

Package ‘simPDF’

July 30, 2026

Type Package

Title Fast Multi-Page PDF Report Layout on the Graphics Device

Version 0.1.1

Date 2026-07-21

Maintainer Kyun-Seop Bae <k@acr.kr>

Description A lightweight, dependency-free engine to build multi-page PDF reports quickly on top of R's built-in graphics device ('pdf'/'cairo_pdf'). Content is placed by a measured flow layout: every text block reports its real width and height via 'strwidth'/'strheight', the vertical cursor advances by measured height, and pages break automatically. This eliminates the text-overlap of dead-reckoned coordinate reports (such as the 'nmw' NONMEM diagnostic reports) and replaces slow '.Rmd'/'knitr'/'LaTeX' pipelines for fixed report generation. Interactive AcroForm CRFs are out of scope and handled by the sibling 'pdfCRF' package.

Depends R (>= 3.6.0)

Imports grDevices, graphics

Suggests tinytest, png, jpeg, knitr, rmarkdown

VignetteBuilder knitr

License GPL-3

Encoding UTF-8

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Kyun-Seop Bae [aut, cre]

Repository CRAN

Date/Publication 2026-07-30 16:50:02 UTC

Contents

block_authorship	2
block_flow_diagram	3

block_image	4
block_keep	5
block_matrix	5
block_para	6
block_plot	7
block_pre	8
block_rule	8
block_signature	9
block_spacer	10
block_table	10
flow_add	11
flow_run	12
frame_set	12
remaining	13
running	13
sp_add_sig_fields	14
sp_close	14
sp_figure_page	15
sp_height	15
sp_new	16
sp_overlaps	17
sp_page	17
sp_trace	18
sp_width	18
Index	19

block_authorship	<i>An authorship block (prepared-by line + metadata)</i>
------------------	--

Description

A compact block naming who prepared the document and when, plus optional affiliation and free-form lines. Purely visible text.

Usage

```
block_authorship(  
  author,  
  role = "Prepared by",  
  affiliation = NULL,  
  date = NULL,  
  extra = NULL,  
  size = 10  
)
```

Arguments

author	character, author name
role	character, e.g. "Prepared by"
affiliation	optional affiliation line
date	optional date string (character); no default is stamped
extra	optional character vector of extra lines
size	font size in points

Value

A block object (a `block_pre` under the hood): a list with components `measure`, `draw`, `keep` and `split` that the flow engine understands. Pass it to `flow_run` or `flow_add` to place it in the document.

<code>block_flow_diagram</code>	<i>A model-development flow-diagram block</i>
---------------------------------	---

Description

Renders a tree of nodes (e.g. NONMEM model runs) as boxes connected parent-to-child. Each box is sized to its measured multi-line label; the tree is laid out tidily (children centred under parents) and scaled uniformly to fit the reserved band, so adding more models shrinks the diagram instead of overlapping it.

Usage

```
block_flow_diagram(
  nodes,
  height = 500,
  size = 8,
  box_pad = 4,
  vgap = 34,
  hgap = 14,
  connect = TRUE,
  box_col = "black",
  line_col = "grey55"
)
```

Arguments

nodes	a data.frame with columns <code>id</code> , <code>parent</code> (an id, or NA/"" for a root) and <code>label</code> (a string; use "\n" for multi-line node text)
height	band height in points
size	maximum node font size in points (shrinks with the scale)

box_pad	padding inside boxes in points
vgap, hgap	vertical/horizontal gaps between nodes in points
connect	draw parent-child connectors
box_col, line_col	box border and connector colours

Value

A block object: a list with components `measure`, `draw`, `keep` and `split` that the flow engine understands. Pass it to `flow_run` or `flow_add` to place it in the document; the constructor itself draws nothing.

block_image	<i>A block that embeds a raster image</i>
-------------	---

Description

Places an image (a raster object, a numeric array/matrix, or a PNG/JPEG file path) into a band of the given height on the flowing page.

