# **Data Regression Functions & Concepts**

This section covers the following functions:

- 1. pairs()
- 2. density()
- 3. lm()
- 4. coef()
- 5. cut()
- 6. predict()
- 7. PRESS()

# 1. pairs()

#### Purpose

Displays **scatterplot matrices** of numeric data, useful for exploring relationships between multiple variables.

# Package

Base R (graphics package)

#### Function Header

```
pairs(x, labels = NULL, panel = points, ..., lower.panel = panel,
    upper.panel = panel, diag.panel = NULL, text.panel = textPanel,
    label.pos = 0.5 + has.upper + 1 * has.lower,
    cex.labels = NULL, font.labels = 1,
    rowlattop = TRUE, gap = 1)
```

#### Parameters

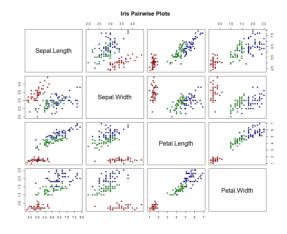
Argument	Description	Accepted Values / Data Types
х	Matrix or data frame	Numeric data frame or matrix
labels	Labels for variables	Character vector
panel	Function to draw panels	<pre>Function (e.g., points, panel.smooth)</pre>
lower.panel, upper.panel	Panel functions for lower/upper triangles	Function
diag.panel	Function for diagonal panels	Function
text.panel	Function for text panels	Function
label.pos	Position of labels	Numeric
cex.labels	Label size	Numeric
font.labels	Label font type	Integer (1: plain, 2: bold, etc.)
row1attop	Row order	Logical
gap	Gap between plots	Numeric

### Graphical Parameters

Same as par() base graphical parameters (col, pch, cex, etc.).

# **Parample Use Case**

```
pairs(iris[1:4], main = "Iris Pairwise Plots", pch = 21, bg = c("red", "green3",
   "blue")[iris$Species])
```



# 2. density()

#### Purpose

Computes kernel density estimates of a numeric vector to visualize its distribution.

# Package

Base R (stats package)

#### Function Header

```
density(x, bw = "nrd0", adjust = 1, kernel = "gaussian",
    window = kernel, width, give.Rkern = FALSE,
    n = 512, from, to, cut = 3, na.rm = FALSE, ...)
```

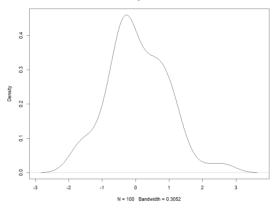
#### Parameters

Argument	Description	Accepted Values / Data Types	
х	Numeric vector of values	Numeric	
bw	Bandwidth method	"nrd0", "nrd", "ucv", "bcv", "SJ-ste", "SJ-dpi", or numeric	
adjust	Bandwidth adjustment	Numeric	
kernel	Kernel function	"gaussian", "epanechnikov", "rectangular", "triangular", "biweight", "cosine", "optcosine"	
n	Number of grid points	Integer	
from, to	Range for density	Numeric	
cut	Extend range	Numeric	
na.rm	Remove NAs	Logical	

# **Parample Use Case**

```
x <- rnorm(100)
d <- density(x)
print(d)
## Call:
## density.default(x = x)
##
## Data: x (100 obs.); Bandwidth 'bw' = 0.3052
##
## x y
## Min. :-2.9114 Min. :0.0001497</pre>
```

#### Kernel Density of Random Normals



# 3. lm()

### Purpose

Fits linear models using least squares, commonly used for regression analysis.

# Package

Base R (stats package)

#### Function Header

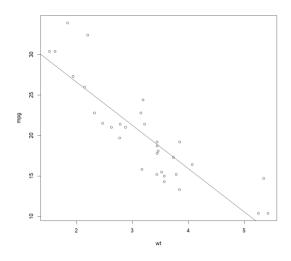
```
lm(formula, data, subset, weights, na.action,
  method = "qr", model = TRUE, x = FALSE, y = FALSE, qr = TRUE, singular.ok = TRUE,
contrasts = NULL, offset, ...)
```

