶ Si el elemento de la lista es un vector, pueden acceder a las diferentes posiciones como en el ejemplo de `lista[[4]][1]`.
- Es muy importante que noten la diferencia entre `lista[1]` y `lista[[1]]`. El primero te regresa una *sublista* mientras que el segundo te regresa el primer elemento de la lista.

# Crear una lista

- Crear una lista es bastante sencillo como ya vieron. Es recomendable que especifiques los nombres de cada elemento aunque no es obligatorio.<sup>2</sup>

```
> lista <- list(nom.1 = ele.1, ...,  
+             nom.n = ele.n)
```

- Una vez creada una lista, pueden añadirle elementos así:

```
> lista[n + 1] <- list(nom.m = ele.m)
```

- Pueden contatenar listas usando `c()`:

```
> lista.ABC <- c(lista.A, lista.B,  
+               lista.C)
```

- Finalmente, pueden borrar elementos de la lista usando `<- NULL`

---

<sup>2</sup>Una lista puede tener funciones como elementos!

# Que son

- Otro tipo de objeto en R son los `factor`. Estos los pueden ver como vectores que tienen alguna información con respecto a la clasificación de los datos.
- En sí son como enumeraciones en otros lenguajes y son útiles para generar datos tabulares.
- Cuando usan la función `read.table`, todo lo que parece un caracter es leído como un `factor`
- Luego lean más sobre la función `cut` para aprender a generar datos tabulares.
- **Un factor no es de tipo numérico!** Por ejemplo, no pueden usar la función `mean`.

# Un ejemplo

- Aquí les muestro un ejemplo donde uso un factor `^^`:

```
> fiesta <- factor(sample(c("muerto",  
+      "happy", "pedo", "sobrio"),  
+      100, replace = TRUE, prob = c(0.1,  
+      0.4, 0.3, 0.2)))  
> fiesta[1:4]
```

```
[1] happy  sobrio pedo  pedo  
Levels: happy muerto pedo sobrio
```

```
> table(fiesta)
```

```
fiesta  
happy muerto  pedo sobrio  
    47      8    31    14
```



# Un ejemplo

## Substituciones

Perl es excelente para manejar strings, pero R también puede hacer sustituciones con la función `sub`. Por ejemplo:

```
> fiesta2 <- sub("o$", "os", as.character(fiesta),  
+               perl = TRUE)  
> fiesta2[1:10]
```

```
[1] "happy"    "sobrios"  "pedos"  
[4] "pedos"    "pedos"    "happy"  
[7] "happy"    "happy"    "happy"  
[10] "sobrios"
```

# La familia apply

- R ofrece una familia de funciones que les será muy útil ya que les ahorra líneas de código. Además, tengo entendido que son más eficientes.
- La función básica de esta familia es **apply** y la completan **sapply**, **tapply** y **lapply**.
- La sintaxis de cada una varia, pero la del **apply** es: `apply(X, MARGIN, FUN, ARGs)`.
  - ▶ X es un arreglo, matriz o data.frame
  - ▶ MARGIN delimita si trabajamos sobre las filas (1), columnas (2) o ambas `c(1,2)`
  - ▶ FUN es una o más funciones, por ejemplo `mean`
  - ▶ ARGs es para los argumentos de la función, si es que son necesarios.

# For vs apply

- Vamos a ver como la familia apply nos facilita las cosas. Retomen el último ejercicio de la anterior sesión. Este lo pudieron resolver así usando el ciclo for:

```
> arch <- "http://kabah.lcg.unam.mx/~lcollado/R/d
> topfagos <- read.table(file = arch,
+       header = T)
> res <- NULL
> for (i in 1:length(topfagos[1,
+       ])) {
+       temp <- sum(topfagos[, i])
+       res <- c(res, temp)
+ }
> res <- res/sum(topfagos)
```

# For vs apply

```
> names(res) <- colnames(topfagos)
> res[res > 0.03]
```

N_AAT	D_GAT	E_GAA
0.03727998	0.04731938	0.04747806
G_GGT	I_ATT	K_AAA
0.03545449	0.03492148	0.04400270

- Ahora vamos a hacerlo con sapply:

```
> res2 <- sapply(topfagos, sum)/sum(topfagos)
> res2[res2 > 0.03]
```

N_AAT	D_GAT	E_GAA
0.03727998	0.04731938	0.04747806
G_GGT	I_ATT	K_AAA
0.03545449	0.03492148	0.04400270

# Tipos de Datos

- Como un super repaso de la estadística descriptiva, acuérdense que hay varios tipos de datos. Estos se analizan y visualizan de diferentes formas.
  - ▶ Datos categorícos: simplemente son datos divididos por categorías. Pueden ser nominales u ordinales.
  - ▶ Númericos discretos: generalmente usan los números enteros y se pueden hacer cierto tipo de operaciones con ellos. Por ejemplo, la edad.
  - ▶ Númericos contínuos: su rango generalmente está en los reales. Estos los puedes volver discretos al truncarlos o categorícos al usar intervalos.
- Si quieren revisar más sobre la estadística descriptiva chequen este pdf. A algunos se les hará familiar :P.

- Que haces cuando tienes un set de datos? Pues quieres analizarlos y explorarlos :P. R es muy fuerte en el *exploratory data analysis*.
- Las gráficas son una parte esencial del EDA. En las siguientes diapositivas veremos que tipo de gráficas son sirve para cierto tipo de datos. Más adelante, ya haremos unas nosotros.
- Por ahora vamos a trabajar con datos univariados. Esto significa que son de una sola variable.

# Datos Categóricos

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Categoricos

Discretos

Continuos

Gráficas

PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

- La forma más simple de resumir este tipo de datos es hacer una tabla. Por ejemplo:

	happy	muerto	pedo	sobrio
1	47	8	31	14

Table 1: Estado de la fiesta

- Aquí usé la función `xtable` del paquete con el mismo nombre. En R básico pueden usar la función `table`<sup>3</sup>.

<sup>3</sup>Lo que le pases tiene que poder ser interpretado como un `factor`

# Categoricos: **barplot**

- Otra forma de resumir este tipo de datos es con gráficas de barras.
- A esta función le pueden pasar como argumento el resultado de la función `table`. Esto les puede ayudar a resumir sus datos.
- Tengan cuidado al usar este tipo de gráficas, ya que siempre te pueden intentar engañar si cambian el rango del eje Y. Es por eso que deberían empezar en 0.



# Barplot Incorrecto

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Categoricos

Discretos

Continuos

Gráficas

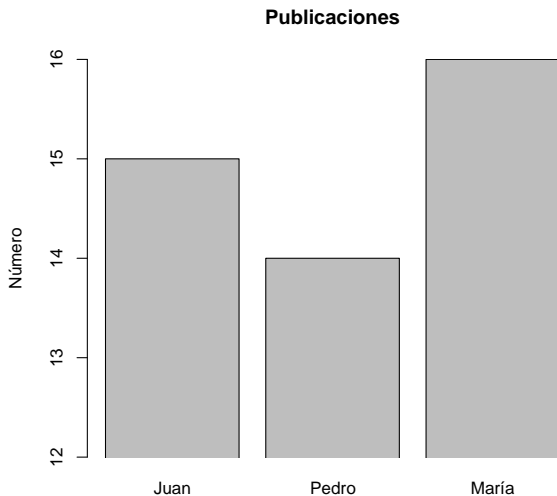
PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart



# Barplot Correcto

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Categoricos

Discretos

Continuos

Gráficas

PNG, PDF

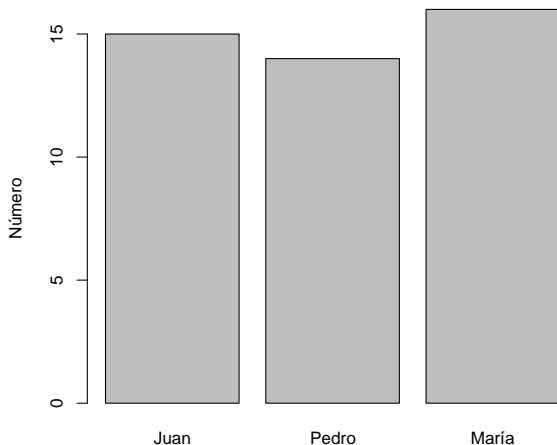
Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

**Publicaciones**



- Las gráficas de barra también las pueden usar para graficar líneas de tiempo. Aquí deben tener cuidado de que sea entendible la gráfica; aka, intenten minimizar el uso de "tinta".
- Otro tipo de gráfica para datos categóricos es la de pie. Sin embargo, este tipo de gráfica es altamente NO recomendable. Prueba de esto, lean lo que dice ?pie:
  - ▶ Pie charts are a very bad way of displaying information. The eye is good at judging linear measures and bad at judging relative areas. A bar chart or dot chart is a preferable way of displaying this type of data.

