# Visualización básica

## HR Analytics: Teoría y Práctica

http://pablohaya.com/contact

09/2018

# Leemos datos

Leemos el conjunto de datos que creamos en el tema anterior como combinación de `core_dataset.csv` y `production_staff.csv`

```
library(here)
load(here("data","join_dataset_clean.RData"))
```

# Resumen

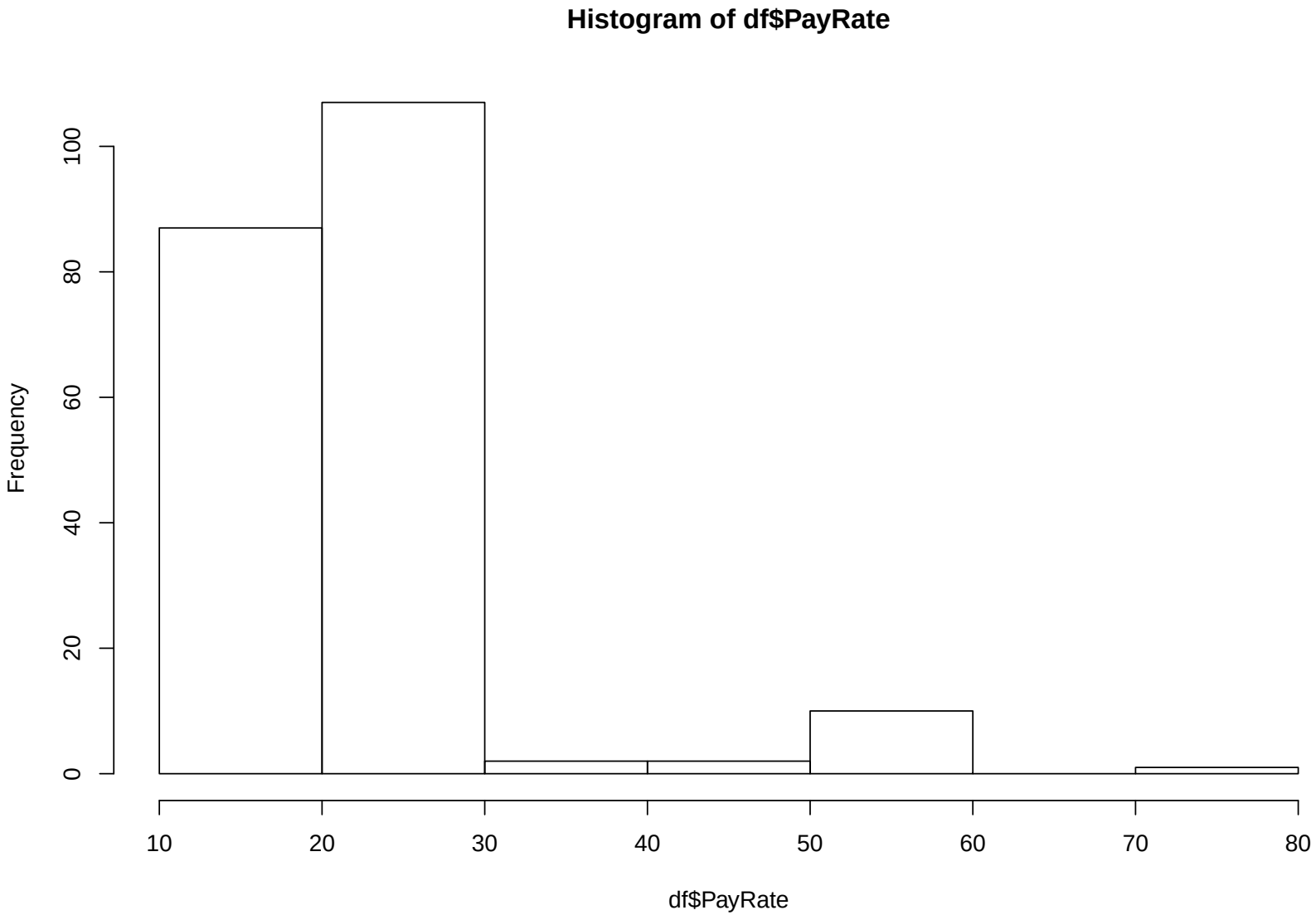
Vamos a ver el resumen de la variable `PayRate`.

```
summary(df$PayRate)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    14.0    18.0    22.0    23.2    24.5    80.0
```

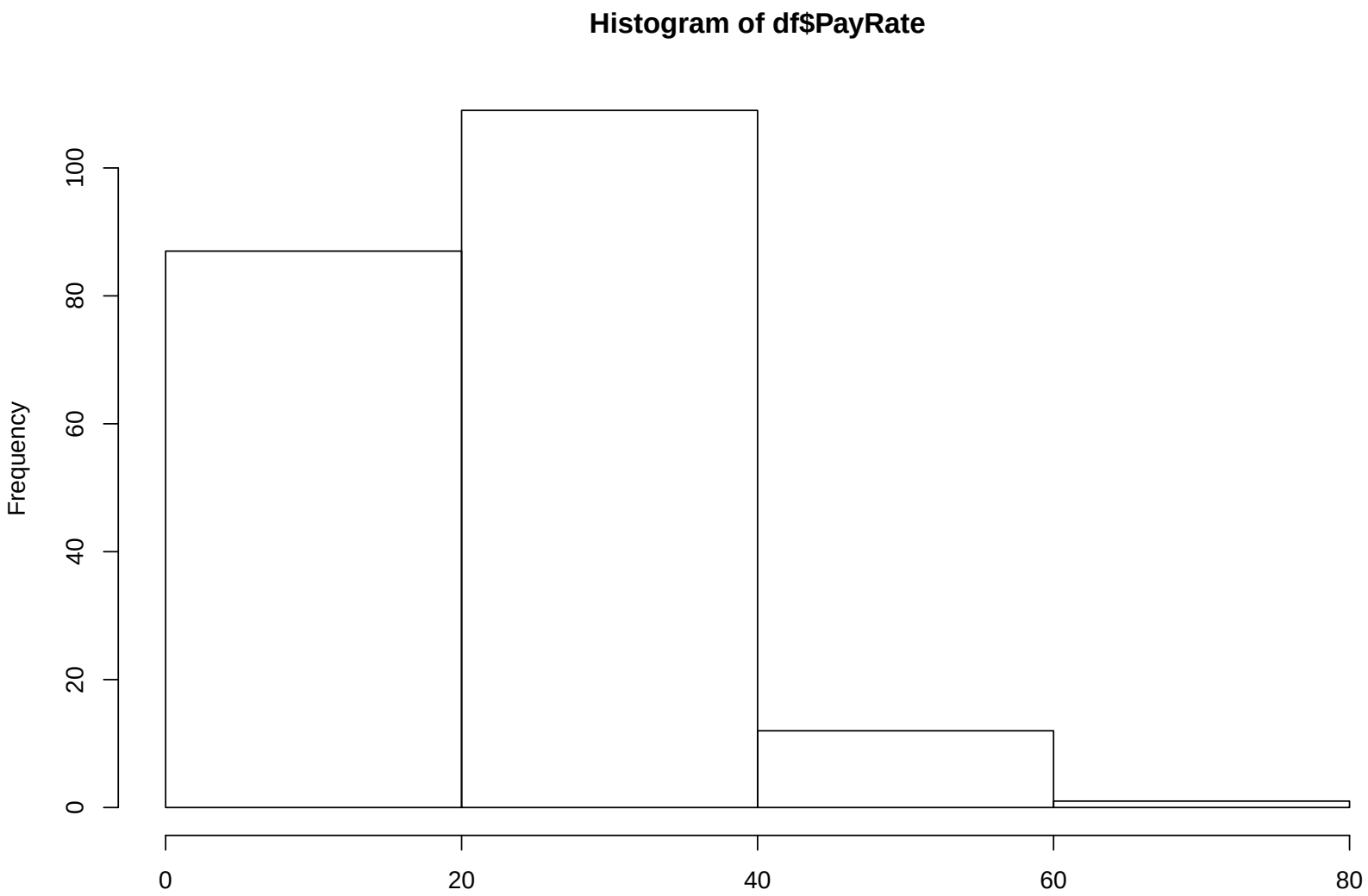
# Histogramas

Los histogramas nos permiten visualizar la forma que tiene la distribución de una variable empleando la función `hist()`.

