

putpdf paragraph — Add text or images to a PDF file

[Description](#) [Quick start](#) [Syntax](#) [Options](#)
[Remarks and examples](#) [Also see](#)

Description

`putpdf paragraph` adds a new paragraph to the active document. The newly created paragraph becomes the active paragraph. All subsequent text or images will be appended to the active paragraph.

`putpdf text` adds content to the paragraph created by `putpdf paragraph`. The text may be a plain text string or any valid Stata expression (see [\[U\] 13 Functions and expressions](#)).

`putpdf image` embeds a portable network graphics (.png) or JPEG (.jpg) file in the paragraph.

Quick start

Add paragraph to the document

```
putpdf paragraph
```

As above, but with a half-inch standard paragraph indentation

```
putpdf paragraph, indent(para,0.5)
```

Append the text “This is paragraph text” to the active paragraph and format the text as bold

```
putpdf text ("This is paragraph text"), bold
```

Add a PNG image saved as `image1` to the paragraph above, with a height and width of 4 inches

```
putpdf image image1.png, height(4) width(4)
```

Add a PNG image saved as `image2` and center the image

```
putpdf paragraph, halign(center)
```

```
putpdf image image2.png
```

Syntax

Add paragraph to document

```
putpdf paragraph [ , paragraph_options ]
```

Add text to paragraph

```
putpdf text (exp) [ , text_options ]
```

Add image to paragraph

```
putpdf image filename [ , image_options ]
```

filename is the full path to the image file or the relative path from the current working directory.

<i>paragraph_options</i>	Description
<code>font(<i>fspec</i>)</code>	set font, font size, and font color
<code>halign(<i>hvalue</i>)</code>	set paragraph alignment
<code>valign(<i>vvalue</i>)</code>	set vertical alignment of characters on each line
<code>indent(<i>indenttype</i>, #<i>[unit]</i>)</code>	set paragraph indentation
<code>spacing(<i>position</i>, #<i>[unit]</i>)</code>	set spacing between lines of text
<code>bgcolor(<i>color</i>)</code>	set background color

<i>text_options</i>	Description
<code>nformat(<i>%fmt</i>)</code>	specify numeric format for text
<code>font(<i>fspec</i>)</code>	set font, font size, and font color
<code>bold</code>	format text as bold
<code>italic</code>	format text as italic
<code>script(sub super)</code>	set subscript or superscript formatting of text
<code>strikeout</code>	strike out text
<code>underline</code>	underline text
<code>bgcolor(<i>color</i>)</code>	set background color
<code>linebreak[(#)]</code>	add line breaks after text
<code>allcaps</code>	format text as all caps

<i>image_options</i>	Description
<code>width(#<i>[unit]</i>)</code>	set image width
<code>height(#<i>[unit]</i>)</code>	set image height
<code>linebreak[(#)]</code>	add line breaks after image

fspec is

fontname [, *size* [, *color*]]

fontname may be any supported font installed on the user's computer. Base 14 fonts, Type 1 fonts, and TrueType fonts with an extension of .ttf and .ttc are supported. TrueType fonts that cannot be embedded may not be used. If *fontname* includes spaces, then it must be enclosed in double quotes. The default font is Helvetica.

size is a numeric value that represents font size measured in points. The default is 11.

color sets the text color.

unit may be in (inch), pt (point), cm (centimeter), or twip (twentieth of a point). An inch is equivalent to 72 points, 2.54 centimeters, or 1440 twips. The default is in.

color may be one of the colors listed in *Colors* of [RPT] [Appendix for putpdf](#); a valid RGB value in the form ### ##, for example, 171 248 103; or a valid RRGGBB hex value in the form #####, for example, ABF867.

Options

Options are presented under the following headings:

Options for putpdf paragraph

Options for putpdf text

Options for putpdf image

Options for putpdf paragraph

`font(fontname [, size [, color]])` sets the font, font size, and font color for the text within the paragraph. The font size and font color may be specified individually without specifying *fontname*. Use `font("", size)` to specify font size only. Use `font("", "", color)` to specify font color only. For both cases, the default font will be used.

Specifying `font()` with `putpdf paragraph` overrides font settings specified with `putpdf begin`.

`halign(hvalue)` sets the horizontal alignment of the text within the paragraph. *hvalue* may be `left`, `right`, `center`, `justified`, or `distribute`. `distribute` and `justified` justify text between the left and right margins equally, but `distribute` also changes the spacing between words and characters. The default is `halign(left)`.