# Cual es mas grande?

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Categoricos

Discretos

Contínuos

Gráficas

PNG, PDF

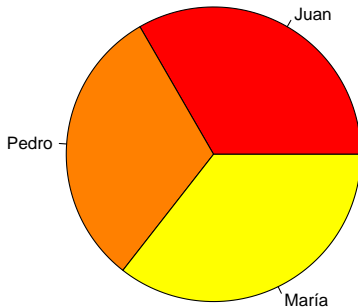
Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

**Publicaciones**



# Dotchart

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catégoricos

Discretos

Contínuos

Gráficas

PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

- El tercer tipo de gráfica que nos ayuda con este tipo de datos se llama **dotchart**.
- Esta función básicamente nos regresa los valores dentro del rango de la variable<sup>4</sup>. Aquí deben tener cuidado con el eje X para ver realmente los valores.
- Usen este tipo de gráficas para resaltar las diferencias.
- Pueden leer más al respecto [aquí](#).

---

<sup>4</sup>Osea, del min al max

# Una dotchart

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Categoricos

Discretos

Contínuos

Gráficas

PNG, PDF

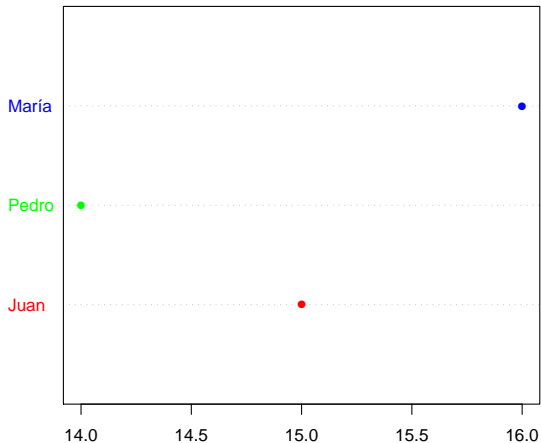
Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

**Publicaciones**



# Números Discretos

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catagóricos

Discretos

Contínuos

Gráficas

PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

- Cuando su variable tome valores numéricos, van a querer entender como se distribuyen. Querrán conocer el rango de sus datos, por donde están centrados, que tan esparcidos están, etc.
- Hay funciones, como la de `summary()` que les pueden ayudar. La otra opción es checar los datos gráficamente ^^.

# Tallo y hojas

- La más básica de estas gráficas es la de tallo y hojas<sup>5</sup> si es que su conjunto de datos no es muy grande. Esta se hace con la función **stem**.
- Supongo que ya han visto este tipo de gráfica antes, pero les recuerdo que pone los decimales como tallos y los dígitos como hojas<sup>6</sup>. En sí, es como un histograma acostado.
- Rápidamente pueden ver la forma de la distribución y el rango. Aquí vemos un ejemplo con los datos de `islands`:

```
> stem(islands)
```



## Tallo y hojas

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

## Datos

Univariados

Discretos

## Gráficas

### Dotchart

The decimal point is 3 digit(s) to the right of the

[illegible]<sup>5</sup>stem-and-leaf

<sup>6</sup>si sus números van de 0 a 99

# Alternativas al stem

- Como ya se dieron cuenta, code ocupa bastante espacio vertical, por lo que existen alternativas.
- Estas muestran prácticamente la misma información, aunque hacia arriba.
- La primera es la función **stripchart**. Para salgan bien las gráficas hay que meterse en más detalle a sus argumentos.
- La segunda es la función **DOTplot**. Su uso es más simple y aquí les muestro un par de ejmplos con los datos de `kid.weights` del paquete **UsingR**.

# Stripchart

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catagóricos

**Discretos**

Contínuos

Gráficas

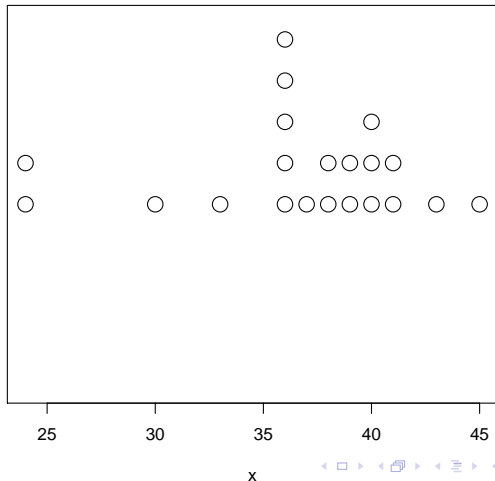
PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart



# Ahora con DOTplot

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catagóricos

**Discretos**

Contínuos

Gráficas

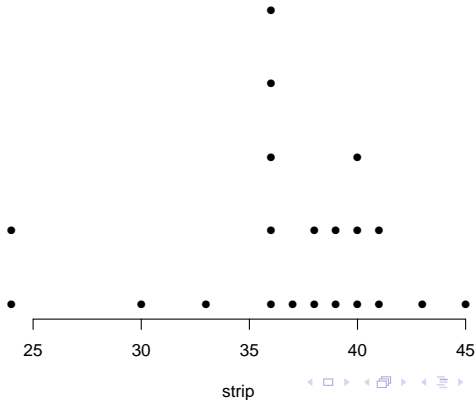
PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart



# Datos continuos

- En sí, las gráficas que veremos a continuación también sirven para datos discretos. Pero su utilidad es resaltada con datos continuos :D.

# Medidas de tendencia central

## Un repaso

Cuando tenga muchos datos van a querer encontrar información sobre un punto medio de sus datos. Con R pueden encontrar unos con funciones:

- La media muestral con **mean** y una media sin cierto porcentaje de valores extremos usando el argumento **trim**
- La mediana muestral con **median**
- La moda<sup>7</sup> aunque no hay una función que te lo haga. Puedes usar `which.max(table(x))`
- El punto medio del rango<sup>8</sup> con `mean(range(x))`

---

<sup>7</sup>No es muy buena para resumir un set de datos

<sup>8</sup>Es más sensible que la media a valores extremos

# Medidas de dispersión

Una vez que conozcan un punto medio, van a querer encontrar que tan dispersos están sus datos. Pueden encontrar:

- El rango (mín, máx) de sus datos con **range** y la diferencia entre ellos con `diff(range(x))`.
- La varianza muestral con **var** y la desviación estándar muestral con **sd**.
- El rango intercuartílico (IQR en inglés) que es la distancia entre los percentiles 25 y 75 con **IQR**.
- Pueden encontrar cualquier cuantil con la función **quantile**. Acuérdense de que la media es el cuantil 0.25.
- Recuerden que una función muy importante es **summary**.

