Title

stata.com

putdocx paragraph — Add text or images to an Office Open XML (.docx) file

Description

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

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

`putdocx textblock begin` adds a new paragraph to which a block of text can be added.

`putdocx textblock append` appends a block of text to the active paragraph.

`putdocx textblock end` ends a block of text initiated by `putdocx textblock begin` or `putdocx textblock append`.

`putdocx textfile` adds a block of preformatted text to a new paragraph with a predefined style.

`putdocx pagenumber` adds page numbers to a paragraph that is to be added to the header or footer.

`putdocx image` embeds a portable network graphics (`.png`), JPEG (`.jpg`), enhanced metafile (`.emf`), or tagged image file format (`.tif`) file in the active paragraph. Adding an image is not supported on console Stata for Mac.

Quick start

Add a paragraph to the document with centered horizontal alignment

```
putdocx paragraph, halign(center)
```

Append the text "This is paragraph text." to the active paragraph and format the text as bold

```
putdocx text ("This is paragraph text."), bold
```

Add a PNG image saved as `myimg` to the active paragraph

```
putdocx image myimg.png
```

Add content to the header `myheading` with the text "My Header " followed by the page number

```
putdocx paragraph, toheader(myheading)
putdocx text ("My Header ")
putdocx pagenumber
```

Add a new paragraph containing the text between the two `putdocx textblock` commands

```
putdocx textblock begin
This paragraph is written as a block of text.
putdocx textblock end
```

Add file `intro.txt` to the document as a new paragraph

```
putdocx textfile intro.txt
```

As above, but append `intro.txt` to the active paragraph

```
putdocx textfile intro.txt, append
```

Syntax

Add paragraph to document

```
putdocx paragraph [, paragraph_options]
```

Add text to paragraph

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

Add a paragraph with a block of text

```
putdocx textblock begin [, textblock_options paragraph_options]
```

Append a block of text to the active paragraph

```
putdocx textblock append [, textblock_options]
```

End the block of text initiated with `putdocx textblock begin` *or* `putdocx textblock append`

```
putdocx textblock end
```

Add page number to paragraph (to be added to header or footer)

```
putdocx pagenumber [, text_options totalpages]
```

Add a textfile as a block of preformatted text

```
putdocx textfile textfile [, append stopat(string[, stopatopt])]
```

Add image to paragraph

```
putdocx image filename [, image_options]
```

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

paragraph_options	Description
`style(`*pstyle*`)`	set paragraph text with specific style
`font(`*fspec*`)`	set font, font size, and font color
`halign(`*hvalue*`)`	set paragraph alignment
`valign(`*vvalue*`)`	set vertical alignment of characters on each line
`indent(`*indenttype*`, #[`*unit*`])`	set paragraph indentation
`spacing(`*position*`, #[`*unit*`])`	set spacing between lines of text
`shading(`*sspec*`)`	set background color, foreground color, and fill pattern
`toheader(`*hname*`)`	add paragraph content to the header *hname*
`tofooter(`*fname*`)`	add paragraph content to the footer *fname*

text_options	Description
nformat(%*fmt*)	specify numeric format for text
font(*fspec*)	set font, font size, and font color
bold	format text as bold
italic	format text as italic
script(sub | super)	set subscript or superscript formatting of text
strikeout	strike out text
underline[(*upattern*)]	underline text using specified pattern
shading(*sspec*)	set background color, foreground color, and fill pattern
linebreak[(#)]	add line breaks after text
allcaps	format text as all caps
*smallcaps	format text as small caps
*hyperlink(*link*)	add the text as a hyperlink
*trim	remove the leading and trailing spaces in the text

*Cannot be specified with putdocx pagenumber.

textblock_options	Description
[no]paramode	specify whether blank lines signal new paragraphs; default is noparamode
[no]hardbreak	specify whether spaces are added after hard line breaks; default is nohardbreak

image_options	Description
width(#[*unit*])	set image width
height(#[*unit*])	set image height
linebreak[(#)]	add line breaks after image
link	insert link to image file

fspec is

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

fontname may be any valid font installed on the user's computer. If *fontname* includes spaces, then it must be enclosed in double quotes.