#### Parameters

Argument	Description	Accepted Values / Data Types
formula	Regression formula	y ~ x1 + x2
data	Data frame	Data frame
subset	Optional subset	Logical or index
weights	Prior weights	Numeric vector
na.action	Missing data handling	na.omit, na.exclude, etc.
method	Fitting method	"qr"
model, x, y, qr	Return components	Logical
singular.ok	Allow singular fits	Logical
contrasts	Factor contrasts	List
offset	Offset to linear predictor	Numeric

# Example Use Case

```
fit <- lm(mpg ~ wt, data = mtcars)
plot(mg~wt, data=mtcars)
abline(fit)</pre>
```



# 4. coef()

### Purpose

Extracts the **coefficients** from a fitted model (e.g., from lm() or glm()).

# Package

Base R (stats package)

#### **Function Header**

```
coef(object, ...)
```

### Parameters

Argument	Description	Accepted Values / Data Types
object	A fitted model	Model object (e.g., result of lm())
	Additional arguments	Depends on method

# **Parample Use Case**

```
model <- lm(mpg ~ wt, data = mtcars)
coef(model) # Returns intercept and slope
## (Intercept) wt
## 37.285126 -5.344472</pre>
```

# 5. cut()

#### Purpose

Divides a numeric variable into factor intervals (bins), useful for categorizing continuous data.

# Package

Base R (base package)

#### **Function Header**

```
cut(x, breaks, labels = NULL, include.lowest = FALSE, right = TRUE,
    dig.lab = 3, ordered_result = FALSE, ...)
```

#### Parameters

Argument	Description	Accepted Values / Data Types
х	Numeric vector	Numeric
breaks	Number or vector of break points	Integer or numeric vector
labels	Labels for bins	Character vector
include.lowest	Include lowest endpoint	Logical
right	Intervals closed on right?	Logical
dig.lab	Digits in labels	Integer
ordered_result	Return ordered factor?	Logical

# 💡 Example Use Case

```
cut(mtcars$mpg, breaks = 3, labels = c("Low", "Medium", "High"))
## [1] Medium Medium Medium
## [4] Medium Medium Low
## [7] Low Medium Medium
## [10] Medium Low
                  Low
## [13] Low Low
## [16] Low Low
                  High
## [19] High High Medium
## [22] Low
             Low
                  Low
## [25] Medium High Medium
## [28] High Low
                   Medium
## [31] Low Medium
## Levels: Low Medium High
```

### 6. predict()

#### Purpose

Generates predicted values from model objects, such as those returned by lm() or glm().

# Package

Base R (stats package)

#### Function Header

#### Parameters

Argument	Description	Accepted Values / Data Types
object	A fitted model	Model object
newdata	New data for prediction	Data frame
se.fit	Return standard errors?	Logical
interval	Type of interval	"none", "confidence", "prediction"
level	Confidence level	Numeric (0–1)
type	Type of prediction	"response", "terms"
na.action	NA handling	na.pass, na.omit, etc.

# 💡 Example Use Case

```
model <- lm(mpg ~ wt + hp, data = mtcars)
predict(model, newdata = data.frame(wt = 3, hp = 120), interval = "confidence")
## fit lwr upr
## 1 21.78102 20.77178 22.79027</pre>
```

# 7. PRESS()`

#### Purpose

Calculates the **Prediction Sum of Squares (PRESS)** statistic, a diagnostic measure for assessing the **predictive power** of a regression model. It is the sum of squares of prediction errors obtained using **leave-one-out cross-validation**.

# Package

MPV

To use:

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

```
### **  Function Header**
'''r
PRESS(model)
```

#### Parameters

Parameter Description		Acceptable Values
model	A fitted model object from functions like $lm()$ or $glm()$	Object of class "lm" or "glm"

**Note:** The model must be a linear or generalized linear model.

# Output

Returns a named list containing:

PRESS: The PRESS statistic (numeric)

#### Interpretation:

Lower PRESS values indicate better predictive ability of the model. Comparing PRESS across models can help select the model with superior prediction accuracy.

# Example Use Case

```
model <- lm(mpg ~ wt + hp, data = mtcars)
PRESS(model) # Returns PRESS statistic
## [1] 246.5063</pre>
```