# Forma de la distribución

- Las gráficas tipo tallas y hojas sirven siempre y cuando no tengan tantos datos. Así que la opción es hacer histogramas con la función `hist`.
- Un argumento muy importante de esta función es el `breaks` que decide cuantos grupos van a tener. Pueden usar el *default*, especificar un número o usar alguno de los algoritmos que ofrece la función.
- Con `hist` pueden generar histogramas idénticos para un set de datos. Uno puede tener frecuencias absolutos en el eje Y, mientras que otro puede ser de probabilidades. Para generar el segundo usen el parámetro `prob=T`.
- Aquí les muestro un par de ejemplos<sup>9</sup>:

---

<sup>9</sup>Noten que en el primero los datos tienen dist. bimodal



# Hist del faithful\$waiting

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catagóricos

Discretos

Contínuos

Gráficas

PNG, PDF

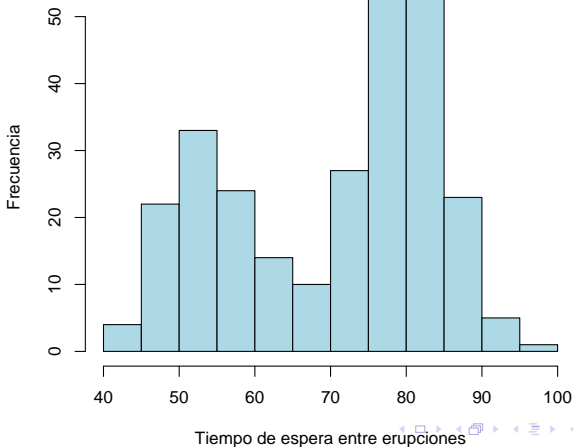
Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

Histograma de faithful\$waiting



# Densidad OBP

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catagóricos

Discretos

Contínuos

Gráficas

PNG, PDF

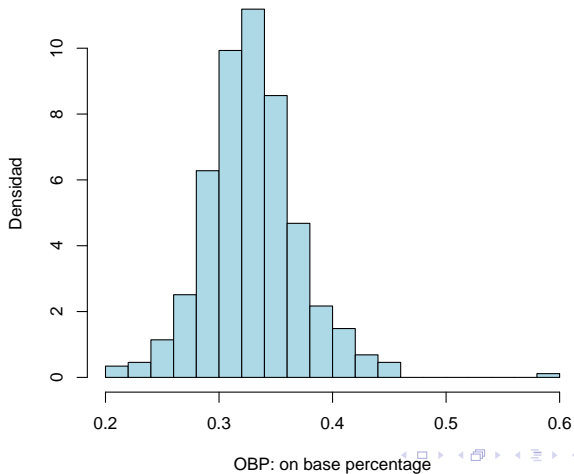
Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

Histograma de OBP



# Simetría, modas, sesgo

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catégoricos

Discretos

Contínuos

Gráficas

PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

- Muchas veces querrán ver si sus datos:
  - ▶ son simétricos
  - ▶ tienen una distribución unimodal, bimodal o multimodal
  - ▶ están sesgados y hacia que lado
- Una forma<sup>10</sup> es hacer un polígono de las frecuencias. Para este no hay una función que te lo haga.
- En fin, nos podemos dar una idea bastante clara de esto usando las funciones **lines** junto con **density**. Incluso los podemos juntar con un histograma.

---

<sup>10</sup>No se las recomiendo, pero a unos les gusta

# Polígono

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Categoricos

Discretos

Contínuos

Gráficas

PNG, PDF

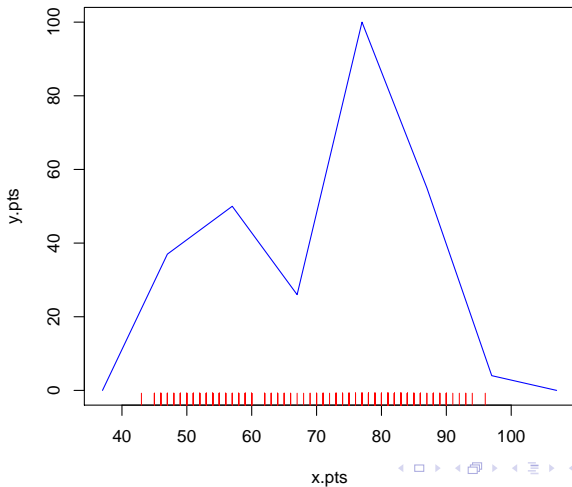
Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

Polígono de faithful\$waiting



# Bimodal

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catagóricos

Discretos

**Contínuos**

Gráficas

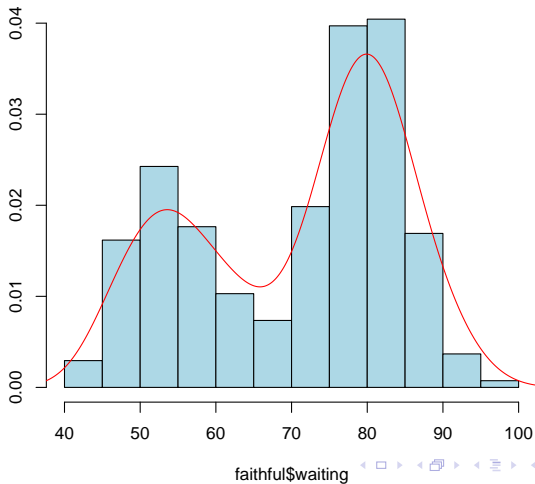
PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart



# Casi unimodal

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Categoricos

Discretos

**Contínuos**

Gráficas

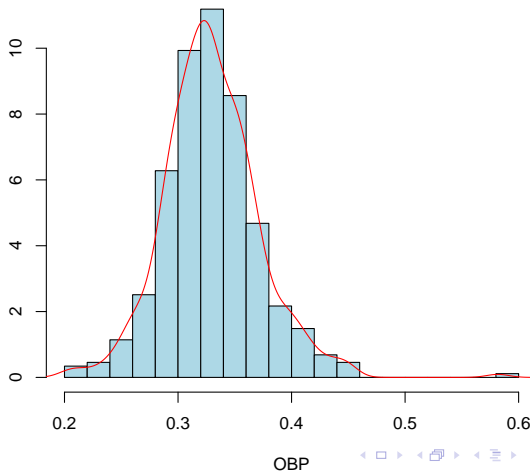
PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart



# Multimodal

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Categoricos

Discretos

Contínuos

Gráficas

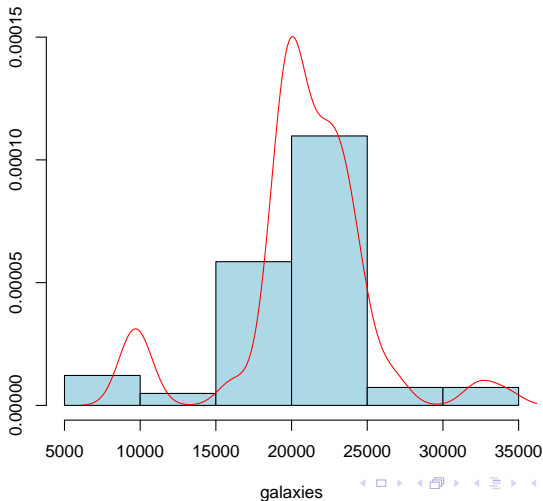
PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart



# Simétrico

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Categoricos

Discretos

Contínuos

Gráficas

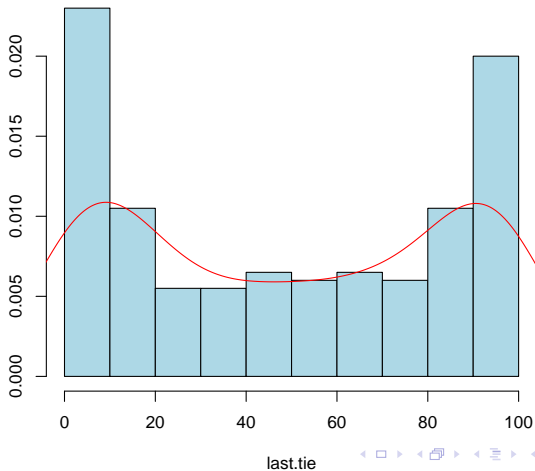
PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart