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

color sets the text color.

sspec is

bgcolor [, *fgcolor* [, *fpattern*]]

bgcolor specifies the background color.

fgcolor specifies the foreground color. The default foreground color is `black`.

fpattern specifies the fill pattern. The most common fill patterns are `solid` for a solid color (determined by *fgcolor*), `pct25` for 25% gray scale, `pct50` for 50% gray scale, and `pct75` for 75% gray scale. A complete list of fill patterns is shown in *Shading patterns* of [RPT] **Appendix for putdocx**.

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

upattern may be any of the patterns listed in *Underline patterns* of [RPT] **Appendix for putdocx**. The most common underline patterns are `double`, `dash`, and `none`.

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`.

Options

Options are presented under the following headings:

Options for putdocx paragraph
Options for putdocx text
Options for putdocx textblock begin
Options for putdocx textblock append
Options for putdocx pagenumber
Options for putdocx textfile
Options for putdocx image

Options for putdocx paragraph

`style(`*pstyle*`)` specifies that the text in the paragraph be formatted with style *pstyle*. Common values for *pstyle* are `Title`, `Subtitle`, and `Heading1`. See the complete list of paragraph styles in *Paragraph styles* of [RPT] **Appendix for putdocx**.

`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 `putdocx paragraph` overrides font properties specified with `putdocx begin`.

`halign(`*hvalue*`)` sets the horizontal alignment of the text within the paragraph. *hvalue* may be `left`, `right`, `center`, `both`, or `distribute`. `distribute` and `both` 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 `auto`, `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`, `hanging`, or `para`. `left` and `right` indent # *units* from the left or the right, respectively. `hanging` uses hanging indentation and indents lines after the first line by # inches unless another *unit* is specified. `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. If both `indent(hanging)` and `indent(para)` are specified, only `indent(hanging)` is used.

`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.

`shading(`*bgcolor* [`,` *fgcolor* [`,` *fpattern*]]`)` sets the background color, foreground color, and fill pattern for the paragraph.

`toheader(`*hname*`)` specifies that the paragraph be added to the header *hname*. The paragraph will not be added to the body of the document.

`tofooter(`*fname*`)` specifies that the paragraph be added to the footer *fname*. The paragraph will not be added to the body of the document.

Options for putdocx 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 the font size only. Use `font("", "",` *color*`)` to specify the font color only. For both cases, the default font will be used.

Specifying `font()` with `putdocx text` overrides all other font settings, including those specified with `putdocx begin` and `putdocx 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`[`(`*upattern*`)`] specifies that the new text in the active paragraph be underlined and optionally specifies the format of the line. By default, a single underline is used.

`shading(`*bgcolor* [`,` *fgcolor* [`,` *fpattern*]]`)` sets the background color, foreground color, and fill pattern for the active paragraph. Specifying `shading()` with `putdocx text` overrides shading specifications from `putdocx 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.

`smallcaps` specifies that all letters of the new text in the active paragraph be capitalized, with larger capitals for uppercase letters and smaller capitals for lowercase letters.

`hyperlink(`*link*`)` adds the text as a hyperlink to the webpage address specified in *link*.

`trim` removes the leading and trailing spaces in the text.

Options for putdocx textblock begin

`style(`*pstyle*`)` specifies that the text in the paragraphs be formatted with style *pstyle*. Common values for *pstyle* are `Title`, `Subtitle`, and `Heading1`. See the complete list of paragraph styles in *Paragraph styles* of [RPT] **Appendix for putdocx**.

`font(`*fontname* [`,` *size* [`,` *color*]]`)` sets the font, font size, and font color for the text within the paragraphs. 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 `putdocx textblock begin` overrides font properties specified with `putdocx begin`.

`halign(`*hvalue*`)` sets the horizontal alignment of the text within the paragraphs. *hvalue* may be `left`, `right`, `center`, `both`, or `distribute`. `distribute` and `both` 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 paragraphs contain characters of varying size. *vvalue* may be `auto`, `baseline`, `bottom`, `center`, or `top`. The default is `valign(baseline)`.