Usage

```
block_image(img, height, width = NULL, interpolate = TRUE)
```

Arguments

img	a raster, array/matrix, or path to a PNG/JPEG file
height	band height in points
width	band width in points (default: full frame width)
interpolate	passed to <code>rasterImage</code>

Value

A block object: a list with components `measure`, `draw`, `keep` and `split` that the flow engine understands. Pass it to `flow_run` or `flow_add` to place it in the document; the constructor itself draws nothing.

block_keep	<i>Keep a group of blocks together on one page (atomic)</i>
------------	---

Description

Keep a group of blocks together on one page (atomic)

Usage

```
block_keep(blocks, gap = 0)
```

Arguments

blocks	a list of blocks
gap	inter-block gap inside the group

Value

A block object: a list with components `measure`, `draw`, `keep` and `split` that the flow engine understands. Pass it to `flow_run` or `flow_add` to place the whole group on one page; the constructor itself draws nothing.

block_matrix	<i>A wide-matrix block that groups columns to fit the page width</i>
--------------	--

Description

Renders a matrix; if it is wider than the frame, the data columns are split into groups that each fit (row labels repeated in every group), and the groups are stacked vertically. This replaces hand-placed column loops such as `PrintXt(row, i * 8, ...)` that overflow the page for many columns.

Usage

```
block_matrix(
  m,
  size = 9,
  family = "mono",
  font = 1L,
  leading = 1.3,
  pad = 4,
  gap = size * 0.8,
  max_cols_per_group = NULL
)
```

Arguments

m	a matrix (or data.frame)
size	font size in points
family	font family (monospace by default for aligned numbers)
font	face
leading	line-spacing factor
pad	horizontal cell padding in points
gap	vertical gap between column groups (points)
max_cols_per_group	optional cap on data columns per group

Value

A block object: a list with components `measure`, `draw`, `keep` and `split` that the flow engine understands. Pass it to `flow_run` or `flow_add` to place it in the document; the constructor itself draws nothing.

block_para	<i>A word-wrapped paragraph block</i>
------------	---------------------------------------

Description

Wraps text to the frame width using measured widths, so it never overflows horizontally, reports its true multi-line height, and splits across pages line-by-line.

Usage

```
block_para(
  text,
  size = 10,
  font = 1L,
  family = "sans",
  leading = 1.2,
  align = c("left", "right", "center", "justify"),
  indent = 0,
  col = "black"
)
```

Arguments

text	a character string (may contain explicit newlines)
size	font size in points
font	face: 1 plain, 2 bold, 3 italic, 4 bold-italic
family	font family ("sans"/"serif"/"mono" or a core-font name)

leading	line-spacing factor
align	"left", "right", "center", or "justify" (justify falls back to left in P0)
indent	left indent in points
col	text colour

Value

A block object: a list with components `measure`, `draw`, `keep` and `split` that the flow engine understands. Pass it to `flow_run` or `flow_add` to place it in the document; the constructor itself draws nothing.

block_plot	<i>A block that draws a base-R plot into a reserved region</i>
------------	--

Description

Evaluates a plotting expression (e.g. `plot(...)`, `hist(...)`, `qqnorm(...)`) inside a band of the given height on the flowing page. The plot is confined to the band and the text canvas is restored afterwards, so plots sit inline with text and participate in pagination (a plot that does not fit the remaining space moves to the next page). The expression is captured unevaluated and run at draw time in the caller's environment.

Usage

```
block_plot(expr, height, width = NULL, mar = c(4, 4, 2, 1))
```

Arguments

<code>expr</code>	a plotting expression (not evaluated until drawn)
<code>height</code>	band height in points
<code>width</code>	band width in points (default: full frame width)
<code>mar</code>	plot margins in lines, <code>par("mar")</code> style

Value

A block object: a list with components `measure`, `draw`, `keep` and `split` that the flow engine understands. Pass it to `flow_run` or `flow_add` to place it in the document; the constructor itself draws nothing.

block_pre	<i>A preformatted (monospace) block for capture.output() dumps</i>
-----------	--

Description

Draws each element of lines as one physical line, and auto-shrinks the font so the widest line fits the frame width (fixing the options(width) wrap/overlap of dead-reckoned reports). Splits across pages line-by-line.

