Time Functions

`This section covers the following functions:`

1. strptime()
2. chron()
3. ts()

---

1. `strptime()`

🔹 Purpose

The `strptime()` function is used to **parse character representations of dates/times into POSIXlt objects**, which can be used for date-time calculations, plotting, or formatting in R.

📦 Package

Base R (`base` package)

🧾 Function Header

```
strptime(x, format, tz = "")
```

🔧 Parameters

Argument	Description	Accepted Values / Data Types
`x`	Character vector of date-time strings	Character
`format`	Format to match against elements in `x`	Character format string using `%` directives
`tz`	Time zone specification	Character (e.g., `"UTC"`, `"America/New_York"`)

📌 Common Format Codes

Directive	Meaning	Example
`%Y`	4-digit year	`2025`
`%y`	2-digit year	`25`
`%m`	2-digit month	`04`
`%d`	Day of the month	`15`
`%H`	Hour (00–23)	`13`
`%M`	Minute	`45`
`%S`	Second	`30`
`%B`	Full month name	`April`
`%a`	Abbreviated weekday	`Tue`

💡 Example Use Cases

➤ Parse simple date-time string

```
strptime("2025-04-15 14:30", format = "%Y-%m-%d %H:%M")
## [1] "2025-04-15 14:30:00 PDT"
```

---

2. `chron()`

🔹 Purpose

The `chron()` function creates date-time objects of class `"chron"` for **dates and/or times without time zones**. It is simpler and more lightweight than `POSIXct`.

📦 Package

`chron` package (must be installed and loaded)

🧾 Function Header

```
chron(dates = NULL, times = NULL, format = c(dates = "m/d/y", times = "h:m:s"))
```

🔧 Parameters

Argument	Description	Accepted Values / Data Types
`dates`	Character vector of dates	Format: `"mm/dd/yy"` or specified via `format`
`times`	Character vector of times	Format: `"hh:mm:ss"` or specified via `format`

Argument	Description	Accepted Values / Data Types
`format`	List specifying formats for `dates` and `times`	Named vector or list (e.g., `c(dates = "d/m/y")`)

💡 Example Use Cases

➤ Basic chron date-time object

```
library(chron)
chron(dates = "04/15/25", times = "14:30:00")
## [1] (04/15/25 14:30:00)
```

➤ Custom format for date

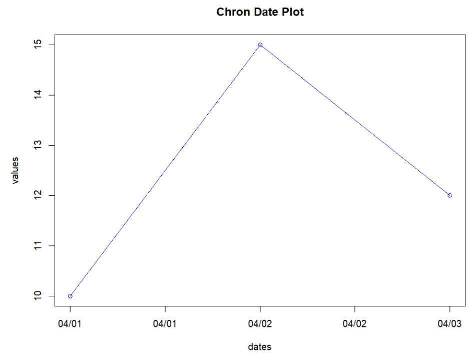
```
chron(dates = "15-04-25", format = c(dates = "d-m-y"))
## [1] 15-04-25
```

➤ Perform calculations

```
d1 <- chron("04/10/25")
d2 <- chron("04/15/25")
d2 - d1  # 5 days difference
## Time in days:
## [1] 5
```

➤ Plotting usage

```
dates <- chron(c("04/01/25", "04/02/25", "04/03/25"))
values <- c(10, 15, 12)
plot(dates, values, type = "o", col = "blue", main = "Chron Date Plot")
```



---

3. `ts()`

🔹 Purpose

The `ts()` function is used to **create time-series objects** from numeric data, adding temporal structure such as start time, frequency, and end time.

📦 Package

Base R (`stats` package)

🧾 Function Header

```
ts(data = NA, start = 1, end = numeric(), frequency = 1, deltat = 1, ts.eps =
getOption("ts.eps"), class = NULL)
```

🔧 Parameters

Argument	Description	Accepted Values / Data Types
`data`	Numeric vector or matrix of data	Numeric
`start`	Time of first observation	Single number or vector (e.g., `c(2025, 1)`)
`end`	Optional time of last observation	Same format as `start`
`frequency`	Number of observations per unit time	Numeric (e.g., `12` = monthly, `4` = quarterly)

Argument	Description	Accepted Values / Data Types
`deltat`	Time interval between observations (reciprocal of `frequency`)	Numeric
`ts.eps`	Numerical fuzz factor	Numeric
`class`	Class of the returned object	`"ts"` (default) or `NULL`

💡 Example Use Cases

➤ Create a simple yearly time series

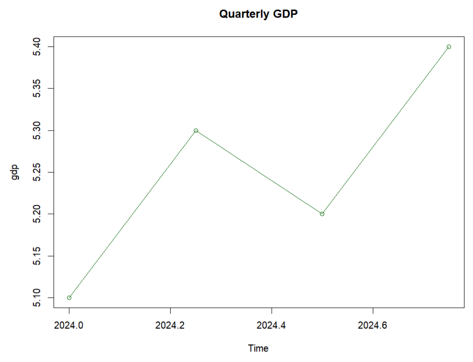
```
ts(c(2, 4, 5, 3), start = 2022, frequency = 1)
## Time Series:
## Start = 2022
## End = 2025
## Frequency = 1
## [1] 2 4 5 3
```

➤ Monthly time series starting in Jan 2023

```
sales <- ts(c(100, 120, 130, 110), start = c(2023, 1), frequency = 12)
print(sales)
##      Jan Feb Mar Apr
## 2023 100 120 130 110
```

➤ Plot quarterly data

```
gdp <- ts(c(5.1, 5.3, 5.2, 5.4), start = c(2024, 1), frequency = 4)
plot(gdp, type = "o", col = "darkgreen", main = "Quarterly GDP")
```


Quarterly GDP
gdp
Time