`valign(vvalue)` sets the vertical alignment of the characters on each line when the paragraph contains characters of varying size. *vvalue* may be `baseline`, `bottom`, `center`, or `top`. The default is `valign(baseline)`.

`indent(indenttype, # [unit])` specifies that the paragraph be indented by # *units*. *indenttype* may be `left`, `right`, or `para`. `left` and `right` indent # *units* from the left or the right, respectively. `para` uses standard paragraph indentation and indents the first line by # inches unless another *unit* is specified. This option may be specified multiple times in a single command to accommodate different indentation settings.

`spacing(position, # [unit])` sets the spacing between lines of text. *position* may be `before`, `after`, or `line`. `before` specifies the space before the first line of the current paragraph, `after` specifies the space after the last line of the current paragraph, and `line` specifies the space between lines within the current paragraph. This option may be specified multiple times in a single command to accommodate different spacing settings.

`bgcolor(color)` sets the background color for the paragraph.

Specifying `bgcolor()` with `putpdf paragraph` overrides background color specifications from `putpdf begin`.

Options for putpdf text

`nformat(%fmt)` specifies the numeric format of the text when the content of the new text appended to the paragraph is a numeric value. This setting has no effect when the content is a string.

`font(fontname [, size [, color]])` sets the font, font size, and font color for the new text within the active paragraph. The font size and font color may be specified individually without specifying *fontname*. Use `font("", size)` to specify font size only. Use `font("", "", color)` to specify font color only. For both cases, the default font will be used.

Specifying `font()` with `putpdf text` overrides all other font settings, including those specified with `putpdf begin` and `putpdf paragraph`.

`bold` specifies that the new text in the active paragraph be formatted as bold.

`italic` specifies that the new text in the active paragraph be formatted as italic.

`script(sub|super)` changes the script style of the new text. `script(sub)` makes the text a subscript. `script(super)` makes the text a superscript.

`strikeout` specifies that the new text in the active paragraph have a strikeout mark.

`underline` specifies that the new text in the active paragraph be underlined.

`bgcolor(color)` sets the background color for the active paragraph.

Specifying `bgcolor()` with `putpdf text` overrides background color specifications from `putpdf begin` and `putpdf paragraph`.

`linebreak[(#)]` specifies that one or *#* line breaks be added after the new text.

`allcaps` specifies that all letters of the new text in the active paragraph be capitalized.

Options for putpdf image

`width(#[unit])` sets the width of the image. If the width is larger than the body width of the document, then the body width is used. If `width()` is not specified, then the default size is used; the default is determined by the image information and the body width of the document.

`height(#[unit])` sets the height of the image. If `height()` is not specified, then the height of the image is determined by the width and the aspect ratio of the image.

`linebreak[(#)]` specifies that one or *#* line breaks be added after the new image.

Remarks and examples

[stata.com](http://www.stata.com)

Once you have created a PDF file in memory with `putpdf begin`, you can add text and images by appending them to an active paragraph. You can control the formatting for the whole paragraph, such as font properties and alignment, with options for `putpdf paragraph`. You can further customize each addition of text with the options available with `putpdf text`. Text can include any valid Stata expression, such as a string, algebraic expression, or a direct reference to a stored result.

Remarks are presented under the following headings:

Adding a paragraph

Adding an image to the document

Adding a paragraph

Before we can add text or an image to a document, we must first begin a new paragraph by using `putpdf paragraph`. The current paragraph remains active until you add a new paragraph, a table, a section break, or a page break.

► Example 1: Add a paragraph with text

Suppose we want to write a description of `auto.dta` to `example.pdf`. We first open the dataset:

```
. use https://www.stata-press.com/data/r17/auto
(1978 automobile data)
```

We begin by adding a paragraph for the title of our report:

```
. putpdf begin
. putpdf paragraph, halign(center) font(,16)
. putpdf text ("Auto report"), bold underline
```

By specifying a font size without the actual font, we have simply incremented the size of the default font in our document. We could similarly change just the color of the font without specifying anything else.