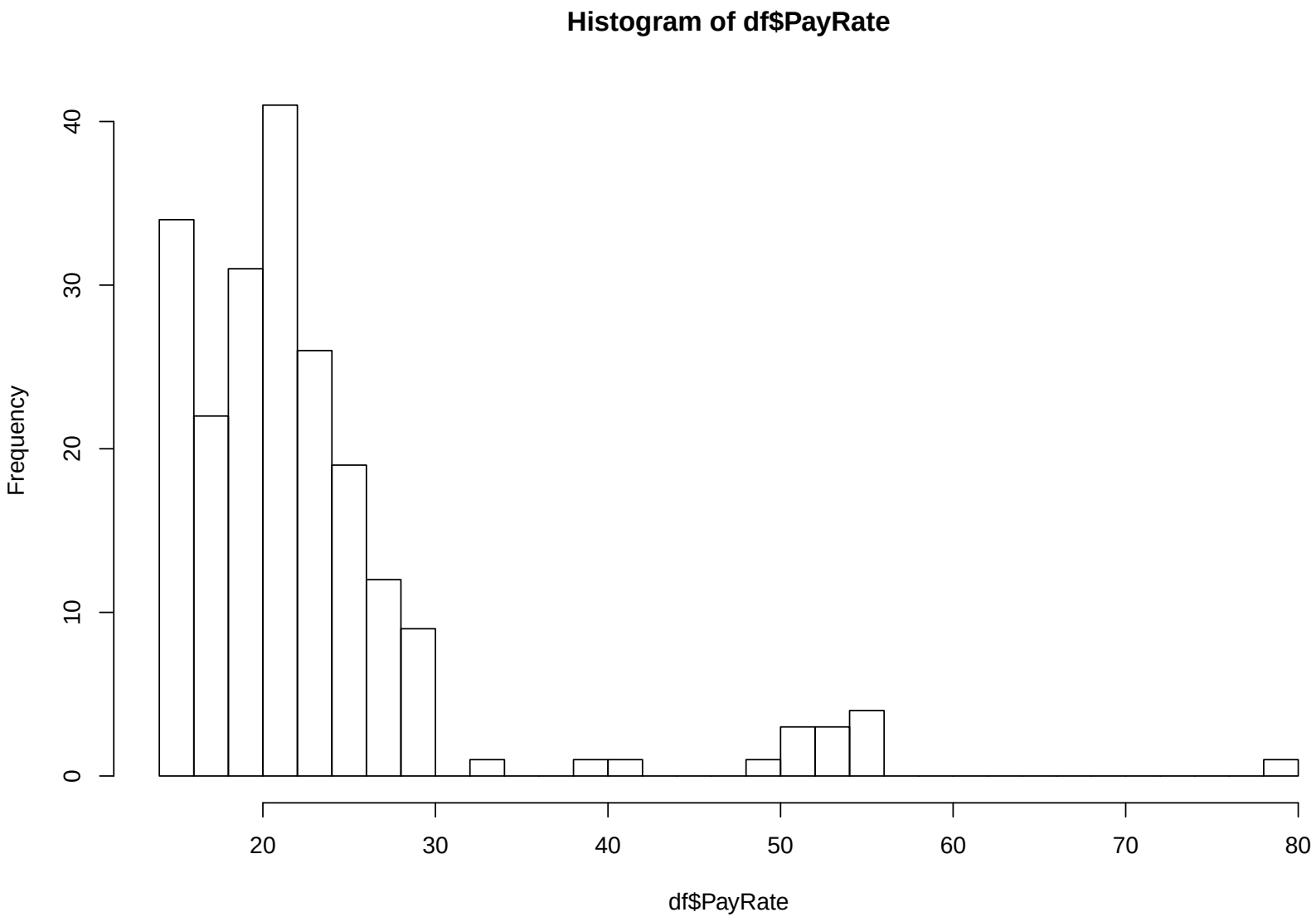

Histogram of df$PayRate
Frequency
0
20
40
60
80
100
10
20
30
40
50
60
70
80
df$PayRate

Es importante tener en cuenta que la elección de los puntos de corte de las barras pueden cambiar el histograma.

```
hist(df$PayRate, breaks = 3)
```



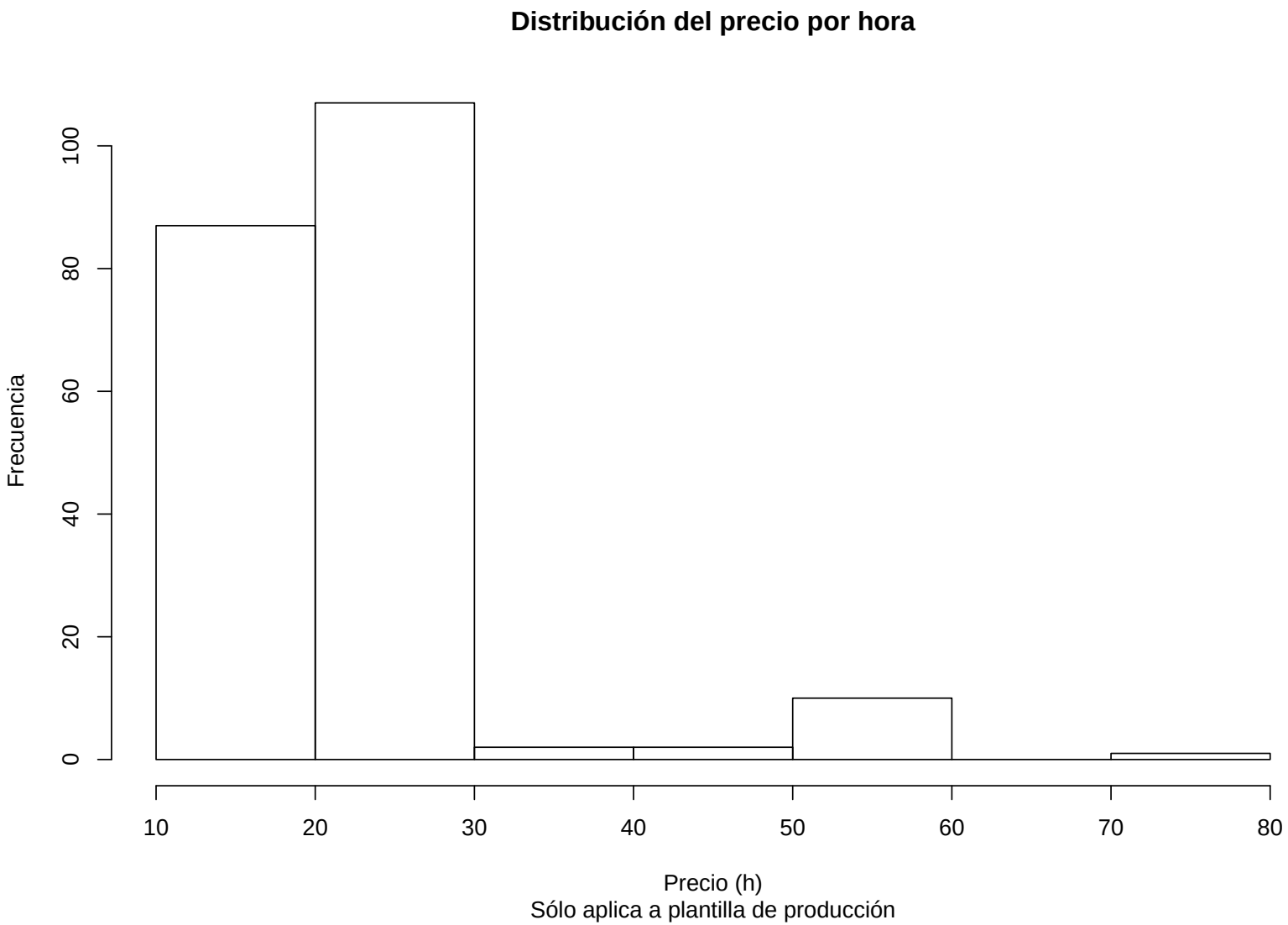
```r
hist(df$PayRate, breaks = 30)
```



# Personalizando la gráfica

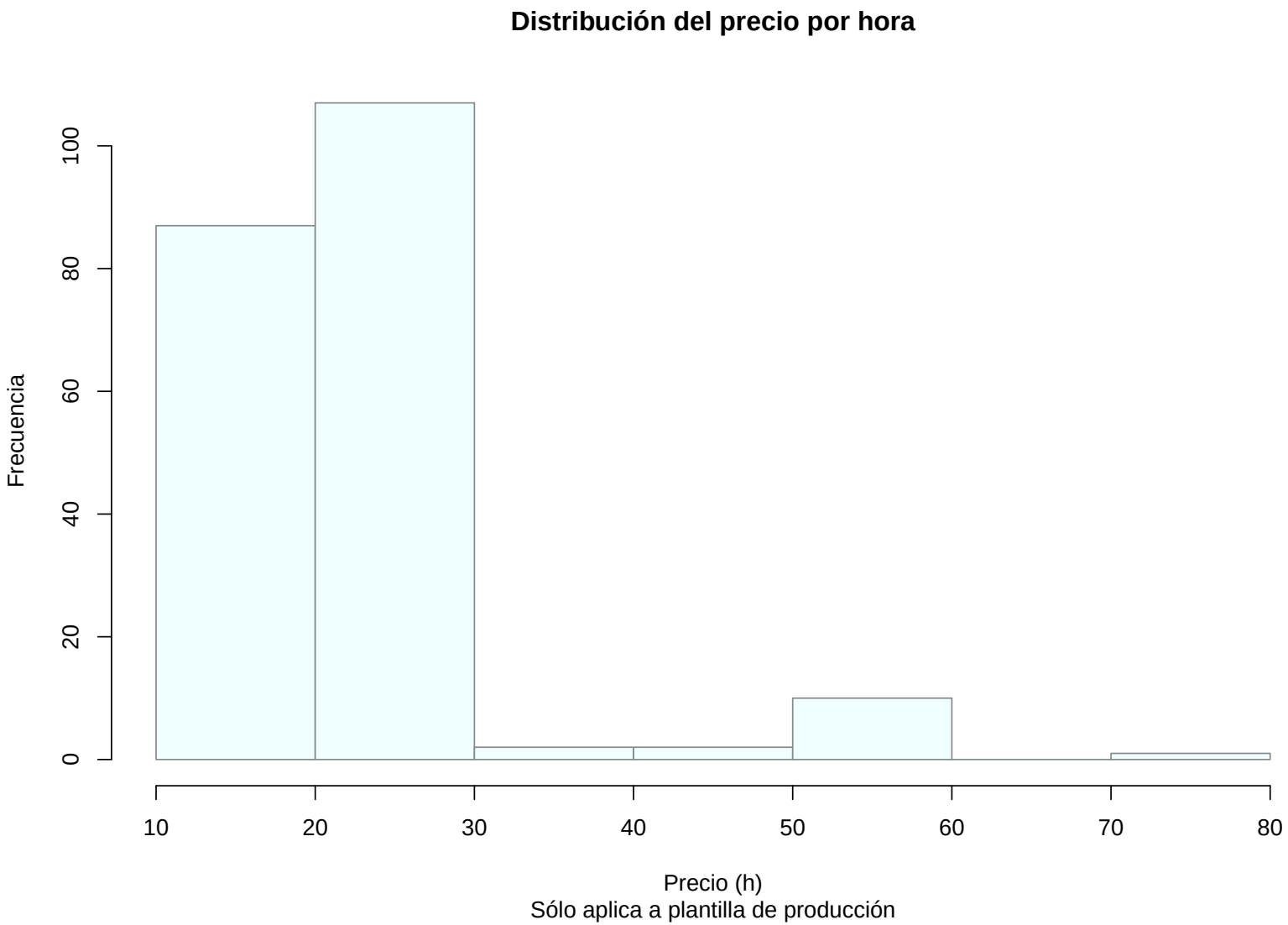
Es posible personalizar la gráfica incluyendo distintos parámetros como el título de la gráfica `main`, la etiquetas de los ejes (`xlab`, `ylab`), o subtítulo (`sub`).

```
hist(df$PayRate,
     main="Distribución del precio por hora",
     sub="Sólo aplica a plantilla de producción",
     xlab="Precio (h)", ylab="Frecuencia")
```


Distribución del precio por hora
Frecuencia
0
20
40
60
80
100
10
20
30
40
50
60
70
80
Precio (h)
Sólo aplica a plantilla de producción

Los colores y bordes de las barras también se pueden modificar con los parámetros `col` y `border` respectivamente.

```
hist(df$PayRate,
     main="Distribución del precio por hora",
     sub="Sólo aplica a plantilla de producción",
     xlab="Precio (h)", ylab="Frecuencia",
     col="azure", border="azure4")
```