`indent(`*indenttype*`, #`[*unit*]`)` specifies that the paragraphs be indented by *# units*. *indenttype* may be `left`, `right`, `hanging`, or `para`. `left` and `right` indent *# units* from the left or the right, respectively. `hanging` uses hanging indentation and indents lines after the first line by *#* inches unless another *unit* is specified. `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. If both `indent(hanging)` and `indent(para)` are specified, only `indent(hanging)` is used.

`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 each paragraph in the text block, `after` specifies the space after the last line of each paragraph, and `line` specifies the space between lines within each paragraph. This option may be specified multiple times in a single command to accommodate different spacing settings.

`shading(`*bgcolor* [`,` *fgcolor* [`,` *fpattern*]]`)` sets the background color, foreground color, and fill pattern for the paragraphs.

`toheader(`*hname*`)` specifies that the first paragraph in a text block be added to the header *hname*. The first paragraph will not be added to the body of the document.

When you specify the `paramode` option with `toheader()`, all paragraphs after the first will be added to the body of the document, not the header.

`tofooter(`*fname*`)` specifies that the first paragraph in a text block be added to the footer *fname*. The first paragraph will not be added to the body of the document.

When you specify the `paramode` option with `tofooter()`, all paragraphs after the first will be added to the body of the document, not the footer.

`noparamode` and `paramode` specify whether blank lines within a block of text signal the beginning of a new paragraph in the document. `noparamode`, the default, exports the block of text as a

single paragraph, removing any empty lines. `paramode` treats blank lines as an indication of a new paragraph. When `paramode` is specified, the text following each blank line will begin a new paragraph. The treatment of paragraphs within text blocks can also be specified by using `set docx_paramode`; see [RPT] **set docx**.

`nohardbreak` and `hardbreak` specify whether `putdocx textblock` inserts a space at the beginning of lines or respects spaces exactly as they are typed. `nohardbreak`, the default, respects spaces at the beginning and end of lines exactly as they are typed in the text block. The `hardbreak` option inserts a space at the beginning of all nonempty lines, excluding the first line in the block of text and the first line of each paragraph. Thus, `hardbreak` adds a space after the hard line breaks in the text block. The treatment of spaces at line breaks can also be specified by using `set docx_hardbreak`; see [RPT] **set docx**.

Options for putdocx textblock append

`noparamode` and `paramode` specify whether blank lines within a block of text signal the beginning of a new paragraph in the document. `noparamode`, the default, exports the block of text as a single paragraph, removing any empty lines. `paramode` treats blank lines as an indication of a new paragraph. When `paramode` is specified, the text following each blank line will begin a new paragraph. The treatment of paragraphs within text blocks can also be specified by using `set docx_paramode`; see [RPT] **set docx**.

`nohardbreak` and `hardbreak` specify whether `putdocx textblock` inserts a space at the beginning of lines or respects spaces exactly as they are typed. `nohardbreak`, the default, respects spaces at the beginning and end of lines exactly as they are typed in the text block. The `hardbreak` option inserts a space at the beginning of all nonempty lines, excluding the first line in the block of text and the first line of each paragraph. Thus, `hardbreak` adds a space after the hard line breaks in the text block. The treatment of spaces at line breaks can also be specified by using `set docx_hardbreak`; see [RPT] **set docx**.

Options for putdocx pagenumber

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

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

`bold` specifies that the page numbers be formatted as bold.

`italic` specifies that the page numbers be formatted as italic.

`script(sub | super)` changes the script style of the page numbers. `script(sub)` makes the page numbers subscripts. `script(super)` makes the page numbers superscripts.

`strikeout` specifies that the page numbers have a strikeout mark.

`underline`[`(`*upattern*`)`] specifies that the page numbers be underlined and optionally specifies the format of the line. By default, a single underline is used.

`shading(`*bgcolor* [`,` *fgcolor* [`,` *fpattern*]]`)` sets the background color, foreground color, and fill pattern for the page numbers. Specifying `shading()` with `putdocx pagenumber` overrides shading specifications from `putdocx paragraph`.

