graphing functions

This section covers the following graphing functions:

1. plot()
2. hist()
3. barplot()
4. xyplot()
5. dotchart()
6. boxplot()

1. `plot()`

◆ Purpose

The `plot()` function is a **generic function** used for producing a wide variety of plots in base R. It adapts its behavior based on the type of object passed (e.g., numeric, formula, factor, etc.).

📦 Package

Base R (`graphics` package)

📃 Function Header

```
plot(x, y = NULL, type = "p", xlim = NULL, ylim = NULL, log = "",
     main = NULL, sub = NULL, xlab = NULL, ylab = NULL, asp = NA, ...)
```

🔧 Parameters

Argument	Description	Accepted Values / Data Types
x	Data for x-axis	Numeric, factor, object
y	Data for y-axis (optional)	Numeric, NULL

Argument	Description	Accepted Values / Data Types
type	Type of plot	"p" (points), "l" (lines), "b" (both), "c" (lines part of "b"), "o" (overplotted), "h" (histogram-like lines), "s" (stair steps), "S" (other stair steps), "n" (no plotting)
xlim / ylim	Axis limits	Numeric vector of length 2
log	Axis scaling	"", "x", "y", "xy"
main, sub	Main and subtitle	Character strings
xlab, ylab	Axis labels	Character strings
asp	Aspect ratio	Numeric
...	Additional graphical parameters	See below

🎨 Graphical Parameters
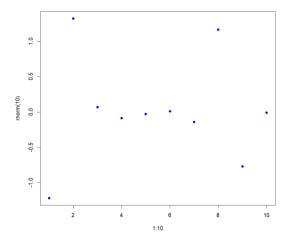
Parameter	Description	Accepted Values
pch	Plotting symbol	Integer or character (e.g., 16, "+", "o")
col	Color	Color names, hex codes
cex	Size of points	Numeric
lty	Line type	0, 1, 2, 3, 4, 5, 6
lwd	Line width	Numeric
bg	Background color for open symbols	Color
xaxs, yaxs	Axis interval expansion	"r" (default), "i"

💡 Example Use Cases
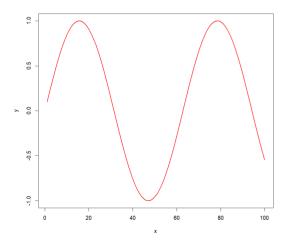
➤ Simple Scatter Plot

```
plot(1:10, rnorm(10), type = "p", col = "blue", pch = 16)
```

➤ Line Plot

```r
x <- 1:100
y <- sin(x / 10)
plot(x, y, type = "l", col = "red", lwd = 2)
```
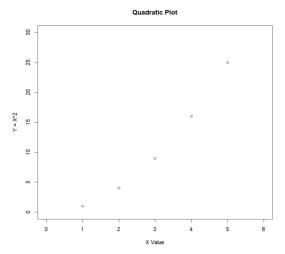

➤ Custom Axes and Titles

```r
plot(1:5, (1:5)^2, xlim = c(0, 6), ylim = c(0, 30),
     main = "Quadratic Plot", xlab = "X Value", ylab = "Y = X^2")
```

2. `hist()`

◆ Purpose

The `hist()` function is used to produce a **histogram**, a graphical representation of the distribution of a numeric dataset by splitting the data into bins and counting the frequency of values in each bin.

📦 Package

Base R (`graphics` package)

📄 Function Header

```r
hist(x, breaks = "Sturges", freq = NULL, probability = !freq,
     include.lowest = TRUE, right = TRUE, density = NULL, angle = 45,
     col = NULL, border = par("fg"), main = paste("Histogram of", xname),
     xlim = range(breaks), ylim = NULL, xlab = xname, ylab,
     axes = TRUE, plot = TRUE, labels = FALSE, ...)
```