Distribución del precio por hora
Frecuencia
0
20
40
60
80
100
10
20
30
40
50
60
70
80
Precio (h)
Sólo aplica a plantilla de producción

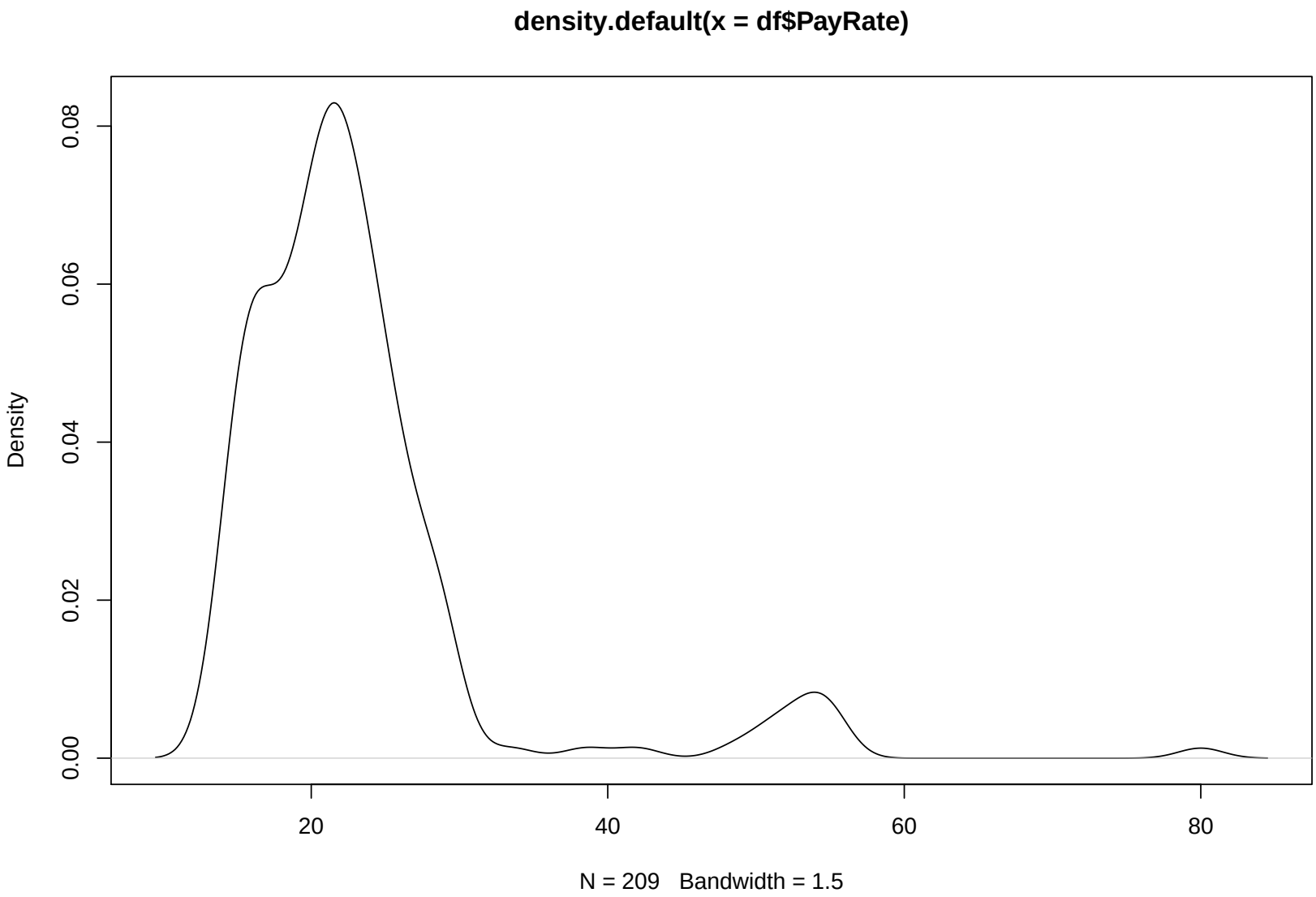
# Parámetros para definir colores

- `col` color por defecto
- `col.axis` color para los ejes
- `col.lab` color para el título de los ejes
- `col.main` color para el título principal
- `col.sub` color para el subtítulo
- `fg` color para los elementos visibles
- `bg` color para el fondo

# Función de densidad

La función de densidad es una aproximación que muestra de manera continua la distribución de una variable.

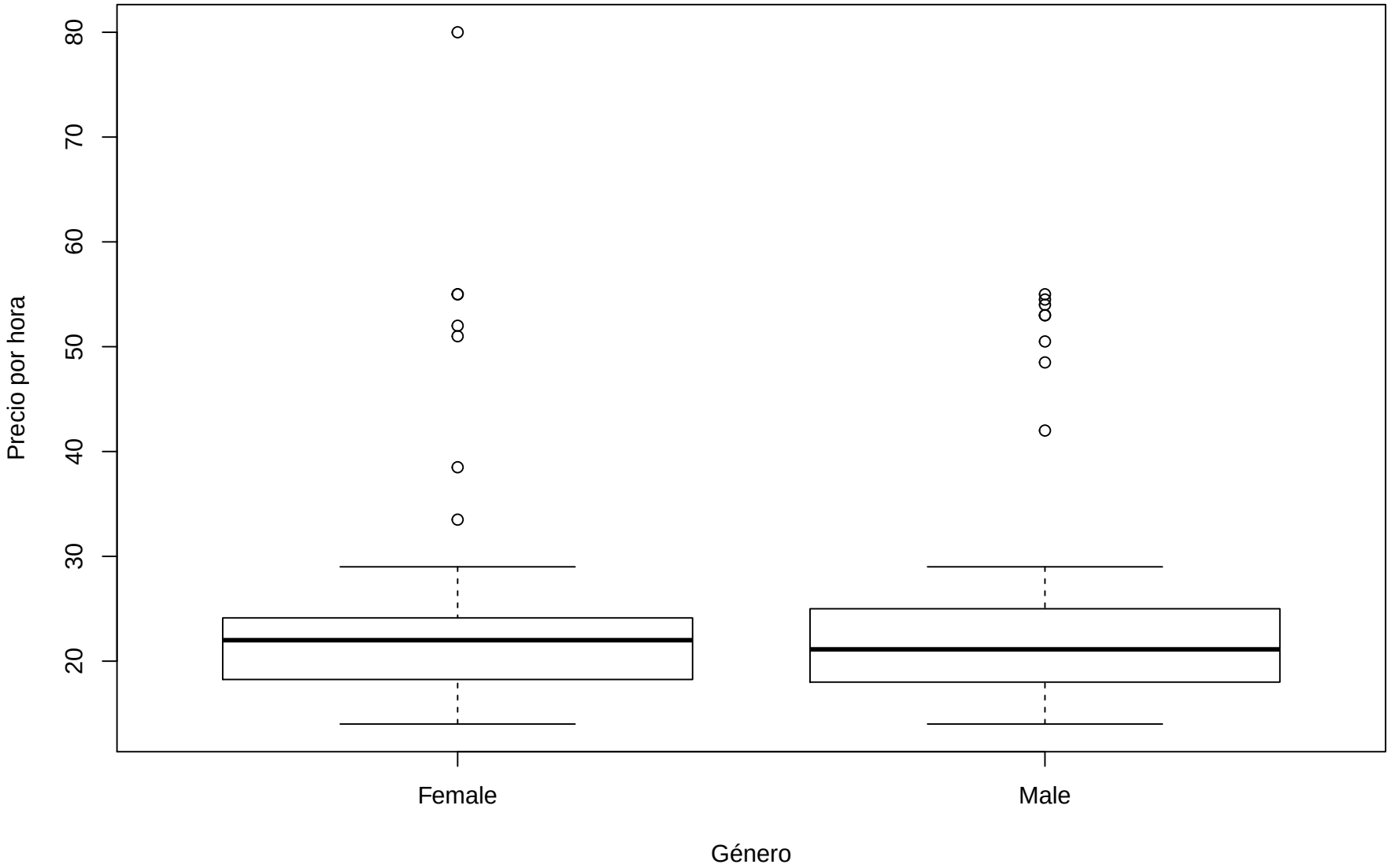
```
plot(density(df$PayRate))
```



# Boxplots

Los diagramas de caja resumen la distribución de una o más variables, y permiten compararlas fácilmente en una sola gráfica.

```r
boxplot(PayRate ~ Sex, data=df,
        xlab="Género", ylab="Precio por hora")
```