`linebreak`[`(#)`] specifies that one or # line breaks be added after the page numbers.

`allcaps` specifies that the page numbers be capitalized; this only applies with the page number formats `lower_letter`, `lower_roman`, `cardinal_text`, and `ordinal_text`.

`totalpages` specifies that the total number of pages be displayed, instead of the current number of the page.

Options for putdocx textfile

`append` specifies that the text file be appended to the current paragraph, rather than being added as a new paragraph.

`stopat(`*string*[`,` *stopatopt*]`)` specifies that only a portion of the text file be included, using the specified string as the end point.

stopatopt must be one of `contain`, `begin`, or `end`. The default is `contain`, which matches the string inside the line. `begin` matches the string at the beginning of the line. `end` matches the string at the end of the line. If the line matches the string, the rest of the text file, including the line, will not be included in the document.

Options for putdocx 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.

`link` specifies that a link to the image *filename* be inserted into the document. If the image is linked, then the referenced file must be present so that the document can display the image.

Remarks and examples

stata.com

Text, headers, footers, page numbers, and images are added to a `.docx` file via paragraphs. You first create a paragraph and then append text and other content to it. A paragraph can be formatted as a whole, and portions of the text within a paragraph can also be formatted individually.

To a paragraph, you can add valid Stata expressions, including strings, algebraic expressions, and direct references to stored results using `putdocx text`. For lengthier additions, you can add blocks of text using `putdocx textblock`. You can also insert preformatted text files in your `.docx` file with `putdocx textfile`. With `putdocx image`, you can add images such as Stata graphs to a paragraph.

With `putdocx paragraph`, you not only can add standard paragraphs to a document but also can create, with the variety of formatting options available to customize paragraphs, a document complete with a title, subtitle, headings, and headers and footers.

Remarks are presented under the following headings:

Adding a paragraph

With each `putdocx paragraph` command we execute, we have the opportunity to format all the text within the paragraph. For instance, we can specify font properties such as size and color along with alignment, indentation, and line spacing.

Specifying `putdocx paragraph` without any options will create an active paragraph in the body of the document. To instead add the paragraph content to a header or footer, use the `toheader()` or `tofooter()` option.

▷ Example 1: Add a title to the document

Suppose we want to create a report on some of the contributing factors to low birthweight and save it in a file called `bweight.docx`. We use the Hosmer and Lemeshow dataset.

```
. use https://www.stata-press.com/data/r16/lbw
(Hosmer & Lemeshow data)
```

We must first create our document as described in [RPT] **putdocx begin**. Because we want to add a header to our document, we specify the `header()` option, and we specify the font size for the whole document.

```
. putdocx begin, header(bwtreport) font(, 14)
```

Now we create a paragraph to which we can append text or an image. Here we indicate that we are adding a title, and then we use `putdocx text` to add the title itself.

```
. putdocx paragraph, style(Title)
. putdocx text ("Factors of low birthweight")
```

The paragraph will remain active until we add a new paragraph, a table, a section break, or a page break (see [RPT] **putdocx table** and [RPT] **putdocx pagebreak**).

◁

▷ Example 2: Define the header content

Next we define the content of our header. To do so, we use the `toheader()` option, which places the contents of the paragraph in the header rather than in the body of our document. We also include page numbers in the header:

```
. putdocx paragraph, toheader(bwtreport)
. putdocx text ("Analysis of birthweight: ")
. putdocx pagenumber
```

Now that our file is set up with a title and header, we add some text to the body of our document. In the next example, we will add a description of our dataset. Because the active paragraph is the one with the header content, we need to start a new paragraph that does not use the `toheader()` option.

```
. putdocx paragraph
```

`putdocx paragraph` with no options will use your computer's default font and the 14-point size that we requested with our previous `putdocx begin` command. See *Options for putdocx paragraph* for paragraph formatting options that can be specified to override defaults and settings specified in `putdocx begin`.

◁

Formatting text

In each `putdocx text` command, we can specify any valid Stata expression (see **[U] 13 Functions and expressions**) including, but not limited to, a string of plain text. The expression can be formatted using the *Options for putdocx text*, which would override the settings specified in `putdocx paragraph` and `putdocx begin`.