Usage

```
block_pre(
  lines,
  size = 9,
  font = 1L,
  family = "mono",
  leading = 1.15,
  col = "black"
)
```

Arguments

lines	a character vector, one element per line
size	font size in points (upper bound; may shrink to fit width)
font	face
family	font family (defaults to monospace)
leading	line-spacing factor
col	text colour

Value

A block object: a list with components `measure`, `draw`, `keep` and `split` that the flow engine understands. Pass it to `flow_run` or `flow_add` to place it in the document; the constructor itself draws nothing.

block_rule	<i>A horizontal rule spanning the frame width</i>
------------	---

Description

A horizontal rule spanning the frame width

Usage

```
block_rule(color = "black", pad = 2, lwd = 1)
```

Arguments

color	line colour
pad	vertical padding above/below the rule (points)
lwd	line width

Value

A block object: a list with components `measure`, `draw`, `keep` and `split` that the flow engine understands. Pass it to `flow_run` or `flow_add` to draw the rule at the cursor; the constructor itself draws nothing.

block_signature	<i>A signature / approval block (visible lines + optional interactive fields)</i>
-----------------	---

Description

Draws one signing line per role with a caption beneath and, when `with_date`, a dated line to the right. Each signing rectangle is recorded on doc (in `doc$sig`) so `sp_add_sig_fields` can place a clickable Acrobat signature field exactly over it after the PDF is written.

Usage

```
block_signature(
  roles = c("Performed by", "Reviewed by"),
  with_date = TRUE,
  size = 10,
  caption_size = 8,
  sig_height = 26,
  sig_frac = 0.5,
  date_frac = 0.22,
  gap = 16,
  field = TRUE
)
```

Arguments

roles	character vector of signer roles (e.g. "Performed by")
with_date	add a dated line beside each signature
size, caption_size	text sizes in points
sig_height	height of the signing space in points
sig_frac, date_frac	signature/date line widths as fractions of the frame
gap	vertical gap between rows in points
field	record interactive-field rectangles on the document

Value

A block object: a list with components `measure`, `draw`, `keep` and `split` that the flow engine understands. Pass it to `flow_run` or `flow_add` to place it in the document; when `field = TRUE` drawing also records each signing rectangle in `doc$sign` for `sp_add_sig_fields`.

block_spacer	<i>A vertical spacer block</i>
--------------	--------------------------------

Description

A vertical spacer block

Usage

`block_spacer(height)`

Arguments

`height` height in points

Value

A block object: a list with components `measure`, `draw`, `keep` and `split` that the flow engine understands. When placed by `flow_run` or `flow_add` it consumes height points of vertical space without drawing anything.

block_table	<i>A table block with measured column widths and automatic page splitting</i>
-------------	---

Description

Renders a `data.frame` or matrix as a table: column widths are measured from the content, numeric columns are right-aligned, the header repeats on every page, and the body splits at row boundaries when it does not fit. If the natural width exceeds the frame, the font is shrunk uniformly to fit (use `block_matrix` to instead split a wide matrix into column groups).

Usage

```
block_table(
  x,
  size = 9,
  family = "sans",
  font = 1L,
  header = TRUE,
  leading = 1.3,
  pad = 4,
  rule = TRUE
)
```

Arguments

x	a data.frame or matrix
size	font size in points (upper bound; may shrink to fit width)
family	font family
font	face
header	show the column-name header row
leading	line-spacing factor
pad	horizontal cell padding in points
rule	draw a rule under the header

Value

A block object: a list with components `measure`, `draw`, `keep` and `split` that the flow engine understands. Pass it to [flow_run](#) or [flow_add](#) to place it in the document; the constructor itself draws nothing.

flow_add	<i>Add a single block imperatively</i>
----------	--

Description

Add a single block imperatively

Usage

```
flow_add(doc, block, gap = 4)
```

Arguments

doc	a document handle
block	a block, or NULL (no-op)
gap	inter-block gap in points

Value

The document handle doc, invisibly (an environment, updated in place). Called for its side effect of drawing block into the PDF.

flow_run	<i>Run a sequence of blocks through the flow engine</i>
----------	---

Description

Measures each block, draws it if it fits, otherwise starts a new page (splitting splittable blocks across the break). NULL entries are dropped, so conditional blocks (if (cond) block_*) are fine.