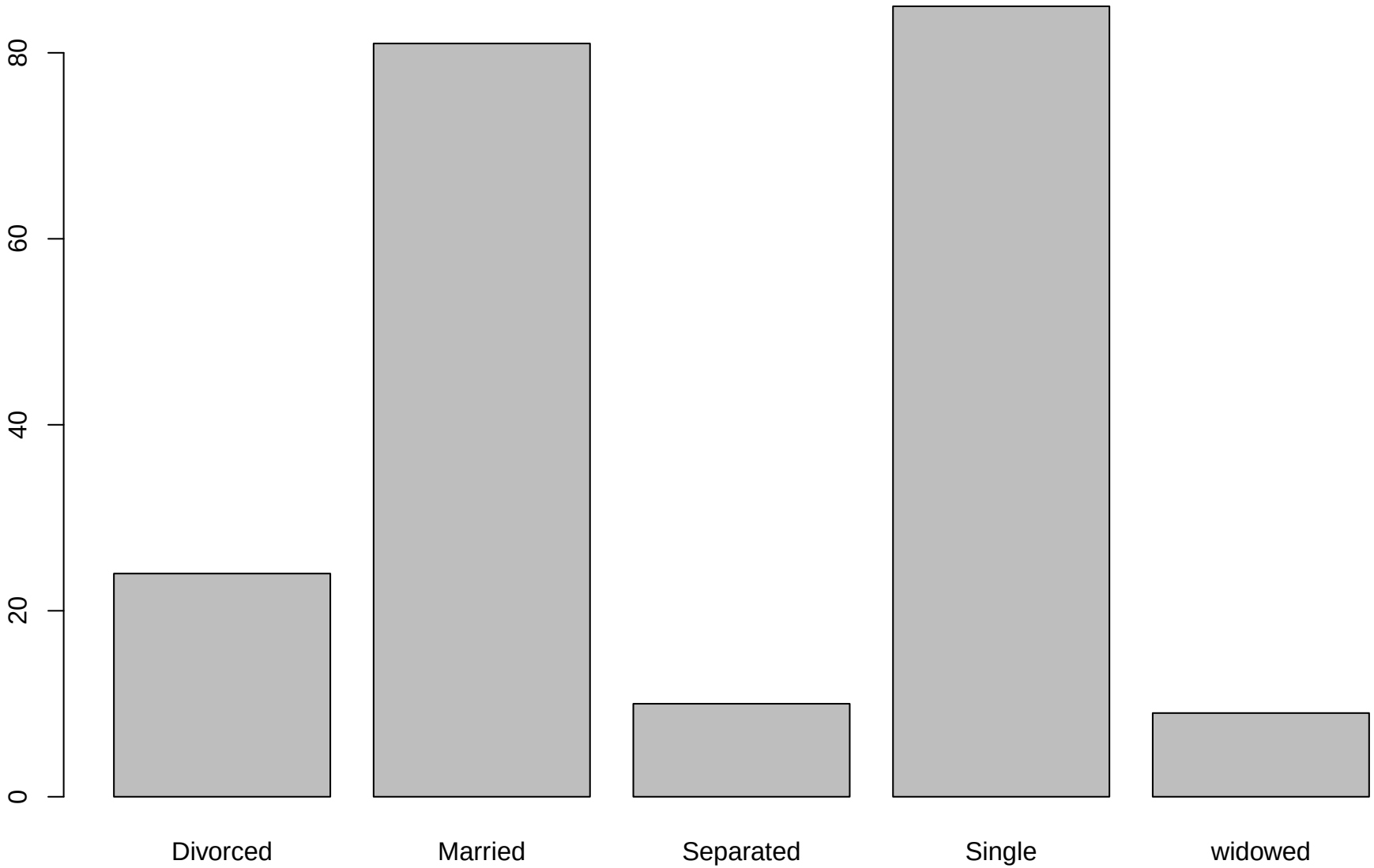


# Gráficos de barras

Los gráficos de barras permiten visualizar variables categóricas.

Aunque no se muestra en el gráfico, siguen siendo aplicables los parámetros para modificar el color de las barras

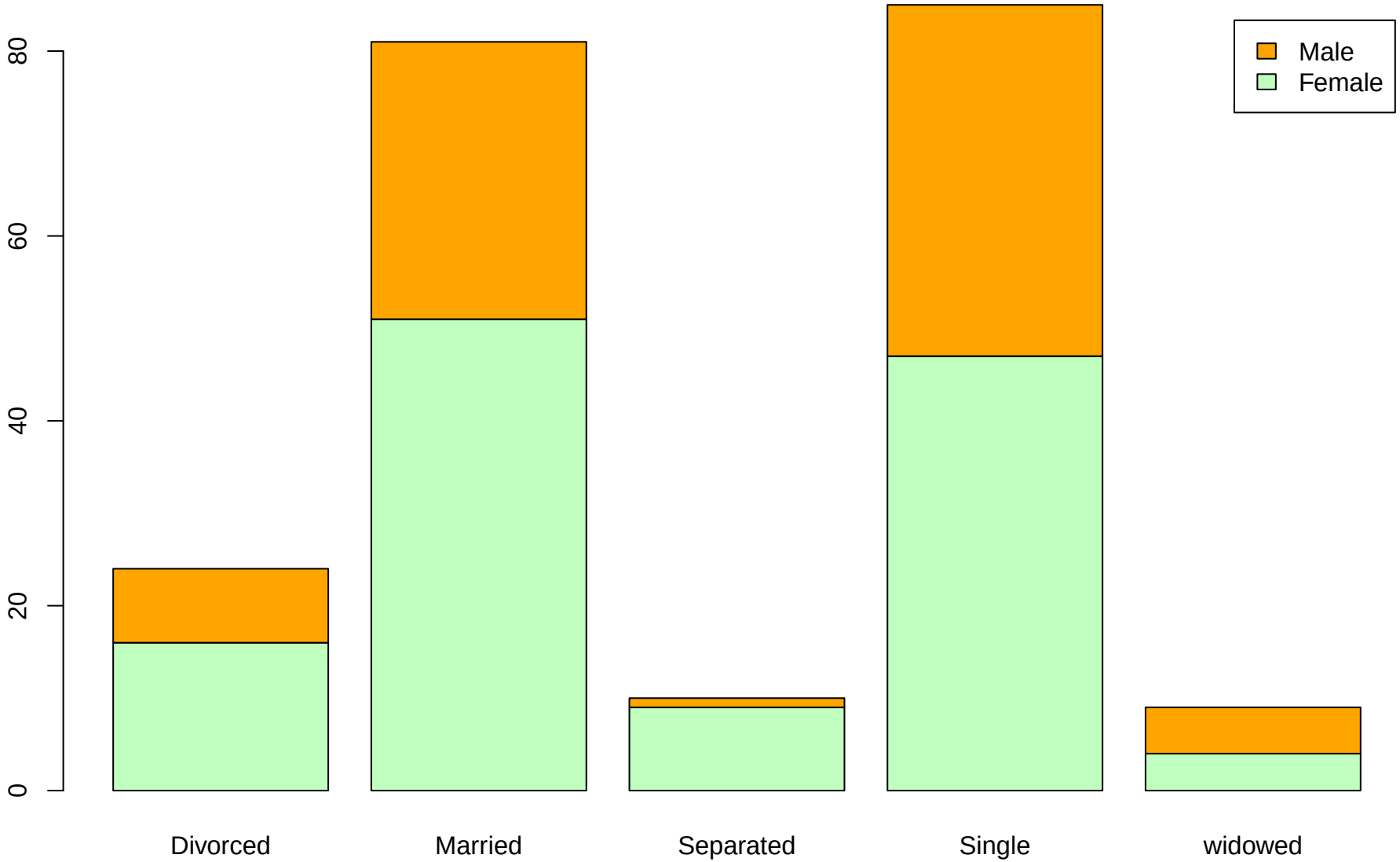
```r
count <- table(df$MaritalDesc)
barplot(count)
```



Se puede mostrar el cruce de dos variables categóricas agrupando las barras por colores.

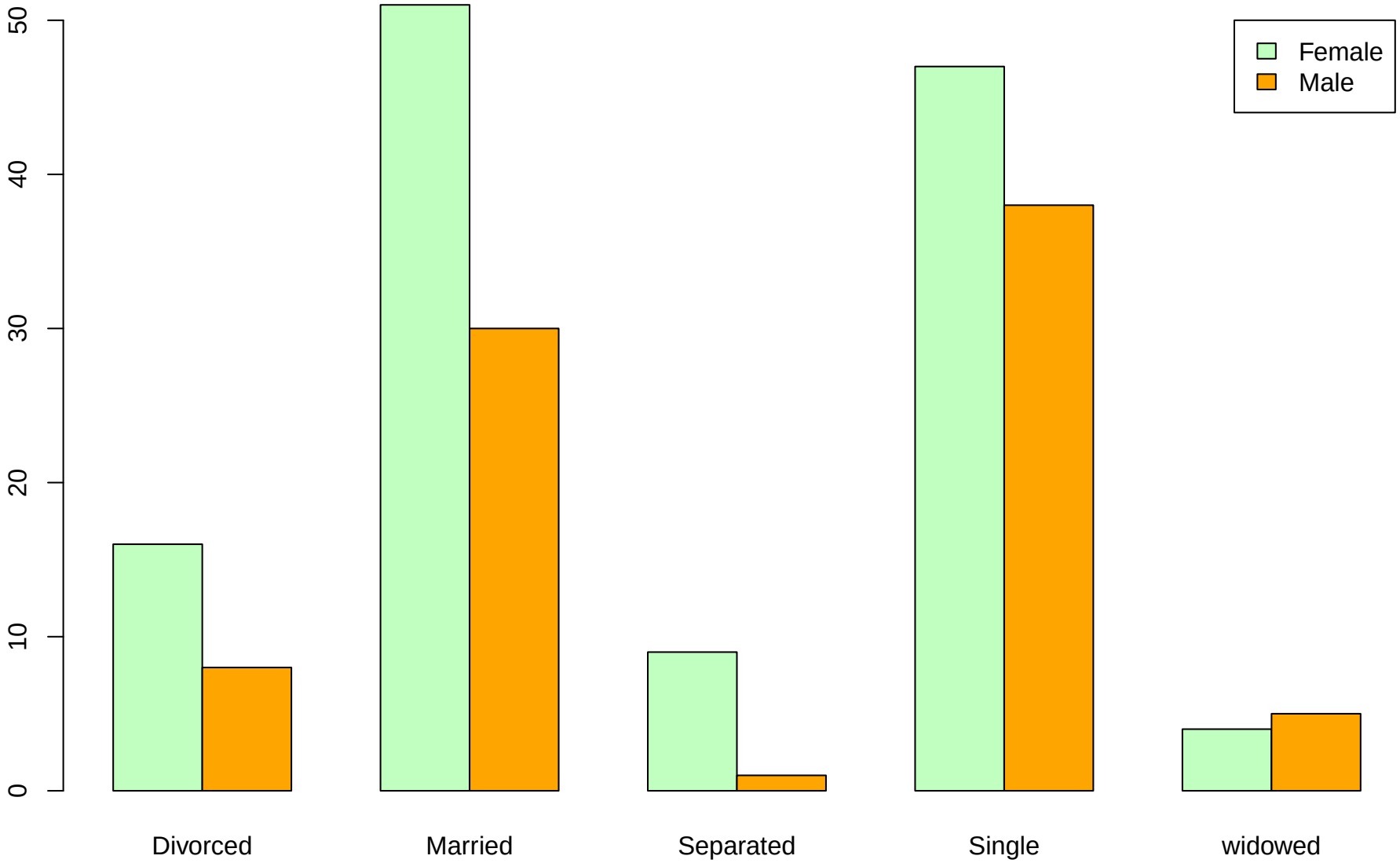
Primero se genera una tabla con las combinaciones, y luego se indican los colores a emplear. Opcionalmente se puede incluir leyenda (`legend`).

```
count <- table(df$Sex, df$MaritalDesc)
barplot(count, col=c("darkseagreen1", "orange"),
        legend = rownames(count))
```