🔧 Parameters

Argument	Description	Accepted Values / Data Types
x	Numeric data vector	Numeric

Argument	Description	Accepted Values / Data Types
`breaks`	Bin boundaries or bin calculation method	Numeric vector, or `"Sturges"`, `"Scott"`, `"FD"`, `"Freedman-Diaconis"`, or an integer (number of bins)
`freq`	If `TRUE`, histogram counts; if `FALSE`, densities	Logical
`probability`	Same as `!freq`	Logical
`include.lowest`	Include lowest value in the first bin	Logical
`right`	If `TRUE`, intervals closed on the right	Logical
`density`	Shading line density in lines per inch	Numeric or `NULL`
`angle`	Angle of shading lines	Numeric (degrees)
`col`	Fill color for bars	Color names or hex codes
`border`	Bar border color	Color name or `NA`
`main`, `xlab`, `ylab`	Titles and labels	Character strings
`xlim`, `ylim`	Axis limits	Numeric vectors of length 2
`axes`	Draw axes?	Logical
`plot`	Plot the histogram?	Logical
`labels`	Show bin counts on bars	Logical or character vector
`...`	Additional graphical parameters	See below

🎨 Graphical Parameters
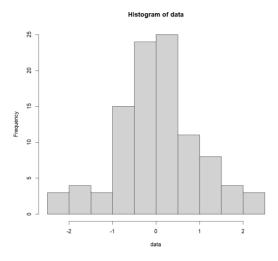
Parameter	Description	Accepted Values
`xpd`	Clipping	Logical
`cex`, `cex.axis`, `cex.lab`, `cex.main`, `cex.sub`	Text and symbol scaling	Numeric
`font`, `font.axis`, `font.lab`, `font.main`, `font.sub`	Font styles	`1` (plain), `2` (bold), `3` (italic), `4` (bold italic)
`col.axis`, `col.lab`, `col.main`, `col.sub`	Text colors	Color names or hex codes
`las`	Axis label orientation	`0` (parallel), `1` (horizontal), `2` (perpendicular), `3` (vertical)

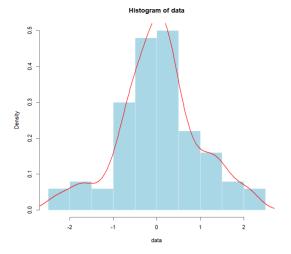## 💡 Example Use Cases

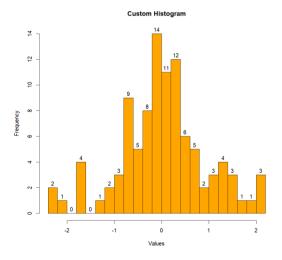### ➤ Simple Histogram

```
data <- rnorm(100)
hist(data)
```


➤ Histogram with Density

```
hist(data, probability = TRUE, col = "lightblue", border = "white")
lines(density(data), col = "red", lwd = 2)
```
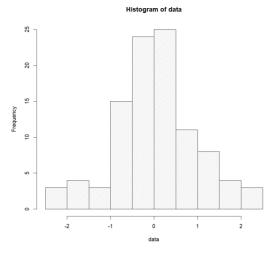
Histogram of data

Histogram of data

➤ Custom Breaks and Labels

```
hist(data, breaks = 20, col = "orange", main = "Custom Histogram", xlab = "Values",
labels = TRUE)
```


Custom Histogram

➤ Histogram with Shaded Lines

```
hist(data, density = 20, angle = 45, col = "gray", border = "black")
```

3. `barplot()`

◆ Purpose

The `barplot()` function is used to create **bar charts** in base R. It can display both **vertical and horizontal bars**, grouped or stacked, from a vector or matrix of values.

📦 Package

Base R (`graphics` package)

📄 Function Header

```
barplot(height, width = 1, space = NULL, names.arg = NULL,
        legend.text = NULL, beside = FALSE, horiz = FALSE,
        density = NULL, angle = 45, col = NULL, border = par("fg"),
        main = NULL, sub = NULL, xlab = NULL, ylab = NULL,
        xlim = NULL, ylim = NULL, xpd = TRUE, log = "",
        axes = TRUE, axisnames = TRUE, cex.axis = par("cex.axis"),
        cex.names = par("cex.axis"), inside = TRUE, plot = TRUE,
        axis.lty = 0, offset = 0, add = FALSE, args.legend = NULL, ...)
```