We use `summarize` to compute summary statistics on the price of cars. We can use the [stored results](#) in the text that we write.

```
. summarize price, detail
```

		Price			
	Percentiles	Smallest			
1%	3291	3291			
5%	3748	3299			
10%	3895	3667	Obs		74
25%	4195	3748	Sum of wgt.		74
50%	5006.5		Mean	6165.257	
		Largest	Std. dev.	2949.496	
75%	6342	13466			
90%	11385	13594	Variance	8699526	
95%	13466	14500	Skewness	1.653434	
99%	15906	15906	Kurtosis	4.819188	

```
. return list
```

```
scalars:
```

```

      r(N) = 74
    r(sum_w) = 74
    r(mean) = 6165.256756756757
    r(Var) = 8699525.97426879
    r(sd) = 2949.495884768919
  r(skewness) = 1.653433511704859
r(kurtosis) = 4.819187528464004
    r(sum) = 456229
    r(min) = 3291
    r(max) = 15906
    r(p1) = 3291
    r(p5) = 3748
    r(p10) = 3895
    r(p25) = 4195
    r(p50) = 5006.5
    r(p75) = 6342
    r(p90) = 11385
    r(p95) = 13466
    r(p99) = 15906
```

With the scalar names in hand, we can begin describing the prices of cars in our data. However, we first create a new paragraph because we do not want the following text to be center-aligned.

```
. putpdf paragraph, indent(para,0.5)
. putpdf text ("The average car price in this dataset is $")
. putpdf text ("'r(mean)')", nformat("%5.2f")
. putpdf text (". The 25th percentile is '$r(p25)', and the 75th percentile
> is '$r(p75)'.")
```

We request the standard paragraph indentation using the `para` option and specify a half-inch indentation. We also format the average car price to two decimal places.



Adding an image to the document

Like text, images are appended to active paragraphs, and thus their alignment within the document is controlled by the options specified with `putpdf paragraph`. Images must be stored in one of the supported formats: portable network graphics (`.png`) or JPEG (`.jpg`) file.

► Example 2: Export an image to a .pdf file

Now that we know some basic information on car pricing, we can see how the price of cars correlates with their `weight` and `mileage`. We visualize this correlation with a scatterplot matrix, using `graph matrix`:

```
graph matrix weight mpg price, half
graph export scatter.png
```

Because Stata graphs use the `.gph` extension, we use `graph export` to save the scatterplot matrix as `scatter.png`. Now we can append it to a paragraph and specify the dimensions of the image.

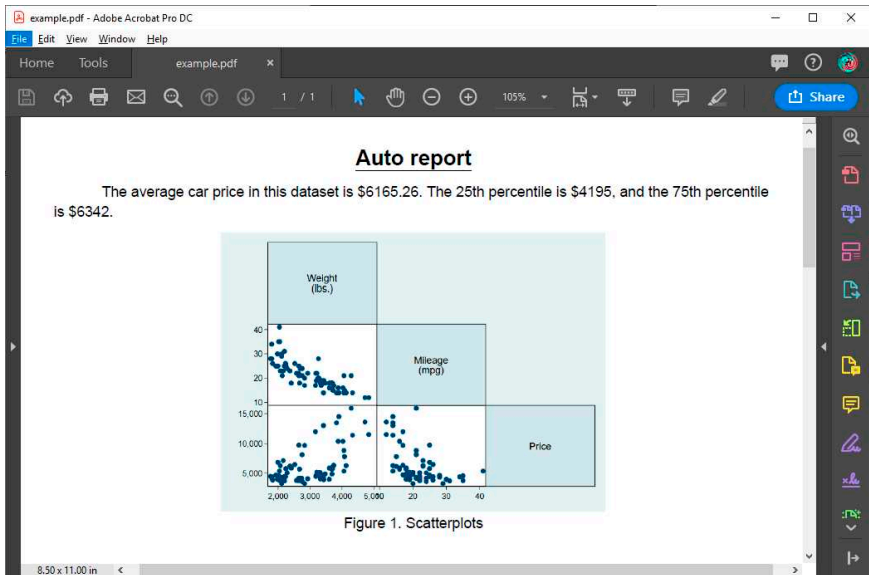
```
putpdf paragraph, halign(center)
putpdf image scatter.png, width(4) linebreak
putpdf text ("Figure 1. Scatterplots")
```

We center the image by specifying the horizontal alignment of the paragraph we are appending it to. To place a caption below the image, rather than beside it, we add a `linebreak`.

Now that we have added our descriptive statistics and a related graph, we can save and close our document:

```
putpdf save example.pdf
```

The document appears as follows:



4

Also see

- [RPT] [putpdf intro](#) — Introduction to generating PDF files
- [RPT] [putpdf begin](#) — Create a PDF file
- [RPT] [putpdf collect](#) — Add a table from a collection to a PDF file
- [RPT] [putpdf pagebreak](#) — Add breaks to a PDF file
- [RPT] [putpdf table](#) — Add tables to a PDF file
- [RPT] [Appendix for putpdf](#) — Appendix for putpdf entries