Male
Female
0
20
40
60
80
Divorced
Married
Separated
Single
widowed

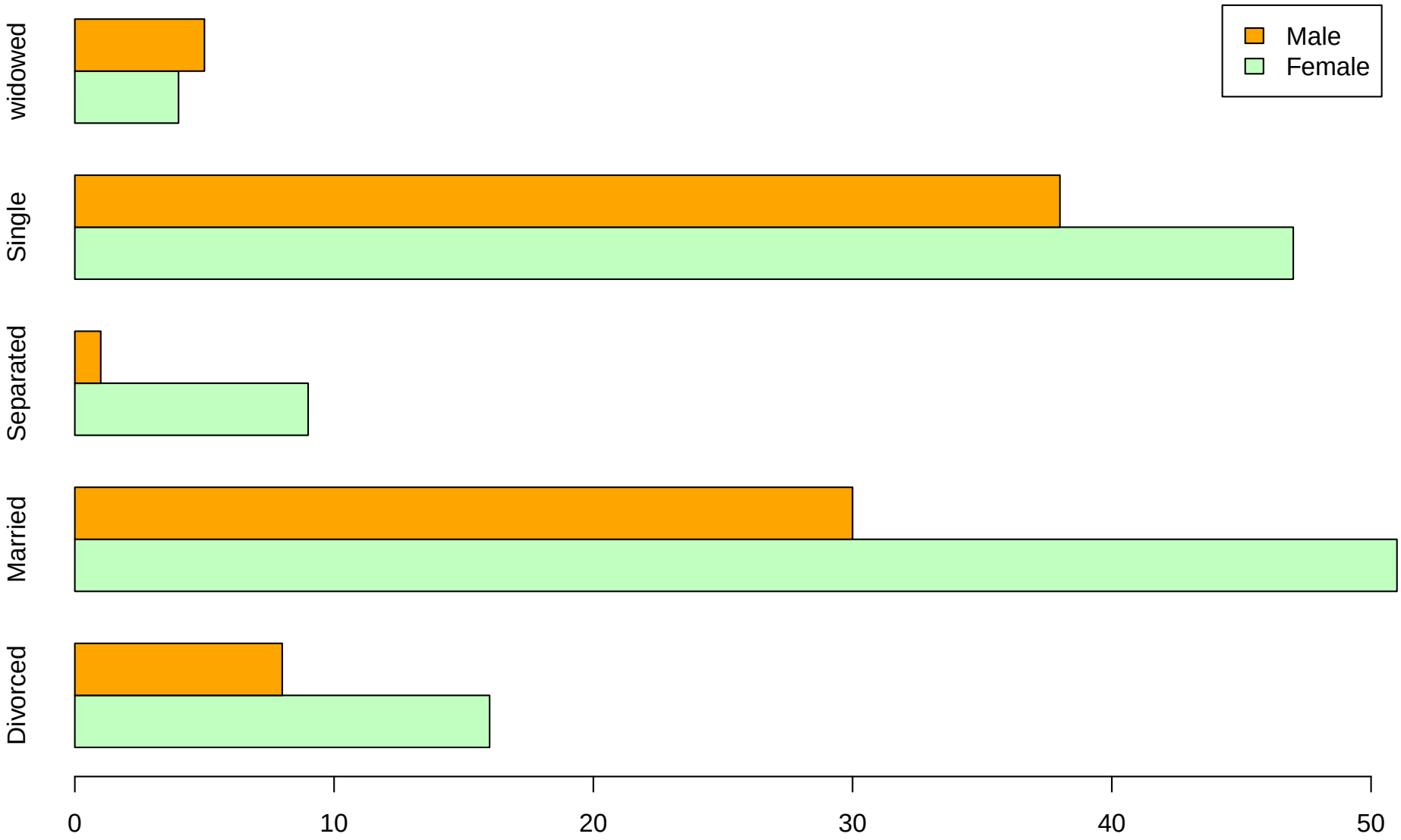
El párametro `beside` permite controlar la posición de las barras.

```
barplot(count,
        col=c("darkseagreen1", "orange"),
        legend = rownames(count),
        beside=TRUE)
```


0
10
20
30
40
50
Female
Male
Divorced
Married
Separated
Single
widowed

Mientras que el parámetro `horiz` nos permite girar las barras en posición horizontal.

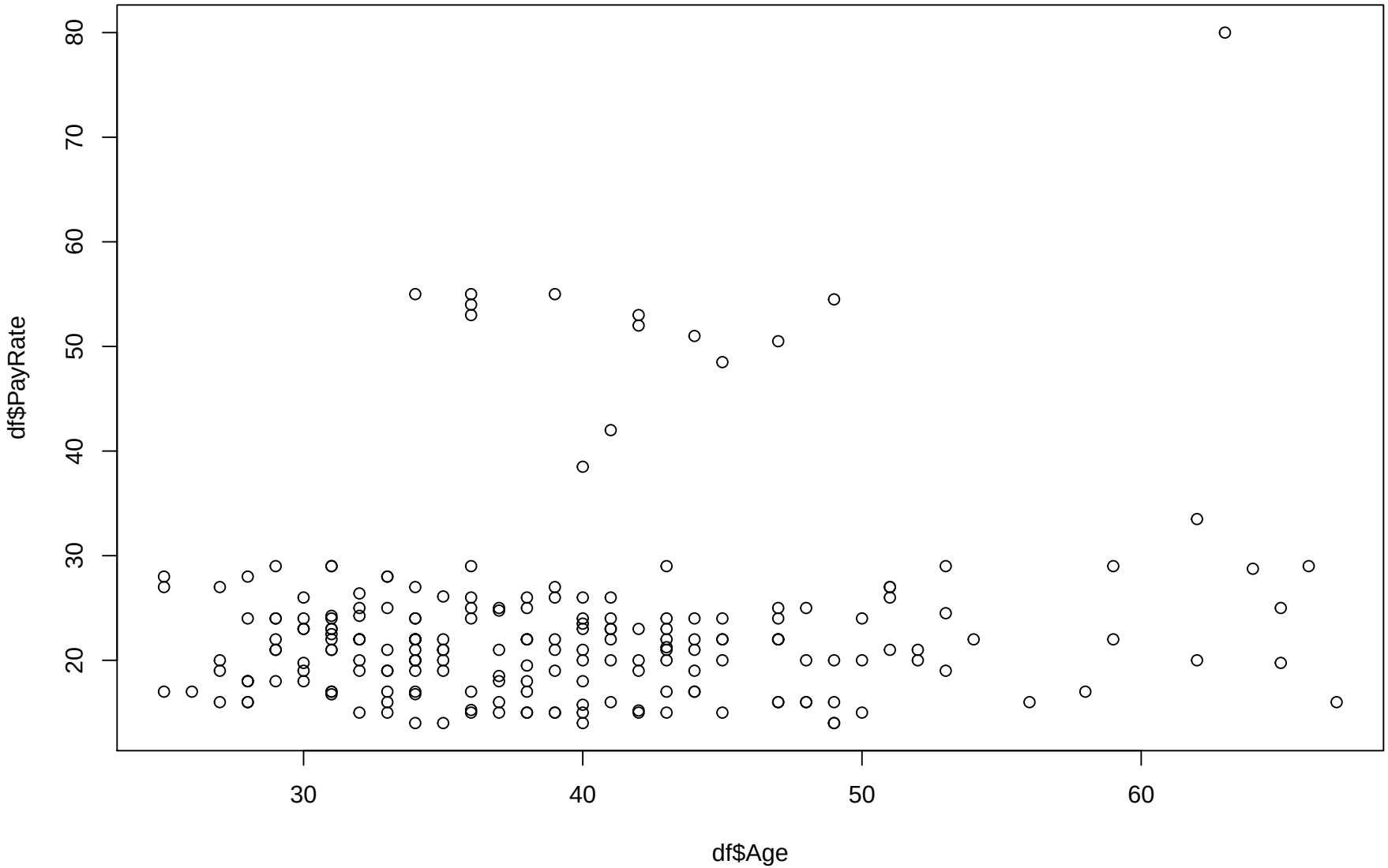
```
barplot(count,
        col=c("darkseagreen1", "orange"),
        legend = rownames(count),
        beside=TRUE, horiz=TRUE)
```


Male
Female
widowed
Single
Separated
Married
Divorced
0
10
20
30
40
50

# Función `plot`

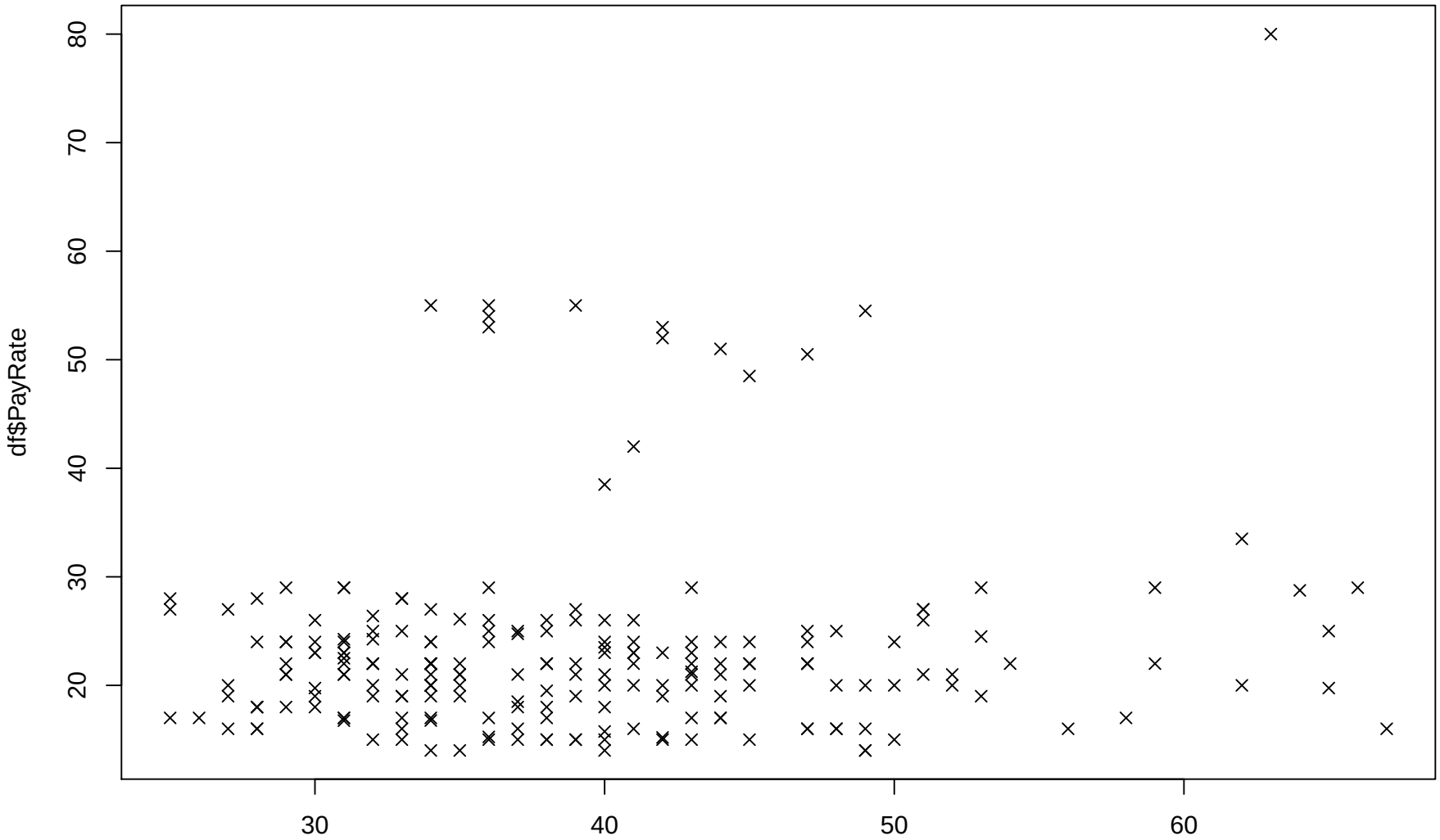
La función genérica `plot()` permite visualizar dos variables escogiendo una representación adecuada.

```r
plot(df$Age, df$PayRate)
```