# Sesgado

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catagóricos

Discretos

**Contínuos**

Gráficas

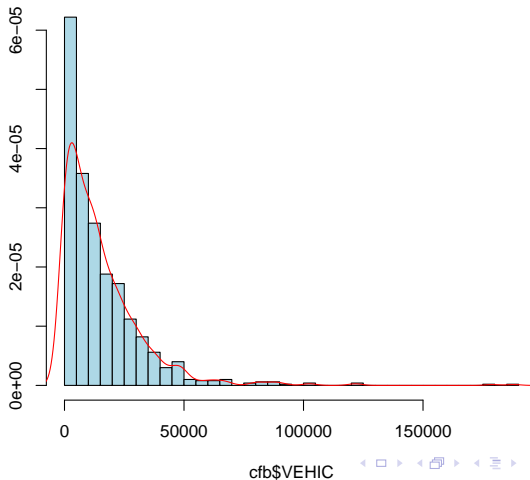
PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart



- Anoten que si sus datos son simétricos, la media, la mediana y la media truncada se van a parecer.
- La otra opción fuerte es usar el **boxplot**. Este en sí utiliza la información del `summary`. Con este tipo de gráfica es fácil visualizar:
  - ▶ Los cuartiles 1, 2 y 3. Recuerden que el segundo corresponde a la mediana.
  - ▶ Te pone un límites máximo y mínimo usando el cuartil 1 - 1.5 veces el IQR, el cuartil 3 + 1.5 veces IQR
  - ▶ Los valores extremos. Estos aparecen como bolitas hasta arriba y/o abajo de los límites.
- A continuación les muestro dos ejemplos:

# Un boxplot

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catagóricos

Discretos

Contínuos

Gráficas

PNG, PDF

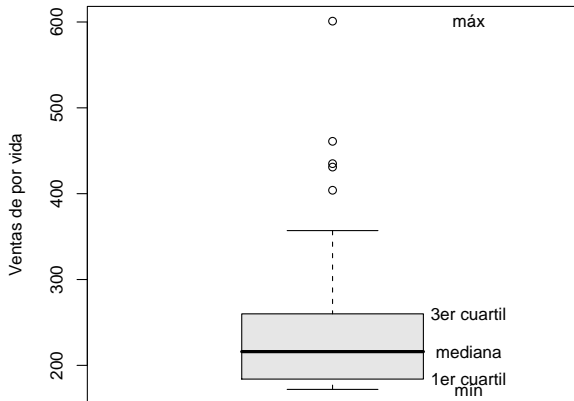
Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

## Usando alltime.movies\$Gross



# Otro boxplot

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catagóricos

Discretos

Contínuos

Gráficas

PNG, PDF

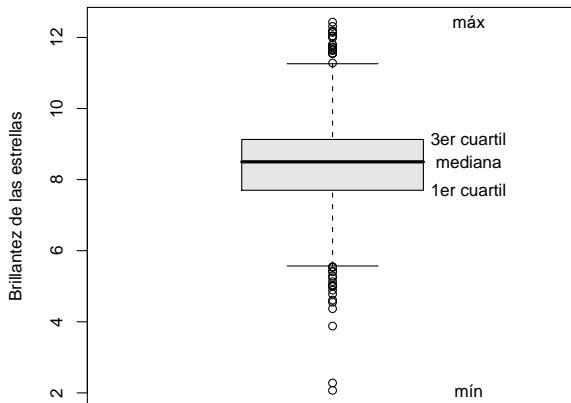
Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

## Usando brightness



# Aprendiendo a hacerlas

- En las siguientes diapositivas vamos a ver como hacer las gráficas simples en R.
- Vamos a aprender a usar varios parámetros de estas al ir mejorando ciertas gráficas.
- Necesitamos el objeto `topfagos` que ya habíamos definido previamente.

# Gráficas de líneas

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Categoricos

Discretos

Continuos

Gráficas

PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

- Primero hagan una gráfica básica con las sumas de las columnas de las primeras 5 líneas de `topfagos`:

```
> top5 <- apply(topfagos[1:5, ],  
+             2, sum)  
> plot(top5)
```

# Graf 1: paso 1

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catégoricos

Discretos

Contínuos

**Gráficas**

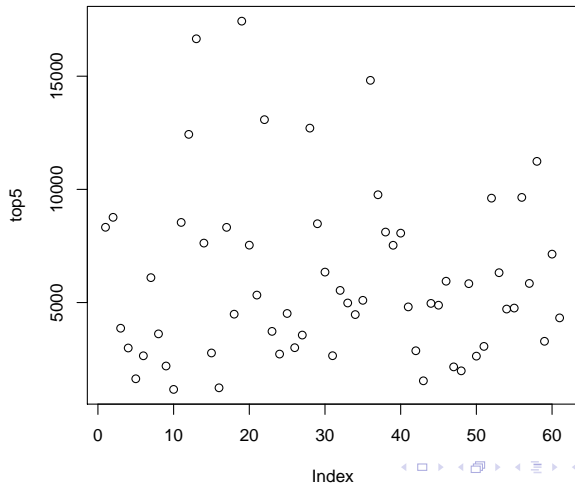
PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart



# type, col, main, title

- Ahora vamos a mejorar esa gráfica.

- Vamos a usar el argumento:

- ▶ type para cambiar el tipo de símbolo.
- ▶ col para cambiar los colores.
- ▶ title para ponerle título a la imagen.
- ▶ main para especificar el título.
- ▶ col.main para cambiar el color del título.
- ▶ font.main para poner el título en *italics*.

```
> plot(top5, type = "o", col = "blue")  
> title(main = "Top5 fagos y sus codones",  
+       col.main = "red", font.main = 4)
```



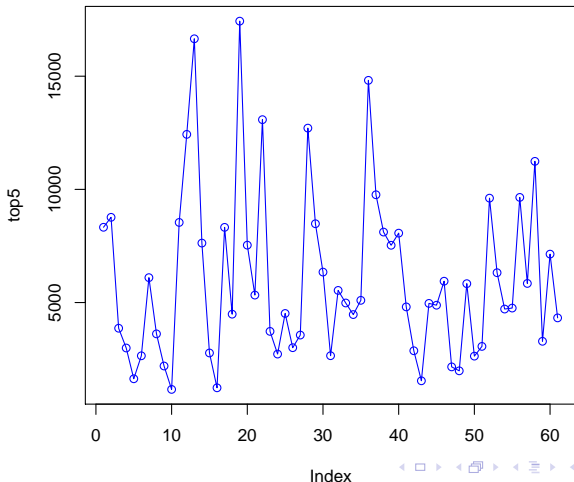
# Graf 1: paso 2

R /  
Bioconductor:  
Curso  
Intensivo

List  
Factor  
Apply  
Datos  
Univariados  
Categoricos  
Discretos  
Contínuos  
Gráficas  
PNG, PDF  
Gráficas de  
Barras  
Histogramas  
Aleatorios

Dotchart

## Top5 fagos y sus codones



- Ahora vamos a manejar 2 vectores de datos y vamos a usar los siguientes argumentos:

- ▶ ylim para delimitar los límites del eje Y.
- ▶ lines para añadir otra línea
- ▶ pch para escoger un símbolo para la segunda línea.
- ▶ lty para especificar el tipo de línea (punteada por ejemplo).

```
> top10 <- apply(topfagos[6:10, ],  
+               2, sum)  
> plot(top5, type = "o", col = "blue",  
+       ylim = c(1000, 10000))  
> lines(top10, typ = "o", pch = 22,  
+       lty = 2, col = "red")  
> title(main = "Top5 y 10 fagos",  
+       col.main = "red", font.main = 4)
```