Usage

```
flow_run(doc, blocks, header = NULL, footer = NULL, gap = 4)
```

Arguments

doc	a document handle
blocks	a list of blocks
header, footer	running header/footer block(s) for every page
gap	inter-block vertical gap in points

Value

The document handle doc, invisibly. The handle is an environment updated in place (pages are added and the cursor advances); the function is called mainly for its side effect of drawing the blocks into the PDF.

frame_set	<i>Set the content frame (and reset the cursor to its top)</i>
-----------	--

Description

Set the content frame (and reset the cursor to its top)

Usage

```
frame_set(doc, top, bottom, left, right)
```

Arguments

doc	a document handle
top, bottom, left, right	frame edges in points

Value

The document handle doc, invisibly (an environment; the frame and the cursor are updated in place). Called for this side effect.

remaining	<i>Vertical space remaining in the current frame (points)</i>
-----------	---

Description

Vertical space remaining in the current frame (points)

Usage

remaining(doc)

Arguments

doc	a document handle
-----	-------------------

Value

A single numeric value: the vertical space in points still available between the cursor and the bottom of the content frame.

running	<i>Set the running header / footer</i>
---------	--

Description

Set the running header / footer

Usage

running_header(doc, blocks)

running_footer(doc, blocks)

Arguments

doc	a document handle
blocks	a block or list of blocks

Value

The document handle doc, invisibly (the running blocks are stored on the handle in place). Called for its side effect: the header (top margin) or footer (bottom margin) is re-drawn on every subsequent page.

sp_add_sig_fields	<i>Add interactive Acrobat signature fields to a finished PDF</i>
-------------------	---

Description

Inserts empty AcroForm digital-signature (/Sig) fields into a base-R pdf() document (such as one produced by simPDF) using a pure base-R incremental update – no external tools or packages. The result can be signed with one click in the free Adobe Acrobat Reader. Generalised from NonCompart::addSigFieldNCA to accept fields on any page, e.g. the rectangles recorded by [block_signature](#) in doc\$sig.

Usage

```
sp_add_sig_fields(pdf, fields, out = pdf)
```

Arguments

pdf	path to the input PDF
fields	a list of fields, each list(page=, name=, rect=c(x0,y0,x1,y1)); or a document handle whose \$sig list is used
out	output path (default: overwrite pdf)

Value

The output file path out, invisibly (a character string). Called for its side effect of writing the modified PDF, with the signature fields appended as an incremental update.

sp_close	<i>Close the document (flush the PDF)</i>
----------	---

Description

Close the document (flush the PDF)

Usage

```
sp_close(doc)
```

Arguments

doc	a document handle
-----	-------------------

Value

No return value (invisible(NULL)); called for its side effect of closing the graphics device, which writes and finalises the PDF file.

sp_figure_page	<i>Draw a full-page multi-panel figure (outside the flow engine)</i>
----------------	--

Description

For per-subject / per-parameter diagnostic pages that are a whole page of base-R graphics (a `par(mfrow)` grid), rather than a block within the text flow. Starts a fresh page, sets `mfrow/oma/mar`, runs the plotting expression, and counts one page. The expression should draw exactly `prod(mfrow)` panels (pad with blank `plot.new()` if fewer) so each call maps to one page.

Usage

```
sp_figure_page(
  doc,
  expr,
  mfrow = c(1, 1),
  oma = c(0, 0, 2, 0),
  mar = c(4, 4, 2, 1)
)
```

Arguments

<code>doc</code>	a document handle
<code>expr</code>	base-R plotting expression (not evaluated until drawn)
<code>mfrow</code>	panel grid, e.g. <code>c(2, 3)</code>
<code>oma, mar</code>	outer/inner margins in lines

Value

The document handle `doc`, invisibly (the page counter is advanced in place). Called for its side effect of drawing one full page of panels.

sp_height	<i>Line height (vertical advance) in points</i>
-----------	---

Description

Line height (vertical advance) in points

Usage

```
sp_height(doc, size = doc$size, lines = 1, leading = 1.2)
```

Arguments

doc	a simPDF document handle
size	font size in points
lines	number of lines
leading	line-spacing factor (line pitch = size * leading)

Value

A single numeric value: the vertical space in points that lines lines occupy at the given size and leading (lines * size * leading).

sp_new	<i>Open a new simPDF document</i>
--------	-----------------------------------

Description

Opens a graphics-device PDF and returns a mutable document handle. Text is laid out in PDF points with the origin at the bottom-left of the page. A default content frame is set inside margin on all sides.