🔧 Parameters

Argument	Description	Accepted Values / Data Types
`height`	Vector or matrix of bar heights	Numeric
`width`	Width of bars	Numeric
`space`	Space between bars/groups	Numeric or vector
`names.arg`	Names for each bar	Character vector
`legend.text`	Labels for legend	Logical or character vector
`beside`	Plot bars beside (not stacked)	Logical
`horiz`	Horizontal bars	Logical
`density`	Shading lines density	Numeric
`angle`	Angle of shading lines	Numeric (degrees)
`col`	Fill colors	Color names or hex codes
`border`	Border color for bars	Color name
`main`, `sub`, `xlab`, `ylab`	Titles and labels	Character strings
`xlim`, `ylim`	Axis limits	Numeric vector of length 2
`xpd`	Allow plotting outside plot region	Logical
`log`	Log axes	`"x"`, `"y"`, `"xy"`, or `""`
`axes`	Draw axes?	Logical
`axisnames`	Show names on axis?	Logical
`cex.axis`, `cex.names`	Scaling for axis and names	Numeric
`inside`	Legend inside plot?	Logical
`plot`	Whether to draw the plot	Logical
`axis.lty`	Line type for axis	Integer (see `lty` values)
`offset`	Offsets for bars	Numeric vector
`add`	Add to existing plot	Logical
`args.legend`	List of legend parameters	List
`...`	Additional graphical parameters	See below

Note: legend = TRUE automatically generates the legend in the top-right
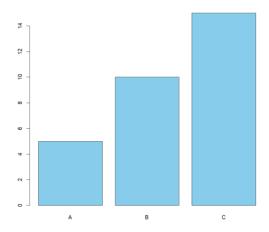
🎨 Graphical Parameters

Parameter	Description	Accepted Values
`lty`	Line type for borders	`0`, `1`, `2`, `3`, `4`, `5`, `6`
`lwd`	Line width for borders	Numeric

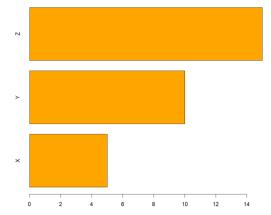| Parameter	Description	Accepted Values
`cex`, `font`, `col.axis`, `col.lab`, `col.main`, `col.sub`	Styling parameters	Various (font type, color, size)
`las`	Axis label orientation	`0`, `1`, `2`, `3`

💡 Example Use Cases

➤ Simple Vertical Barplot

```
values <- c(5, 10, 15)
barplot(values, col = "skyblue", names.arg = c("A", "B", "C"))
```
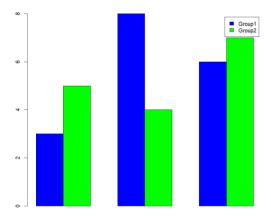

➤ Horizontal Barplot

```
barplot(values, horiz = TRUE, col = "orange", names.arg = c("X", "Y", "Z"))
```
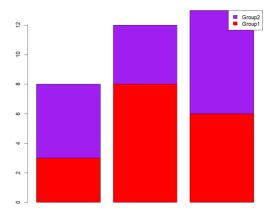
➤ Grouped Barplot from Matrix

```
m <- matrix(c(3, 5, 8, 4, 6, 7), nrow = 2)
barplot(m, beside = TRUE, col = c("blue", "green"), legend.text = c("Group1",
"Group2"))
```


➤ Stacked Barplot with Legend

```
barplot(m, beside = FALSE, col = c("red", "purple"),
        legend.text = c("Group1", "Group2"), args.legend = list(x = "topright"))
```

4. `xyplot()`

Note: dotplot(x) calls xyplot(x) if x is a formula and lattice package has been loaded

◆ Purpose

`xyplot()` is the **main high-level plotting function** in the **lattice package**, used to create **scatterplots and conditioned plots** (i.e., plots split by factors). It handles complex visualizations like multi-panel plots with formulas and grouping.