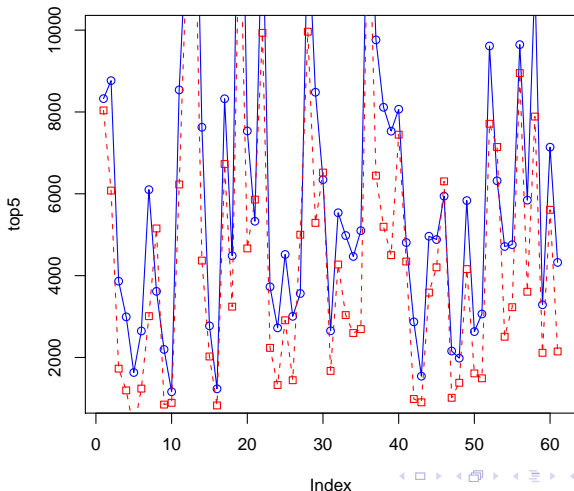
# Graf 1: paso 3

R /  
Bioconductor:  
Curso  
Intensivo

List  
Factor  
Apply  
Datos  
Univariados  
Categóricos  
Discretos  
Contínuos  
Gráficas  
PNG, PDF  
Gráficas de  
Barras  
Histogramas  
Aleatorios

Dotchart

## Top5 y 10 fagos



- Con los mismos 2 vectores, vamos a añadirle información al eje X y una leyenda a la gráfica. Usaremos:
  - ▶ `ann` y `axes` para apagar la información de los ejes que se crea automáticamente.
  - ▶ `axis` para especificar manualmente la información del eje X (1) y del eje Y (2).
  - ▶ `box` para hacer una caja y delimitar el espacio de la gráfica.
  - ▶ `title` para especificar los títulos de los ejes.
  - ▶ `legend` para crear la leyenda de la grafica.
  - ▶ `cex` y `cex.axis` para cambiar el tamaño de la letra relativo al tamaño default
  - ▶ `las` nos sirve para marcar donde queremos los valores del eje Y con respecto a este eje: horizontales, perpendiculares, etc.

## axes, ann, box

```
> rango <- range(0, top5, top10)
> plot(top5, type = "o", col = "blue",
+       ylim = rango, axes = FALSE,
+       ann = FALSE)
> axis(1, at = c(1, 5, 11, 13, 15,
+               17, 19, 21, 25, 27, 30, 36,
+               38, 39, 41, 45, 51, 55, 56,
+               58), lab = c("A", "R", "N",
+               "D", "C", "Q", "E", "G", "H",
+               "I", "L", "K", "M", "F", "P",
+               "S", "T", "W", "Y", "V"), font = 1,
+       cex.axis = 0.5)
> axis(2, las = 1, at = 3000 * 0:rango[2])
> box()
> lines(top10, type = "o", pch = 22,
```

# axes, ann, box

```
+      lty = 2, col = "red")
> title(main = "Top5 y 10 fagos",
+      col.main = "red", font.main = 4)
> title(xlab = "Codones", col.lab = rgb(0,
+      0.5, 0))
> title(ylab = "Frecuencia Absoluta",
+      col.lab = rgb(0, 0.5, 0))
> legend(40, rango[2], c("top5",
+      "top10"), cex = 0.8, col = c("blue",
+      "red"), pch = 21:22, lty = 1:2)
```

# Graf 1: paso 4

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catégoricos

Discretos

Contínuos

Gráficas

PNG, PDF

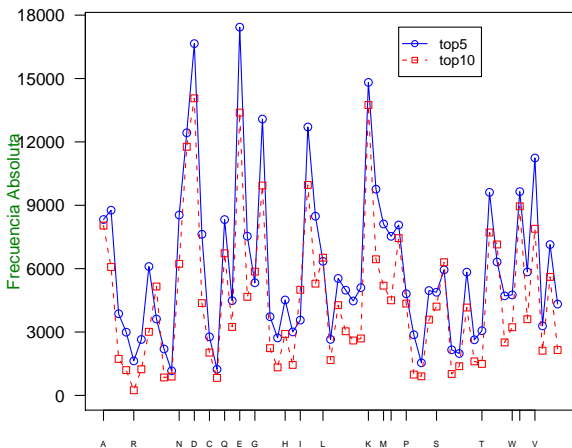
Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

## Top5 y 10 fagos



Codones

# Creando archivos png

- Ahora vamos a aprender a guardar la imagen en un archivo .png. Trabajaremos con 3 columnas específicas. Usaremos:

- ▶ `png` para especificar el nombre del archivo y que queremos que la gráfica se guarde en un png.
- ▶ `plot_colors` que es un objeto definido por nosotros para los colores que vamos a usar.
- ▶ `dev.off()` para especificar que terminas de hacer la gráfica. Al usar está línea se crea el archivo.

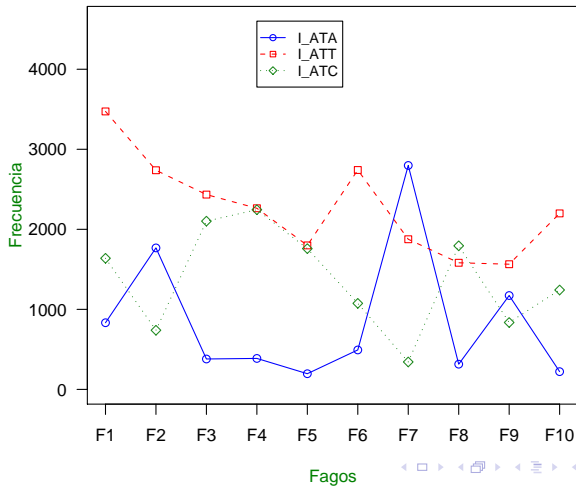


◀ ◻ ▶ ◀ ◻ ▶ ◀ ≡ ▶ ◀ ≡ ▶ ≡ ≡ ≡ ≡ ↺ 🔍 ↻

```
+      pch = 22, lty = 2, col = plot_colors[2])
> lines(topfagos$I_ATC, type = "o",
+      pch = 23, lty = 3, col = plot_colors[3])
> title(main = "Codones de I en top10 fagos",
+      col.main = "red", font.main = 4)
> title(xlab = "Fagos", col.lab = rgb(0,
+      0.5, 0))
> title(ylab = "Frecuencia", col.lab = rgb(0,
+      0.5, 0))
> legend(4, max_y, names(topfagos)[grep("^I",
+      names(topfagos))], cex = 0.8,
+      col = plot_colors, pch = 21:23,
+      lty = 1:3)
> dev.off()
```

## Graf 2: paso 1

### Codones de I en top10 fagos



# Creando pdfs

- Ahora vamos a guardar la imagen en formato pdf. Además vamos a usar:
  - ▶ **pdf** de igual forma que usamos png.
  - ▶ **xlab** y **ylab** para especificar los títulos de los ejes.
  - ▶ **text** para poner los nombres del eje X inclinados a 45 grados.
  - ▶ **lwd** para cambiar el grosor de la línea.
  - ▶ **mar** para cambiar los márgenes de la imagen.