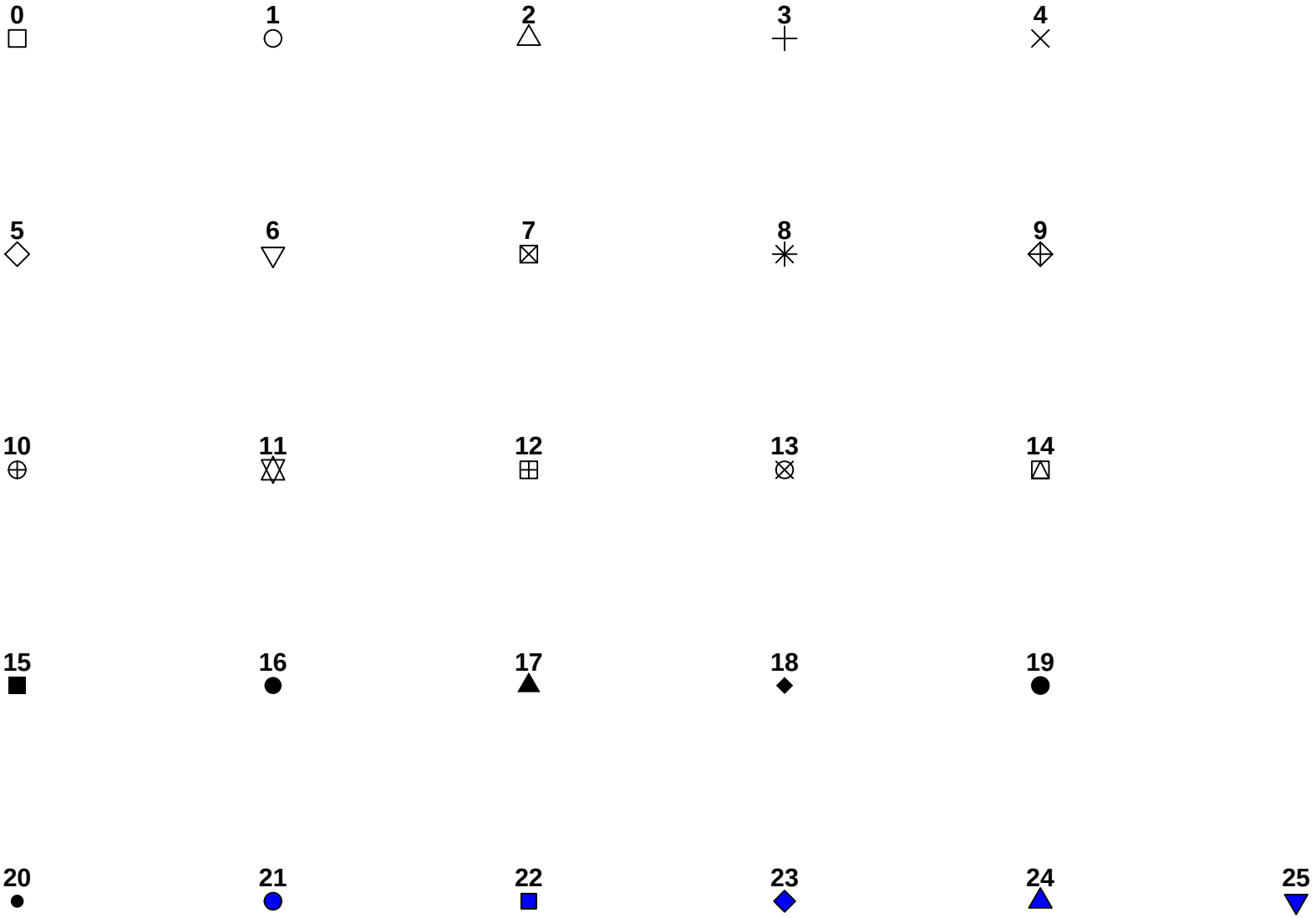


Se puede modificar el símbolo para representar un punto con el parámetro `pch`.

```
plot(df$Age, df$PayRate, pch=4)
```

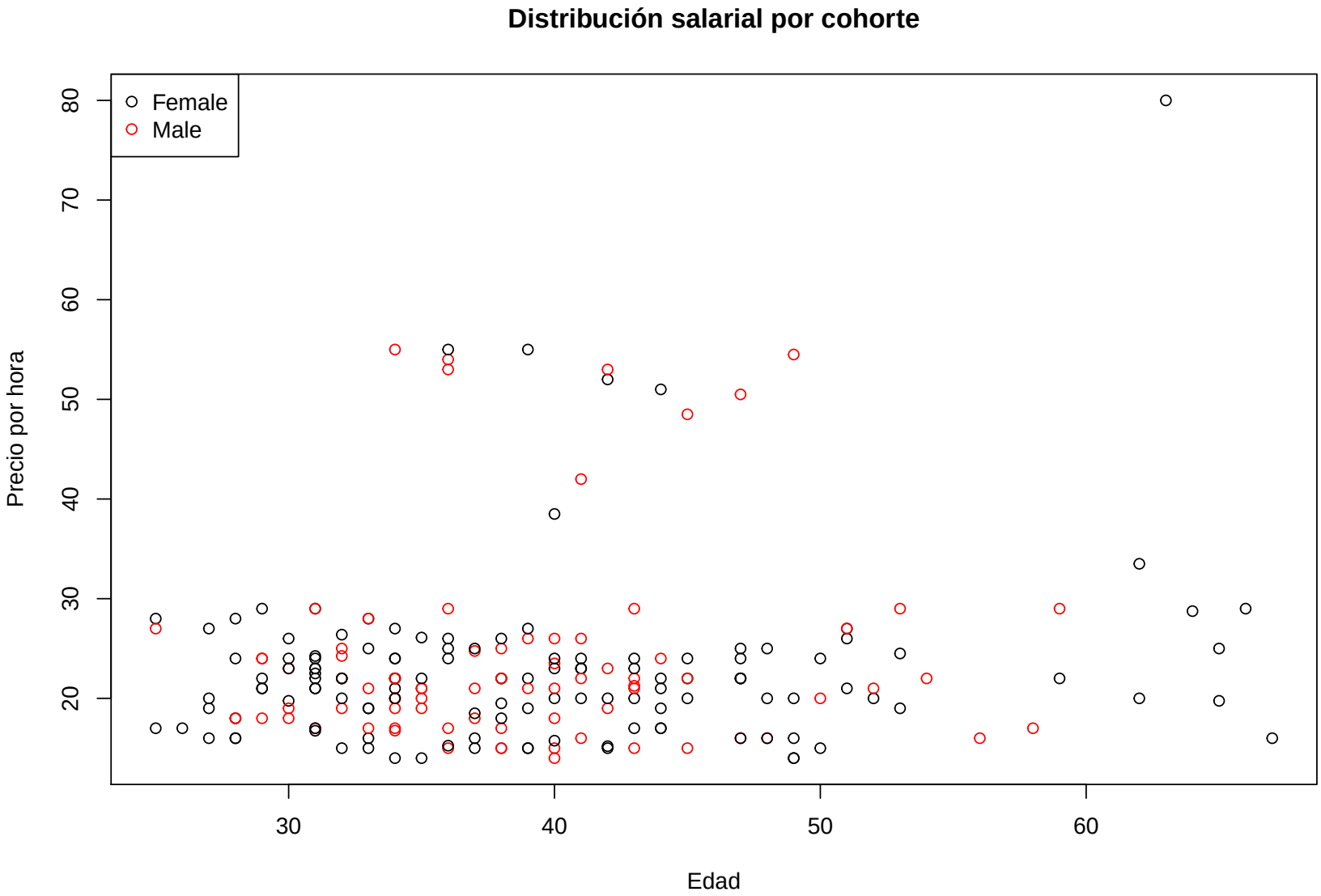


# Lista de símbolos disponibles



De nuevo es posible agrupar la gráfica por una variable adicional, y colorear los puntos. Así, como añadir los parámetros de personalización anteriormente visto.

```
plot(df$Age, df$PayRate, col=df$Sex,
     main="Distribución salarial por cohorte",
     xlab="Edad", ylab="Precio por hora")
legend("topleft",
       levels(df$Sex),
       col=1:length(levels(df$Sex)),
       pch = 1)
```


Distribución salarial por cohorte
Female
Male
Precio por hora
20
30
40
50
60
70
80
30
40
50
60
Edad

# Series temporales

Cuando el eje `x` se corresponde con una medida del tiempo, tenemos una serie temporal de una variable.