📦 Package

lattice

> To use:

```
> library(lattice)
> ```
📄 Function Header
```r
xyplot(formula, data, allow.multiple = is.null(groups) || outer,
       outer = !is.null(groups), auto.key = FALSE,
       aspect = "fill", panel = lattice.getOption("panel.xyplot"),
       prepanel = NULL, scales = list(), strip = TRUE, groups = NULL,
       xlab, xlim, ylab, ylim, drop.unused.levels =
```

```
lattice.getOption("drop.unused.levels"),
        subscripts = !is.null(groups), subset, ...)
```

🔧 Parameters

Argument	Description	Accepted Values / Data Types	
formula	Formula describing the plot	Typically `y ~ x` or `y ~ x`	factor`
data	Data frame used in formula	Data frame	
allow.multiple	Allow multiple y variables	Logical	
outer	Draw separate panels for groups	Logical	
auto.key	Automatically draw legend	Logical or list of legend parameters	
aspect	Aspect ratio	`"fill"`, `"iso"` (1:1), numeric	
panel	Panel function used to draw each panel	Function (`panel.xyplot` by default)	
prepanel	Function to precompute limits	Function	
scales	Axis configuration	List (e.g., `list(x = list(log = TRUE))`)	
strip	Panel strip function	Logical or function	
groups	Grouping variable for colors/symbols	Factor or categorical vector	
xlab, ylab	Axis labels	Character	
xlim, ylim	Axis limits	Numeric vector of length 2	
drop.unused.levels	Drop unused factor levels	Logical	
subscripts	Pass subscripts to panel	Logical	
subset	Logical expression to filter data	Logical expression	
...	Additional arguments passed to lattice or panel functions	See graphical params	

🎨 Graphical Parameters via `par.settings` or `panel.xyplot()`

Parameter	Description	Accepted Values
col, col.symbol	Color of points	Color names or hex codes
pch	Plotting character/symbol	Integer or single-char string
cex	Size of points	Numeric

Parameter	Description	Accepted Values
lty, lwd	Line type, line width	Integer for `lty`; numeric for `lwd`
type	Plot type	`"p"` (points), `"l"` (lines), `"o"` (overplotted), `"b"` (both), `"g"` (grid), `"r"` (regression), etc.
main, sub	Titles	Character
par.settings	Theme settings	List (see `trellis.par.get()`)

💡 Example Use Cases

```
> head(mtcars)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
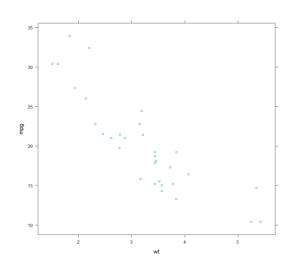Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
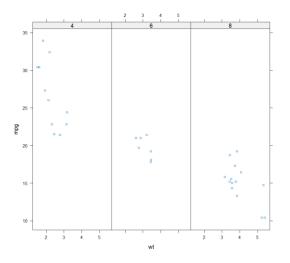Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

➤ Basic Scatterplot

```
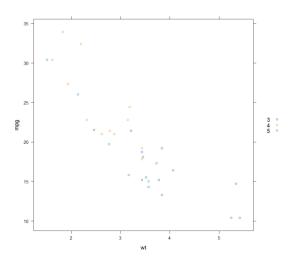library(lattice)
xyplot(mpg ~ wt, data = mtcars)
```


➤ Scatterplot Conditioned on Factor

```
xyplot(mpg ~ wt | factor(cyl), data = mtcars, layout = c(3, 1))
```



```
panel.xyplot(x, y, ...)
panel.abline(lm(y ~ x), col = "red", lwd = 2)
})
```


➤ Grouped Scatterplot with Auto Legend

```
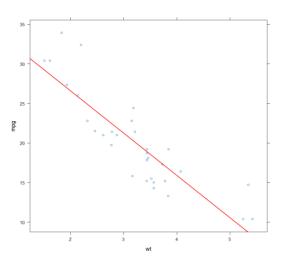xyplot(mpg ~ wt, data = mtcars, groups = gear, auto.key = TRUE)
```