# Creando pdfs

```
> plot_colors <- c(rgb(r = 0, g = 0,  
+      b = 0.9), "red", "forestgreen")  
> pdf(filename = "figure.pdf", height = 3.5,  
+      width = 5)  
> par(mar = c(4.2, 3.8, 0.2, 0.2))  
> plot(topfagos$I_ATA, type = "l",  
+      col = plot_colors[1], ylim = c(0,  
+      max_y), axes = F, ann = F,  
+      xlab = "Fagos", ylab = "Frecuencia",  
+      cex.lab = 0.8, lwd = 2)  
> axis(1, lab = F)  
> text(x = c(1:10), par("usr")[3] -  
+      200, srt = 45, adj = 1, labels = c("F1",  
+      "F2", "F3", "F4", "F5", "F6",  
+      "F7", "F8", "F9", "F10"), xpd = T,
```

# Creando pdfs

```
+      cex = 0.8)
> axis(2, las = 1, cex.axis = 0.8)
> box()
> lines(topfagos$I_ATT, type = "l",
+      lty = 2, col = plot_colors[2])
> lines(topfagos$I_ATC, type = "l",
+      lty = 3, col = plot_colors[3])
> legend("topleft", names(topfagos)[grep("^I",
+      names(topfagos))], cex = 0.8,
+      col = plot_colors, lty = 1:3,
+      lwd = 2, bty = "n")
> dev.off()
> par(mar = c(5, 4, 4, 2) + 0.1)
```

## Graf 2: paso 2

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catégoricos

Discretos

Contínuos

Gráficas

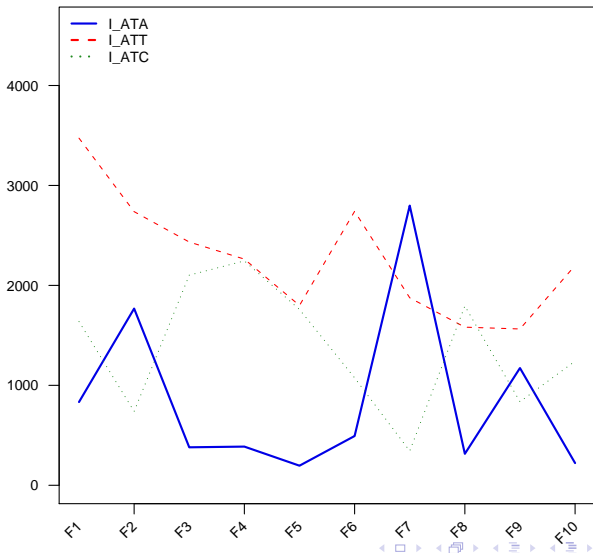
PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart



# Gráficas de barras

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catagóricos

Discretos

Continuos

Gráficas

PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

- Simplemente vamos a aprender a usar **barplot**



```
> barplot(topfagos$I_ATA)
```

# Graf 3: paso 1

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Categoricos

Discretos

Contínuos

Gráficas

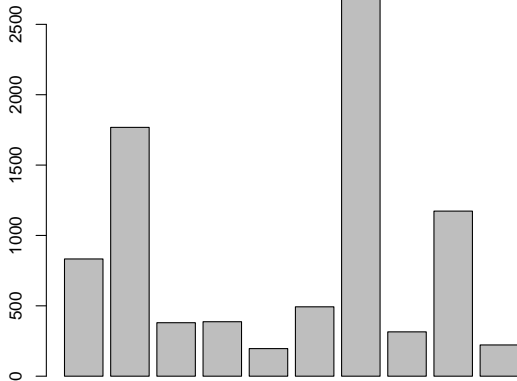
PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart



# Gráficas de barras

- Ahora la modificamos con argumentos que en su mayoría ya les son conocidos.

- ▶ `border` lo vamos a usar para especificar el color del borde de las barras.
- ▶ `density` lo vamos a usar para marcar que tan denso es el color adentro de cada barra.
- ▶ `names.arg` es para especificar los nombres de cada barra.

```
> barplot(topfagos$I_ATA, main = "Codon I_ATA",  
+         xlab = "Fagos", ylab = "Frecuencia",  
+         names.arg = c("F1", "F2", "F3",  
+                       "F4", "F5", "F6", "F7",  
+                       "F8", "F9", "F10"), border = "blue",  
+         density = seq(10, 100, by = 10))
```

## Graf 3: paso 2

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catagóricos

Discretos

Contínuos

Gráficas

PNG, PDF

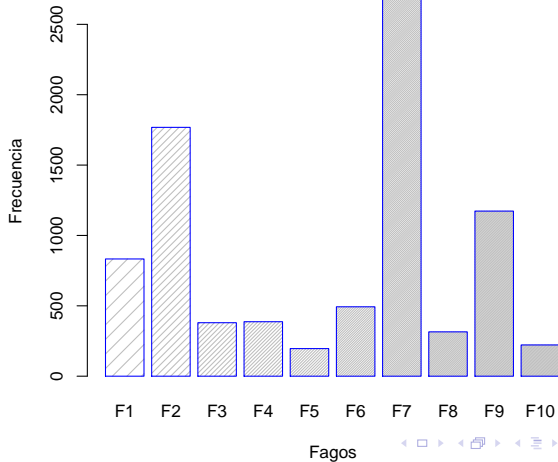
Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

Codon I\_ATA



# Algo de colores

- Ahora le vamos a agregar otros datos, una leyenda y unos colores.

- ▶ Usamos `rainbow` para obtener colores tipo arcoiris para nuestra gráfica.
- ▶ `beside` nos sirve para juntar las barras.

```
> codonesI <- matrix(c(topfagos$I_ATA,  
+ topfagos$I_ATT, topfagos$I_ATC),  
+ 3, 10)  
> rownames(codonesI) <- c("I_ATA",  
+ "I_ATT", "I_ATC")  
> barplot(t(codonesI), main = "Codones de I",  
+ xlab = "Fagos", ylab = "Frecuencia",  
+ beside = TRUE, col = rainbow(10))  
> legend(x = 9, y = 3500, c("F1",
```

# Algo de colores

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catagóricos

Discretos

Contínuos

Gráficas

PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

```
+      "F2", "F3", "F4", "F5", "F6",  
+      "F7", "F8", "F9", "F10"), cex = 0.8,  
+      bty = "n", fill = rainbow(10))
```

# Graf 3: paso 3

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catégoricos

Discretos

Contínuos

Gráficas

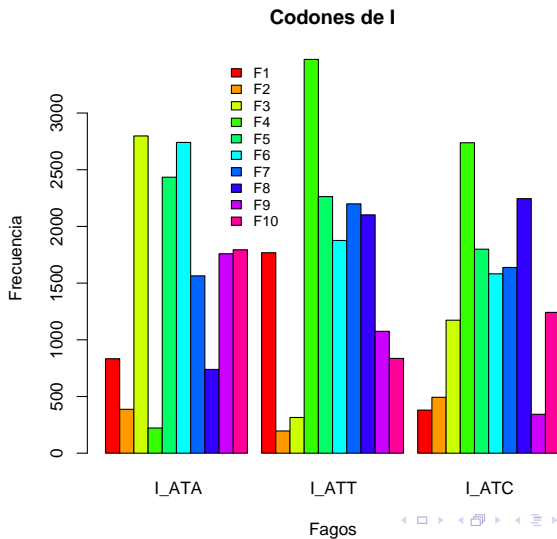
PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart



## Otro tipo de barras

- Ahora vamos a cambiar nuestra gráfica de barras. Vamos a hacerlas acumulativas. Además, usaremos nuevos colores.

- ▶ Usamos `heat.colores` para obtener colores tipo heatmap para nuestra gráfica.
- ▶ `space` para marcar el espacio entre cada barra

```
> par(xpd = T, mar = par()$mar +  
+     c(0, 0, 0, 4))  
> barplot(codonesI, main = "Codones de I",  
+         ylab = "Frecuencia", col = heat.colors(3),  
+         space = 0.1, cex.axis = 0.8,  
+         las = 1, names.arg = c("F1",  
+                                "F2", "F3", "F4", "F5",  
+                                "F6", "F7", "F8", "F9",  
+                                "F10"), cex = 0.8)
```



# Otro tipo de barras

```
> legend(0.5, 5500, rownames(codonesI),  
+       cex = 0.8, fill = heat.colors(3))  
> par(mar = c(5, 4, 4, 2) + 0.1)
```

# Graf 3: paso 4

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catagóricos

Discretos

Contínuos

Gráficas

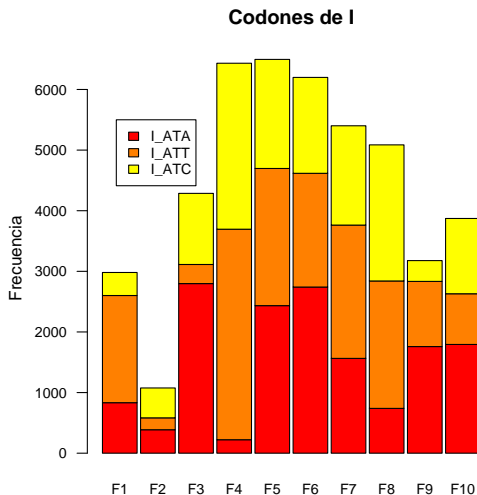
PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart



# Histogramas

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catagóricos

Discretos

Contínuos

Gráficas

PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

- Para empezar, hacemos un histograma simple.
- En este caso vamos a usar los datos de babyboom<sup>11</sup>. La variable `running.time` almacena la hora después de medianoche a la cual nacieron los bebés.
  - ▶ Simplemente usamos la función `hist`.