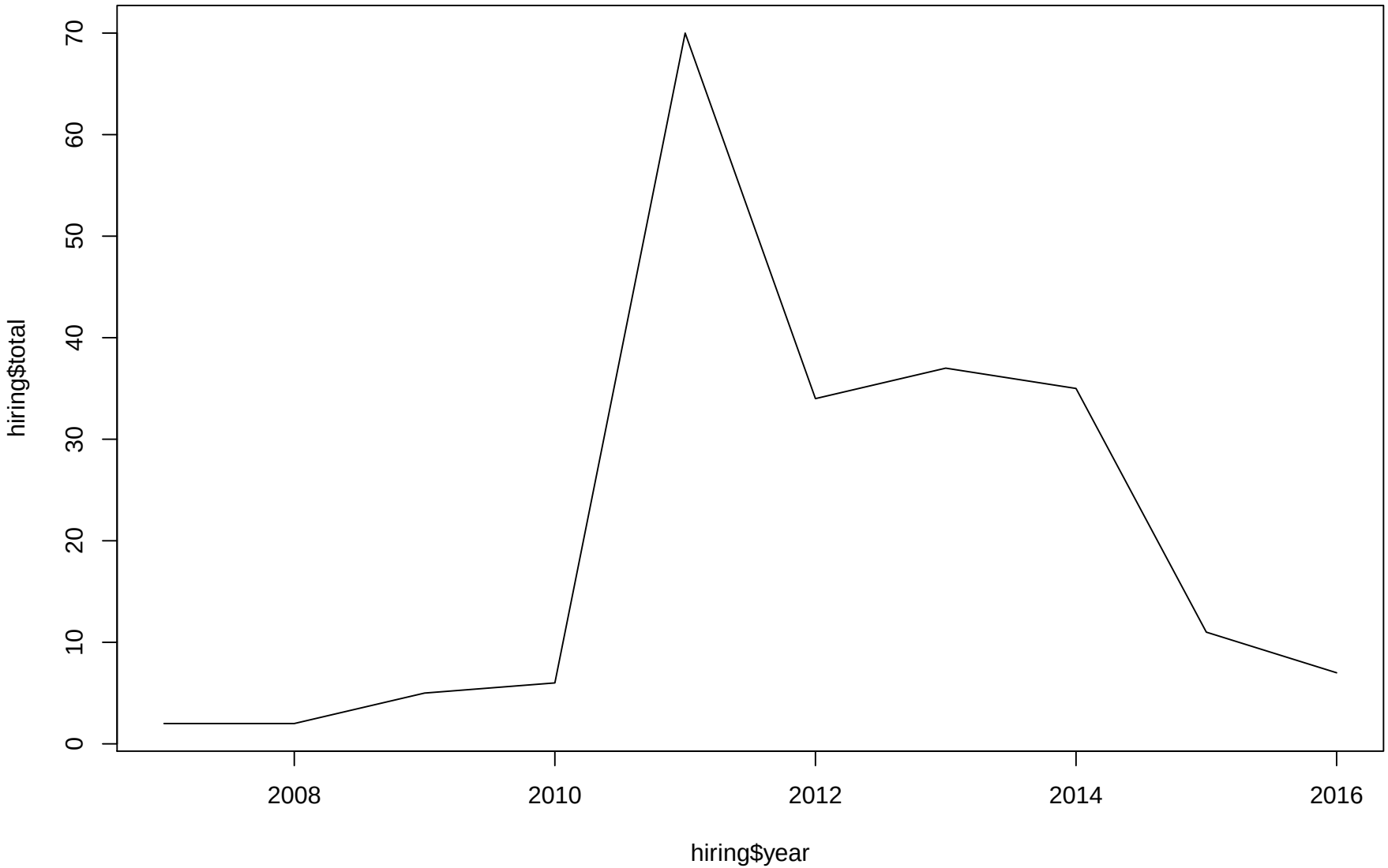
Primeramente, añadimos una variable que especifique el eje y la escala del tiempo.

```
library(tidyverse)
hiring <- df %>%
            mutate(year = format(DateofHire, "%Y")) %>%
            group_by(year) %>%
            summarise(total = n())
hiring
```

```
## # A tibble: 10 x 2
##    year  total
##    <chr> <int>
##  1 2007      2
##  2 2008      2
##  3 2009      5
##  4 2010      6
##  5 2011     70
##  6 2012     34
##  7 2013     37
##  8 2014     35
##  9 2015     11
## 10 2016      7
```

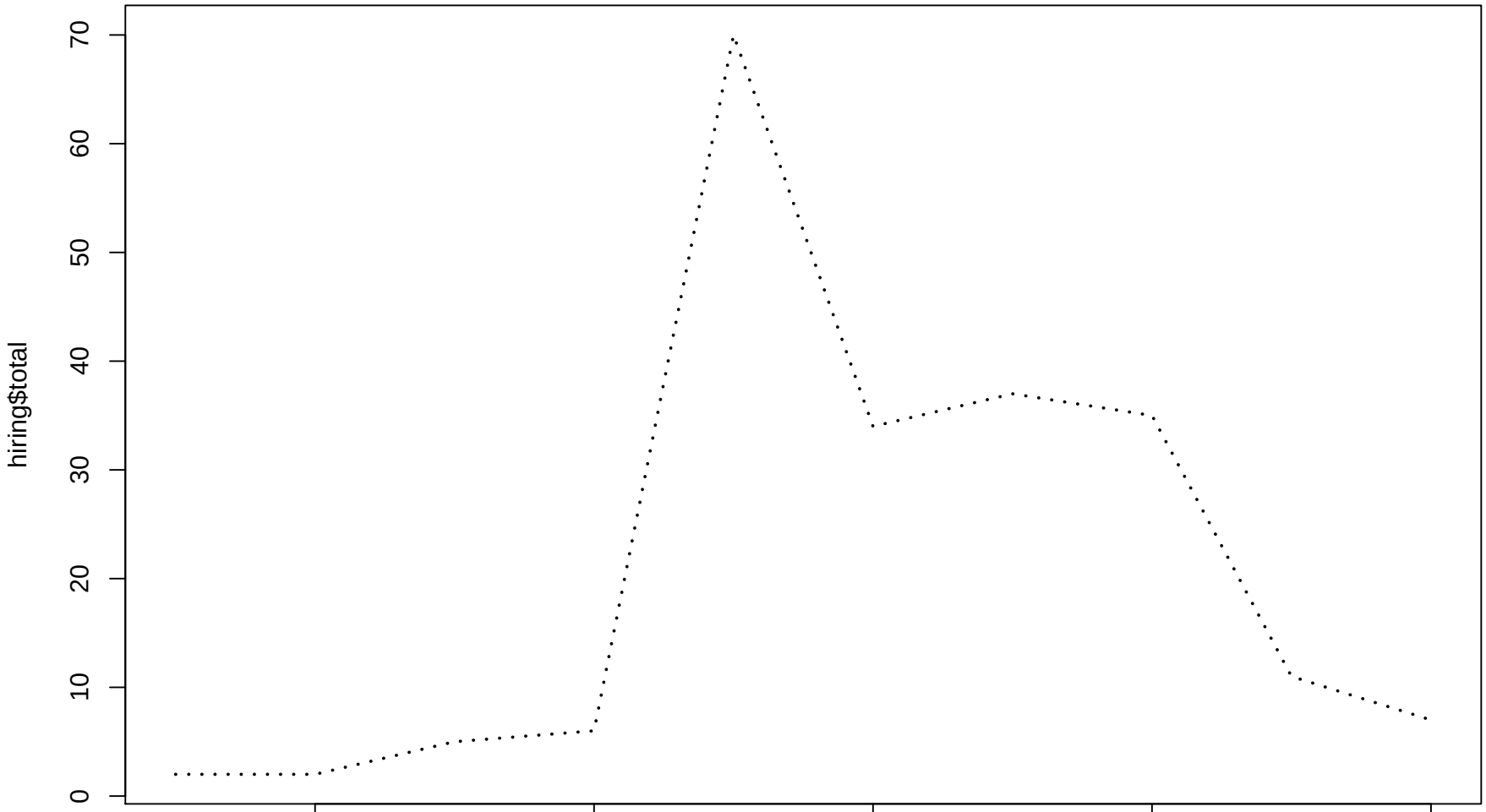
La serie temporal se visualiza indicándolo el parámetro `type`.

```
plot(hiring$year, hiring$total, type="l")
```



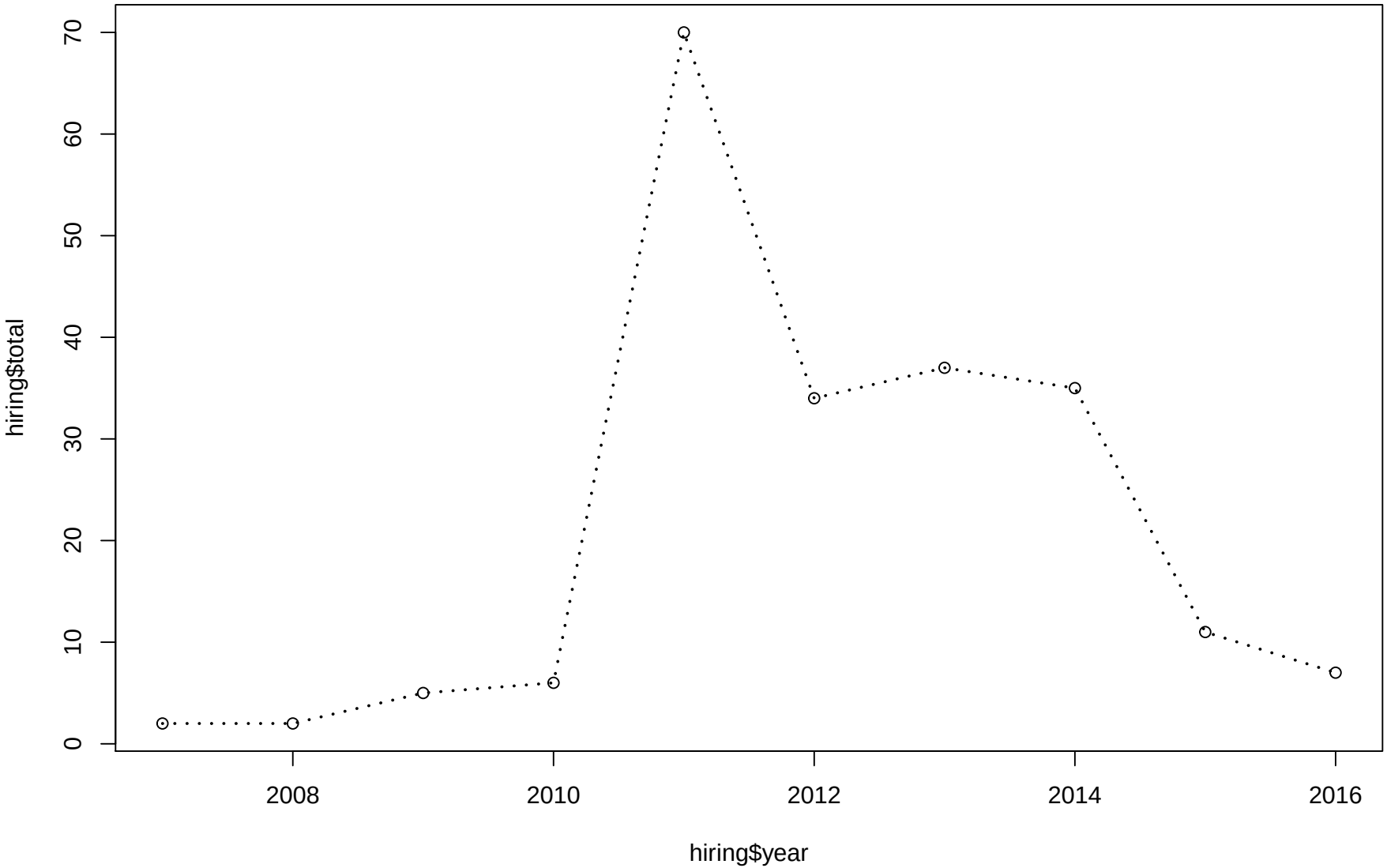
Los parámetros `lty` y `lwd` permiten cambiar el tipo de línea y el ancho respectivamente.

```
plot(hiring$year, hiring$total, type="l",
     lty=3, lwd=2)
```