▷ Example 3: Format text individually

Suppose we want to write a description of `lbw.dta` to `bweight.docx`, including the number of women in the dataset and the average birthweight in grams (`bwt`) for their infants. We use the `summarize` command to get these descriptive statistics for the `bwt` variable.

```
. summarize bwt
```

Variable	Obs	Mean	Std. Dev.	Min	Max
bwt	189	2944.286	729.016	709	4990

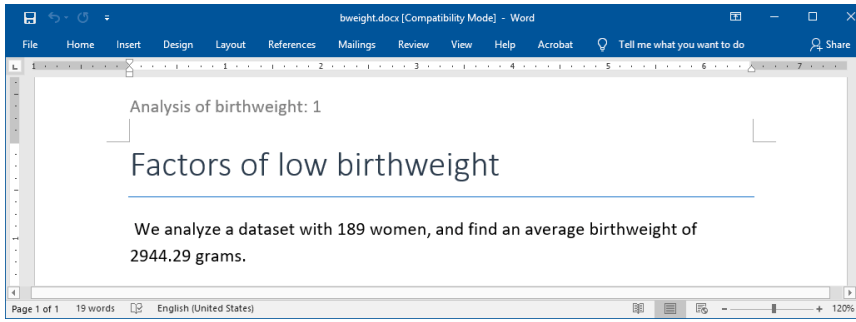
We can use the results from `summarize` in the text we write. To see the available returned results, we type `return list`. (See [P] **return** for more about stored results from Stata commands.)

```
. return list
scalars:
                  r(sum) =  556470
                  r(max) =  4990
                  r(min) =  709
                   r(sd) =  729.0160177275554
                  r(Var) =  531464.3541033434
                 r(mean) =  2944.285714285714
                r(sum_w) =  189
                    r(N) =  189
```

We see that `r(N)` stores the number of observations and `r(mean)` stores the average birthweight. Below, we create a paragraph containing this information and save the changes to our document.

```
. putdocx text (" We analyze a dataset with `r(N)' women,")
. putdocx text (" and find an average birthweight of ")
. putdocx text ( r(mean)),  nformat(%5.2f)
. putdocx text (" grams.")
. putdocx save bweight
successfully created "C:/mypath/bweight.docx"
```

Above, we formatted the mean birthweight to report only two digits after the decimal. The `bweight.docx` file now looks like this:



Working with blocks of text

While `putdocx text` is great for customizing small bits of text, it is more efficient to use the `putdocx textblock` commands when adding big blocks of text to your document. To add a paragraph with a block of text to your document, simply enclose your text between the `putdocx textblock` commands as follows:

```
putdocx textblock begin
... block of text to add
putdocx textblock end
```

To instead append a block of text to the active paragraph, use `putdocx textblock append`.

For example, continuing with the Hosmer and Lemeshow dataset, suppose we wanted to add more details about the data. We might include a text block by typing the following:

```
putdocx textblock begin
We use data presented in Hosmer, Lemeshow, and Sturdivant (2013, 24).
These data record women's demographics, such as age and race, and their medical
history, including whether they have a history of hypertension.
putdocx textblock end
```

Inserting a block of text does not limit us to using the same format throughout the block. Before we include this text block in our document, we want to add a bit of text that is formatted differently from the rest of the text block.

We can use the `<<dd_docx_display>>` dynamic tag within our text block to include Stata results and modify the text style. This dynamic tag will execute Stata's `display` command and then replace the tag with its output. We can format the output by specifying any of the text options that are also available with `putdocx text`. The entire `<<dd_docx_display` *text_option : display_directive*`>>` tag must be contained on one line—there cannot be a line break within the tag. See [RPT] **Dynamic tags** for more information on the `<<dd_docx_display>>` dynamic tag.