---

<sup>11</sup>Tienen que haber usado `library(UsingR)`

```
> hist(diff(babyboom$running.time))
```

# Graf 4: paso 1

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catagóricos

Discretos

Contínuos

Gráficas

PNG, PDF

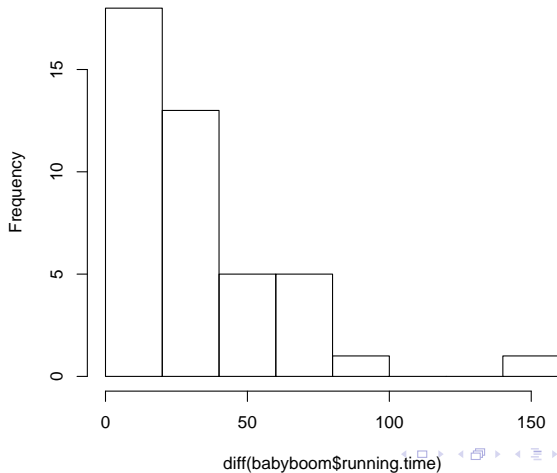
Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

Histogram of `diff(babyboom$running.time)`



# Ahora con color

- En este caso vamos a usar los datos de `nym.2002`. La variable `time` almacena los tiempos que hicieron los concursantes del maratón de ese año.

- ▶ Simplemente usamos el tan conocido argumento `col`.
- ▶ Además, delimito el eje Y con `ylim`

```
> hist(nym.2002$time, col = "light blue",  
+      ylim = c(0, 500))
```

## Graf 4: paso 2

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catagóricos

Discretos

Contínuos

Gráficas

PNG, PDF

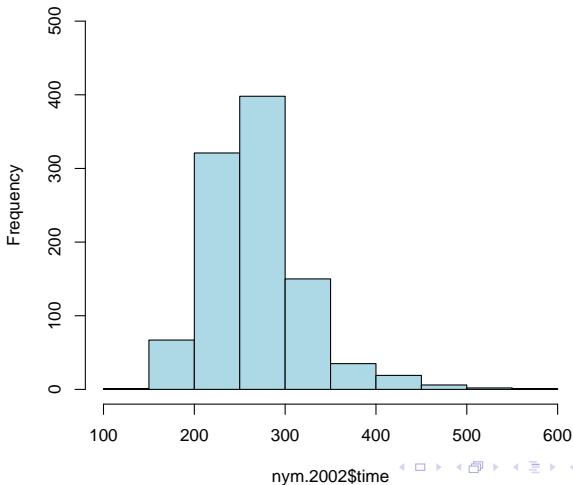
Gráficas de  
Barras

**Histogramas**

Aleatorios

Dotchart

Histogram of nym.2002\$time



# Usando `heat.colors`

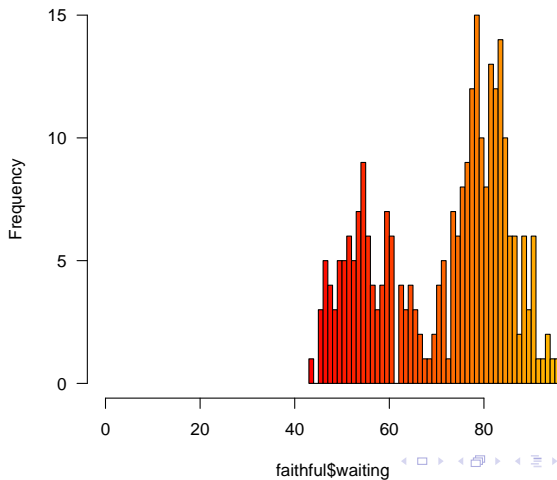
- En este caso vamos a usar los datos de `faithful`. La variable `waiting` almacena los tiempos una y otra erupción de "Faithful".
  - ▶ Usamos el argumento `breaks` para especificar cuantas intervalos queremos en el histograma.
  - ▶ Con `right` estamos deshabilitando el que cierren por el lado derecho los intervalos del histograma.
  - ▶ `heat.colors` nos sirve para obtener colores tipo heatmap.

```
> max_num <- max(faithful$waiting)
> hist(faithful$waiting, col = heat.colors(max_num
+       breaks = max_num/2, xlim = c(0,
+       max_num), right = F, main = "Tiempos en
+       las = 1)
```



## Graf 4: paso 3

Tiempos entre erupciones de Faithful



# Densidad de probabilidad

- Aquí seguimos con los mismo datos y vamos a decidir los intervalos manualmente.

- ▶ Noten que a `breaks` le paso un vector manualmente.
- ▶ Como queremos la densidad de probabilidad, cambié el valor de `freq`

```
> brk <- c(0, 46, 55, 64, 69, 82,  
+         90, 100)  
> hist(faithful$waiting, col = heat.colors(length  
+     breaks = brk, xlim = c(0, max_num),  
+     right = F, main = "Densidad de Probabilidad  
+     cex.axis = 0.8, freq = F)
```

# Graf 4: paso 4

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catégoricos

Discretos

Contínuos

Gráficas

PNG, PDF

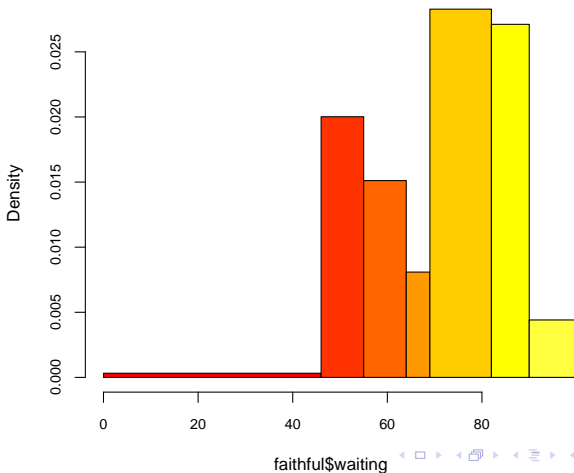
Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

Densidad de Probabilidad de las erupciones de Faithful



# Unos números aleatorios

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catégoricos

Discretos

Contínuos

Gráficas

PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

- Usando el paquete **mcmc** generamos unos número al azar. Estos los vamos a comparar visualmente con los que generamos en base a una distribución "log-normal".

```
> r <- rlnorm(1000)
> hist(r)
> library(mcmc)
> h <- function(x) if (all(x >= 0) &&
+   sum(x) <= 1) return(1) else return(-Inf)
> out <- metrop(h, rep(0, 5), 1000)
> out <- metrop(out, scale = 0.1)
> out <- metrop(out, nbatch = 10000)
> hist(out$batch[, 1])
```

# Graf 5: paso 1a

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catagóricos

Discretos

Contínuos

Gráficas

PNG, PDF

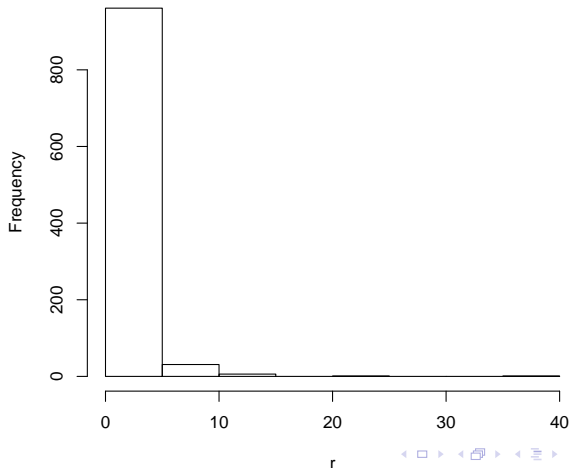
Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