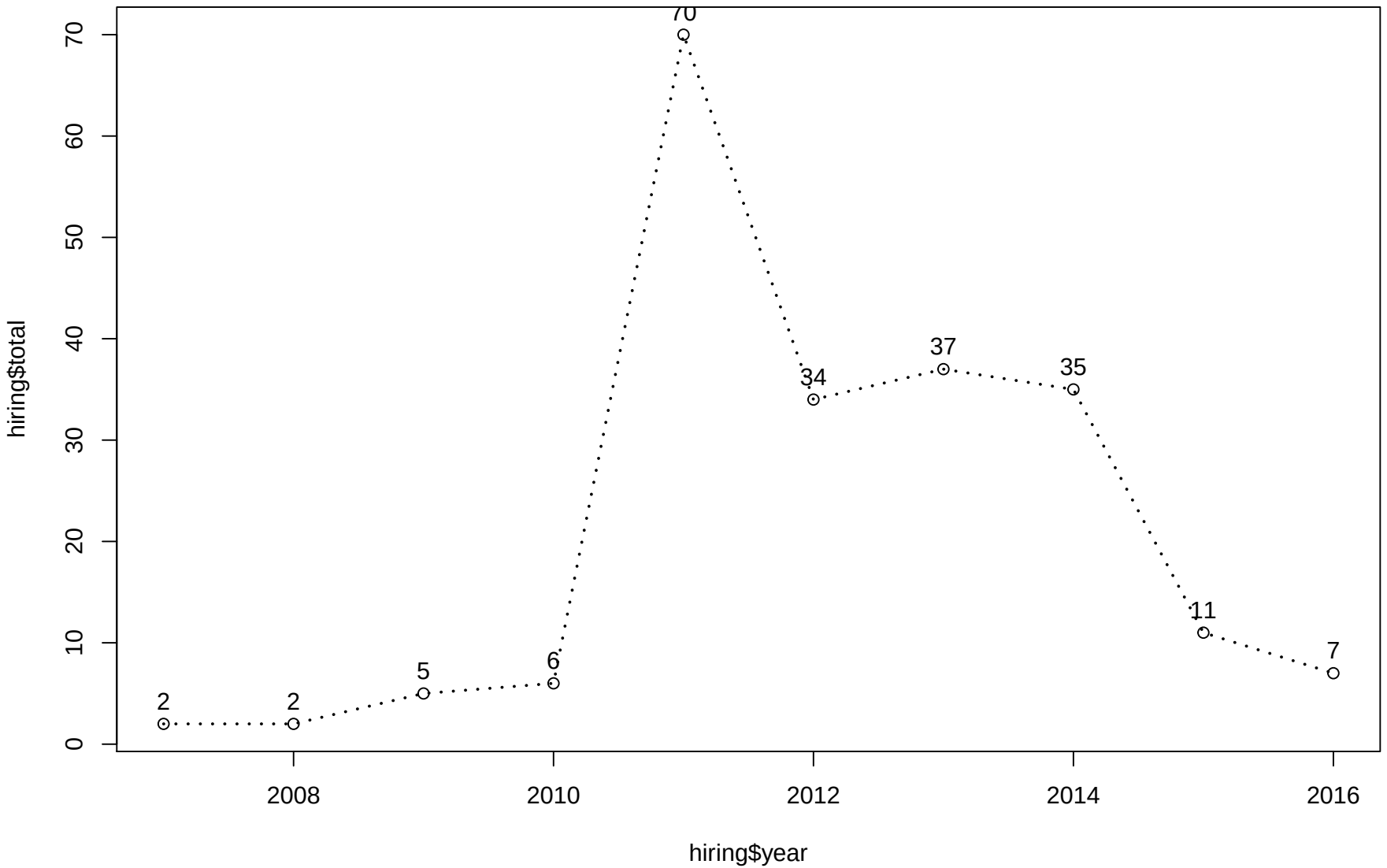
Es posible superponer varias visualizaciones en un mismo gráfico.

```
plot(hiring$year, hiring$total)
lines(hiring$year, hiring$total,
      lty=3, lwd=2)
```


hiring$total
0
10
20
30
40
50
60
70
2008
2010
2012
2014
2016
hiring$year

Así como incluir etiquetas para cada uno de los puntos.

```
plot(hiring$year, hiring$total)
lines(hiring$year, hiring$total,
      lty=3, lwd=2)
text(hiring$year, hiring$total, labels=hiring$total, pos=3)
```


hiring$total
0
10
20
30
40
50
60
70
2
2
5
6
70
34
37
35
11
7
2008
2010
2012
2014
2016
hiring$year

# Exportar una imagen

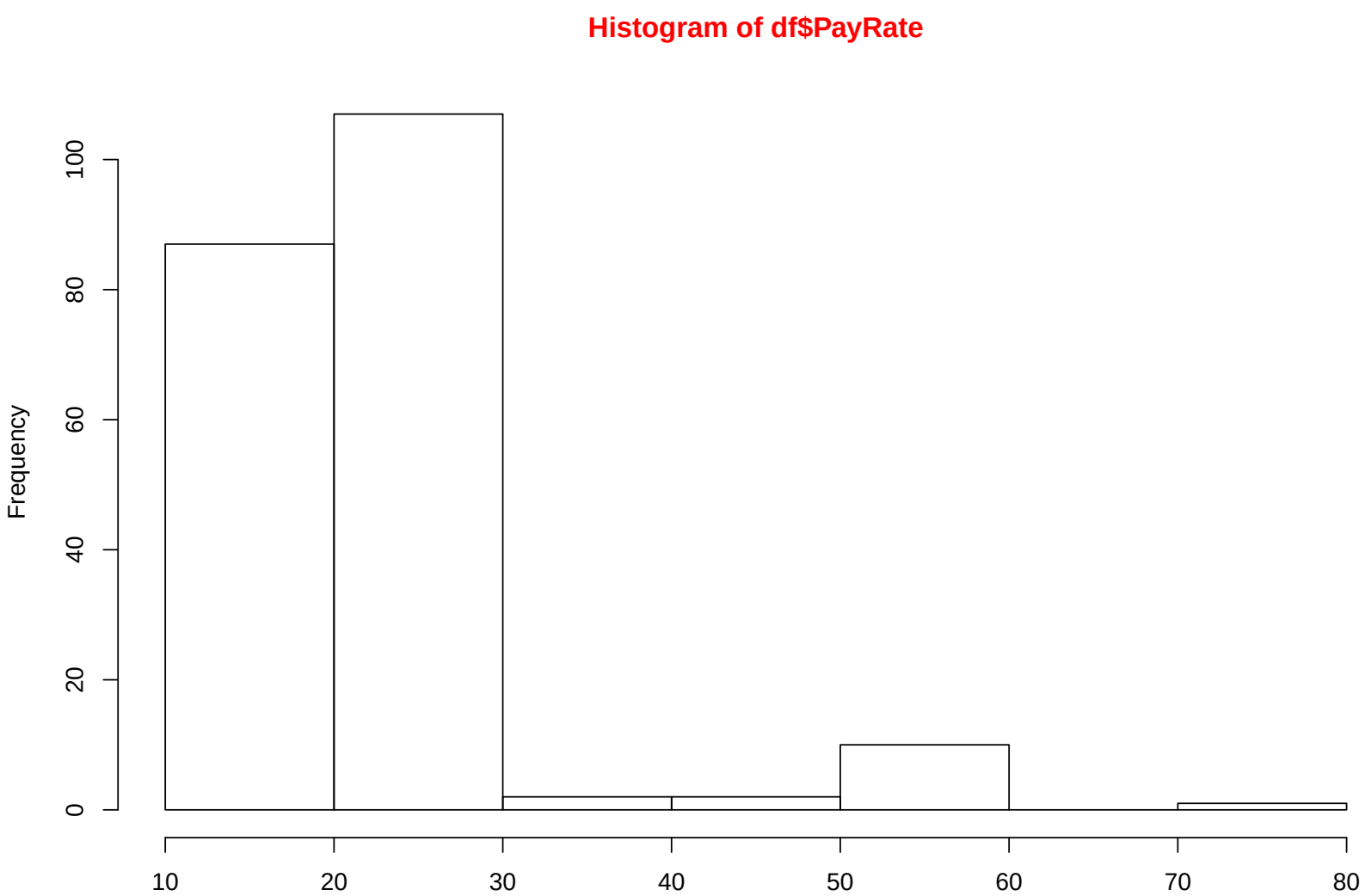
Es posible exportar una imagen a distintos formatos (png, jpg, tiff, bmp, pdf...)

```
png(file = here("img","myplot.png"), bg = "transparent")
plot(hiring$year, hiring$total)
lines(hiring$year, hiring$total,
      lty=3, lwd=2)
text(hiring$year, hiring$total, labels=hiring$total, pos=3)
dev.off()
```

# Configuración por defecto

La función `par()` permite modificar la configuración que se aplica por defecto a cada gráfica.

```r
old_par <- par()
par(col.main="red")
hist(df$PayRate)
```



# Recursos y material

- Colores en R
- Párametros gráficos
- Gráficas más habituales