Usage

```
sp_new(
  file,
  paper = "A4",
  margin = 25,
  family = "Helvetica",
  size = 10,
  pointsize = 12,
  cairo = FALSE,
  onefile = TRUE
)
```

Arguments

file	output PDF path
paper	"A4", "letter", or "A3"
margin	page margin in points
family	default font family
size	default font size in points
pointsize	device pointsize (cex base); leave at default
cairo	use cairo_pdf (font embedding, Unicode/CJK) instead of pdf
onefile	single multi-page file (TRUE) vs one file per page

Value

A document handle: an environment holding the open graphics device, page geometry, content frame, cursor position and running header/footer state. It is mutable and is updated in place by the other simPDF functions; pass it to them and finish with [sp_close](#).

sp_overlaps	<i>Return overlapping text bounding-box pairs recorded while tracing</i>
-------------	--

Description

Return overlapping text bounding-box pairs recorded while tracing

Usage

```
sp_overlaps(doc, eps = 0.1)
```

Arguments

doc	a document handle
eps	overlap tolerance in points

Value

A data.frame with one row per overlapping pair of recorded text bounding boxes and columns i, j (indices of the two boxes in the trace) and page (the page they overlap on). It has zero rows when nothing overlaps.

sp_page	<i>Start a new page</i>
---------	-------------------------

Description

Emits a new PDF page, resets the cursor to the frame top, and re-stamps the running header/footer.

Usage

```
sp_page(doc, redraw_running = TRUE)
```

Arguments

doc	a document handle
redraw_running	redraw running header/footer on the new page

Value

The document handle doc, invisibly (the page counter and cursor are updated in place). Called for its side effect of emitting a new PDF page.

sp_trace	<i>Turn bounding-box tracing on for no-overlap checks</i>
----------	---

Description

Turn bounding-box tracing on for no-overlap checks

Usage

```
sp_trace(doc, on = TRUE)
```

Arguments

doc	a document handle
on	logical

Value

The document handle doc, invisibly (the tracing state is toggled in place). Called for its side effect; the boxes recorded while tracing are inspected with [sp_overlaps](#).

sp_width	<i>Measure text width in points</i>
----------	-------------------------------------

Description

Returns the rendered width of text in PDF points, using the open device's font metrics. Measured in inches and scaled by 72, so it does not require an active plot page and never has a side effect on the document. AFM-exact for the 14 core PDF fonts (identical to what the viewer draws).

Usage

```
sp_width(doc, text, size = doc$size, font = 1L, family = doc$family)
```

Arguments

doc	a simPDF document handle from sp_new
text	character vector to measure
size	font size in points
font	integer face: 1 plain, 2 bold, 3 italic, 4 bold-italic
family	font family ("sans"/"serif"/"mono" or a core-font name)

Value

A numeric vector of the same length as text: the rendered width of each element in PDF points (1/72 inch).

Index

block_authorship, [2](#)
block_flow_diagram, [3](#)
block_image, [4](#)
block_keep, [5](#)
block_matrix, [5](#), [10](#)
block_para, [6](#)
block_plot, [7](#)
block_pre, [3](#), [8](#)
block_rule, [8](#)
block_signature, [9](#), [14](#)
block_spacer, [10](#)
block_table, [10](#)

flow_add, [3–11](#), [11](#)
flow_run, [3–11](#), [12](#)
frame_set, [12](#)

rasterImage, [4](#)
remaining, [13](#)
running, [13](#)
running_footer (running), [13](#)
running_header (running), [13](#)

sp_add_sig_fields, [9](#), [10](#), [14](#)
sp_close, [14](#), [17](#)
sp_figure_page, [15](#)
sp_height, [15](#)
sp_new, [16](#), [18](#)
sp_overlaps, [17](#), [18](#)
sp_page, [17](#)
sp_trace, [18](#)
sp_width, [18](#)