➤ Customized Axes and Style

```
xyplot(mpg ~ wt, data = mtcars,
       scales = list(x = list(log = TRUE), y = list(relation = "free")),
       par.settings = list(superpose.symbol = list(pch = 19, col = "darkblue")))
```


➤ Custom Panel Function with Regression Line
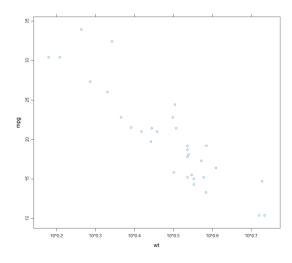
```
xyplot(mpg ~ wt, data = mtcars,
       panel = function(x, y, ...) {
```

5. `dotchart()`

Note: dotplot(x) calls dotchart(x) if x is a numeric vector

◆ Purpose

The `dotchart()` function creates a **Cleveland-style dot chart**, useful for visualizing **labeled numeric values** (e.g., means, rates, or frequencies). It provides an effective way to compare small to medium numbers of values.

📦 Package

Base R (`graphics` package)

📋 Function Header

```
dotchart(x, labels = NULL, groups = NULL, gcolor = par("fg"),
         color = par("fg"), cex = par("cex"), pch = 21, bg = par("bg"),
         xlim = range(x[is.finite(x)]), main = NULL, xlab = NULL, ...)
```

🔧 Parameters

Argument	Description	Accepted Values / Data Types
x	Numeric vector of values	Numeric
labels	Labels for each value	Character vector
groups	Grouping variable (defines color groups and spacing)	Factor or categorical vector
gcolor	Color of group labels	Color name or hex code
color	Dot color	Color name or hex code
cex	Size of dots	Numeric
pch	Plotting character (symbol)	Integer (0–25), or character (e.g., `"o"`)
bg	Background color for filled symbols (`pch` 21–25)	Color
xlim	X-axis limits	Numeric vector of length 2
main, xlab	Titles and labels	Character strings
...	Additional graphical parameters	See below

🎨 Graphical Parameters
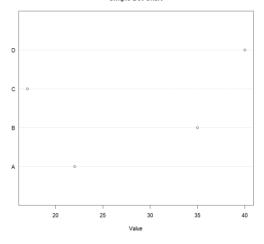
Parameter	Description	Accepted Values
`col.main`, `col.lab`, `col.axis`	Title and label colors	Color names or hex codes
`cex.axis`, `cex.lab`, `cex.main`, `cex.sub`	Scaling for text and axes	Numeric
`font`, `font.axis`, `font.lab`, `font.main`	Font style	1 (plain), 2 (bold), 3 (italic), 4 (bold italic)
`las`	Axis label orientation	0, 1, 2, 3
`mar`	Margins around plot	Numeric vector of length 4

💡 Example Use Cases

➤ Simple Dotchart

```
x <- c(22, 35, 17, 40)
names(x) <- c("A", "B", "C", "D")
dotchart(x, main = "Simple Dot Chart", xlab = "Value")
```


Simple Dot Chart

➤ Dotchart with Groups

```
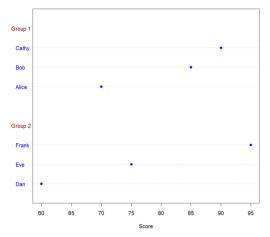scores <- c(70, 85, 90, 60, 75, 95)
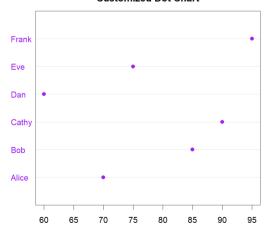names(scores) <- c("Alice", "Bob", "Cathy", "Dan", "Eve", "Frank")

# Convert groups to a factor
groups <- factor(c("Group 1", "Group 1", "Group 1", "Group 2", "Group 2", "Group 2"))
```

```
dotchart(scores, groups = groups, color = "blue", gcolor = "darkred",
         pch = 19, main = "Grouped Dot Chart", xlab = "Score")
```


Grouped Dot Chart

➤ Custom Symbols and Size

```
dotchart(scores, pch = 16, cex = 1.5, color = "purple", bg = "lightgray",
         main = "Customized Dot Chart")
```


Customized Dot Chart

**6. boxplot()

◆ Purpose

Creates box-and-whisker plots to visualize the distribution, median, and spread of a numeric variable or compare across groups.