Histogram of r



# Graf 5: paso 1b

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Categoricos

Discretos

Continuos

Gráficas

PNG, PDF

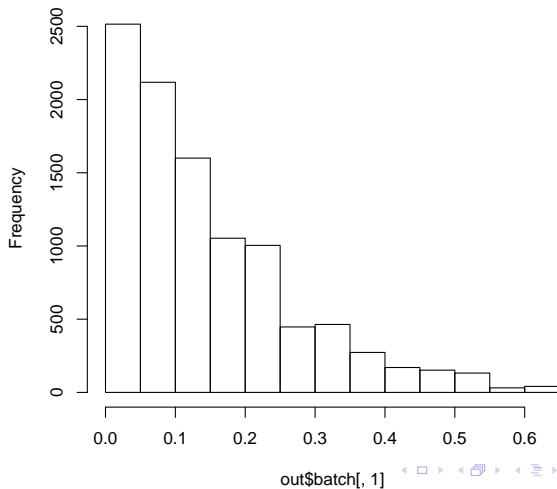
Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

Histogram of out\$batch[, 1]



# Viendo la dif

```
> h1 <- hist(r, plot = F, breaks = c(seq(0,
+   max(r) + 1, 0.1)))
> plot(h1$counts, log = "xy", pch = 20,
+   col = "blue", main = "Distribuci363n Log-normal",
+   xlab = "Valor", ylab = "Frecuencia")
> h2 <- hist(out$batch[, 1])
> plot(h2$counts, log = "xy", pch = 20,
+   col = "blue", main = "Distribuci363n con MCMC",
+   xlab = "Valor", ylab = "Frecuencia")
```



# Graf 5: paso 2a

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catagóricos

Discretos

Contínuos

Gráficas

PNG, PDF

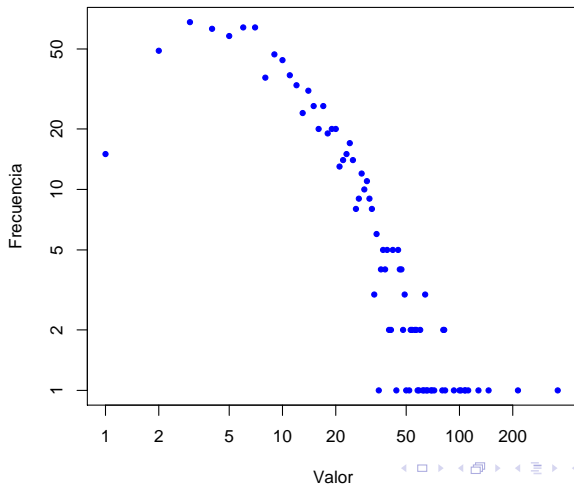
Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

**Distribución Log-normal**



# Graf 5: paso 2b

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catagóricos

Discretos

Contínuos

Gráficas

PNG, PDF

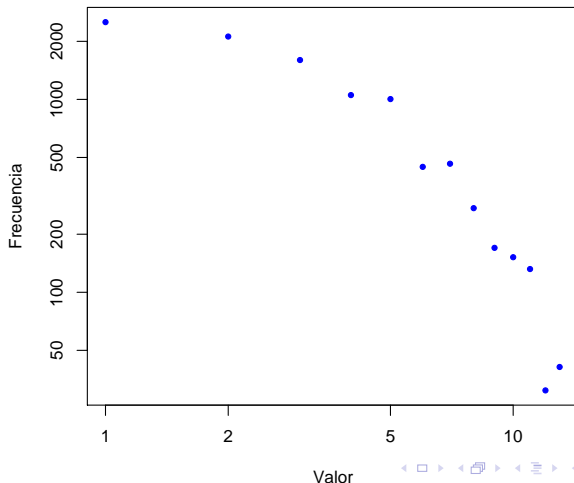
Gráficas de  
Barras

Histogramas

Aleatorios

Dotchart

Distribución con MCMC



- El último tipo de gráfica que vamos a ver hoy es el **dotchart**. Primero hacemos una normal y luego le ponemos colores! :)

```
> dotchart(t(codonesI))  
> dotchart(t(codonesI), col = rainbow(10),  
+         pch = 16, cex = 0.8)
```

# Graf 6: paso 1

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Categoricos

Discretos

Contínuos

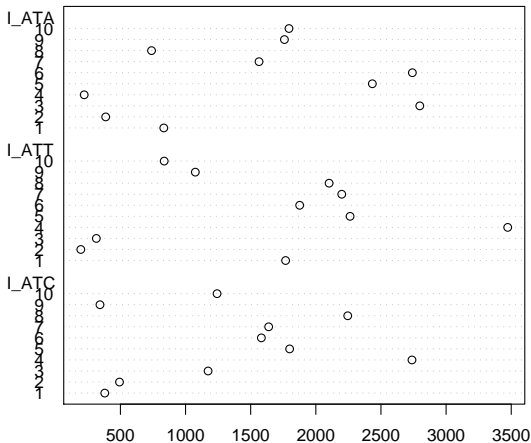
Gráficas

PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios



# Graf 6: paso 2

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos

Univariados

Catagóricos

Discretos

Contínuos

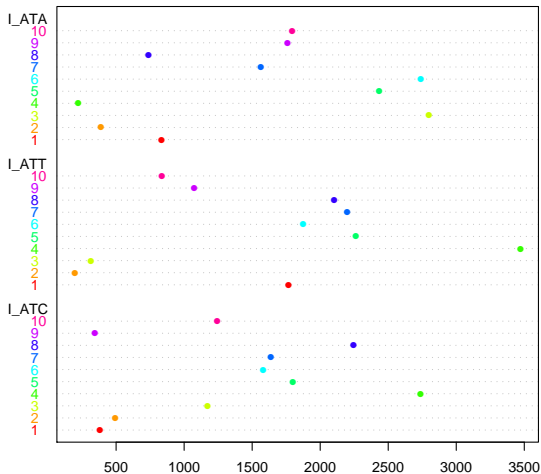
Gráficas

PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios



# Ya casi!

- Para empezar les muestro una gráfica con los diferentes símbolos que pueden usar. Acuérdense de que son parámetros del argumento `pch`.

# pch

R /  
Bioconductor:  
Curso  
Intensivo

List

Factor

Apply

Datos  
Univariados

Catégoricos

Discretos

Contínuos

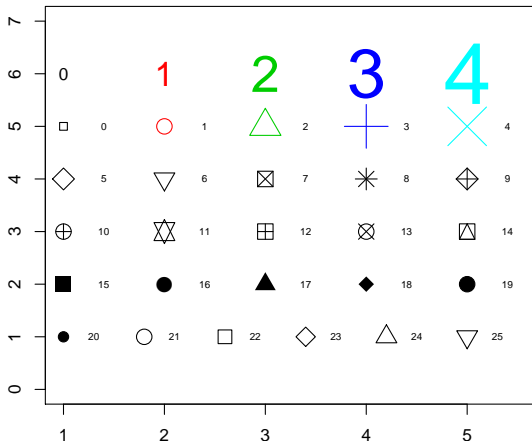
Gráficas

PNG, PDF

Gráficas de  
Barras

Histogramas

Aleatorios





- Como se habrán dado cuenta hay una infinidad de parámetros para las gráficas... muchos de los cuales se nos olvidan que son. Su manual de gráficas es simplemente escribir `?par`.
- Por ejemplo, puedes tener 2 o 4 gráficas en lugar de una usando el argumento `mfrow`.
- Ahora, por favor chequen el `ejercicio 2`. Suerte!