📦 Package

`graphics` (Base R)

📄 Function Header

```
boxplot(x, ..., range = 1.5, width = NULL, varwidth = FALSE,
        notch = FALSE, outline = TRUE, names, plot = TRUE,
        border = par("fg"), col = NULL, log = "",
        pars = NULL, horizontal = FALSE, add = FALSE,
        at = NULL)
```

🔧 **Parameter

Argument	Description	Accepted Values
x	A formula, numeric vector, list, or data frame	e.g. `mpg ~ cyl`, `list(A=1:10, B=11:20)`
...	Additional data vectors or formulas	Depends on context
range	Controls whisker length as a multiple of IQR	`numeric`, default `1.5`
width	Widths of boxes	`numeric vector` or `NULL`
varwidth	Whether box width is proportional to sample size	`TRUE`, `FALSE`
notch	Draws a notch to indicate confidence interval for median	`TRUE`, `FALSE`
outline	Plot outliers as individual points	`TRUE`, `FALSE`
names	Labels for the boxes	`character vector`
plot	If `FALSE`, computes values but doesn't plot	`TRUE`, `FALSE`
border	Color for box borders	Any color name or code
col	Fill color for boxes	Same as `border`
log	Log scale axes	`""`, `"x"`, `"y"`, `"xy"`
pars	List of additional graphical parameters passed to `bxp()`	See graphical parameters below
horizontal	Orientation	`TRUE`, `FALSE`
add	Add to existing plot	`TRUE`, `FALSE`

Argument	Description	Accepted Values
`at`	Horizontal positions of boxes	`numeric vector`
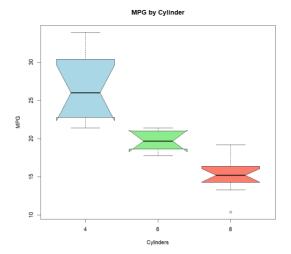
🎨 **Graphical Parameters

Passed to `bxp()` function:

Parameter	Description	Accepted Values
`boxwex`	Box width expansion factor	`numeric`
`staplewex`	Width of staples	`numeric`
`outwex`	Width of outlier points	`numeric`
`pch`	Point symbol	$0 - 25$ or symbol characters
`cex`	Character expansion factor	`numeric`
`col`	Color of fill (for box)	Any valid R color
`border`	Color of border	Any valid R color
`outcol`	Color of outliers	Any valid R color
`medcol`	Color of median line	Any valid R color
`medlty`	Line type for median	$0 - 6$, e.g., 1 = solid, 2 = dashed
`medlwd`	Line width for median	`numeric`

💡 Example Use Cases

➤ Simple boxplot

```
data(mtcars)
boxplot(mpg ~ cyl, data = mtcars, col = c("lightblue", "lightgreen", "salmon"),
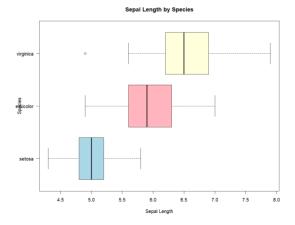        main = "MPG by Cylinder", xlab = "Cylinders", ylab = "MPG", notch = TRUE)
```


MPG by Cylinder

➤ Horizontal Boxplots with Custom Axis and Colors

```
data(iris)
boxplot(Sepal.Length ~ Species, data = iris,
        horizontal = TRUE,
        col = c("lightblue", "lightpink", "lightyellow"),
        main = "Sepal Length by Species",
        xlab = "Sepal Length",
        las = 1)
```


Sepal Length by Species

➤ Using Notches, Variable Widths, and Custom Outliers

```
data(PlantGrowth)
boxplot(weight ~ group, data = PlantGrowth,
        notch = TRUE,
        varwidth = TRUE,
        col = "skyblue",
        pch = 17,
        outcol = "red",
        main = "Plant Growth Study",
        xlab = "Group", ylab = "Weight")
```