Example 4: Formatting text within a block

Let's modify the code above to include the name of the book that presented the dataset. Because we saved our document to view it, we must first create a new active document. Then we add the underlined book title within our text block as follows:

```
putdocx begin, font(, 14)
putdocx textblock begin
We use data presented in the book
<<dd_docx_display underline: "Applied Logistic Regression, 3rd Edition">>
by Hosmer, Lemeshow, and Sturdivant. These data record women's demographics, such
as age and race, and their medical history, including whether they have
a history of hypertension.
putdocx textblock end
```

In addition to formatting text, we can use the `<<dd_docx_display>>` dynamic tag to include Stata results. For example, we briefly mention how many women in this dataset smoked while pregnant and how many of them gave birth to an infant that weighed less than 2,500 grams. To add this content, we first store some results in local macros and then `summarize` the data.

```
. count if smoke==1
  74
. local smoke = `r(N)'
. count if low==1
  59
. local lbw = `r(N)'
. summarize
```

Variable	Obs	Mean	Std. Dev.	Min	Max
id	189	121.0794	63.30363	4	226
low	189	.3121693	.4646093	0	1
age	189	23.2381	5.298678	14	45
lwt	189	129.8201	30.57515	80	250
race	189	1.846561	.9183422	1	3
smoke	189	.3915344	.4893898	0	1
ptl	189	.1957672	.4933419	0	3
ht	189	.0634921	.2444936	0	1
ui	189	.1481481	.3561903	0	1
ftv	189	.7936508	1.059286	0	6
bwt	189	2944.286	729.016	709	4990

We can now reference these macros and any stored results in our block of text, as follows:

```
putdocx textblock begin
Out of the <<dd_docx_display: `r(N)'>> women in this dataset,
<<dd_docx_display shading(yellow): `smoke'>> smoked during their pregnancy, and
<<dd_docx_display : %4.2f `lbw'/`r(N)'>> % gave birth to an infant that weighed
less than 2,500 grams.
putdocx textblock end
```

Adding an image to the document

We can add any existing `.png`, `.jpg`, `.emf`, and `.tif` image files to a `.docx` file with `putdocx image`. For example, you could include a company logo. We can also add graphs from Stata output. Because Stata graphs use the `.gph` extension, we must first use `graph export` to convert the Stata graph to one of the supported image formats; see [G-2] **graph export**.

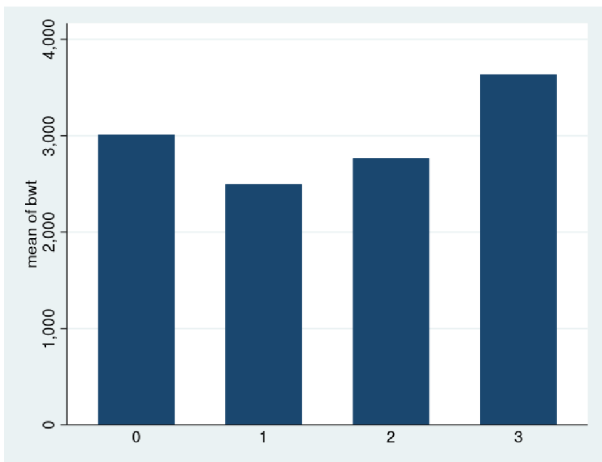
By default, images are embedded in the file. If the image is embedded, it becomes a part of the document and has no further relationship with the original image on the disk. We can instead link the image by specifying the `link` option. Using linked images means that if the saved image file is updated, then the linked image in the document will reflect the change.

If we add an image after text and want the paragraph that contains the image to have the same format as the active paragraph, we insert the image with no additional step. However, when we want to change the formatting or if there is no active paragraph, we must create one using `putdocx paragraph`. Note that we do not need to declare a new paragraph to insert an image into the cell of a table; see example 7 in [RPT] **putdocx table**.

▷ Example 5: Export a Stata graph

Another probable factor of low birthweight is the mother's history of premature labor. We have a variable (`ptl`) that records the number of times a mother prematurely went into labor. Let's add a graph showing the mean birthweight separately for mothers who never went into premature labor and those who went into premature labor once, twice, and three times. We use the `graph bar` command followed by `graph export` to create a `.png` file.

```
. graph bar bwt, over(ptl)
```

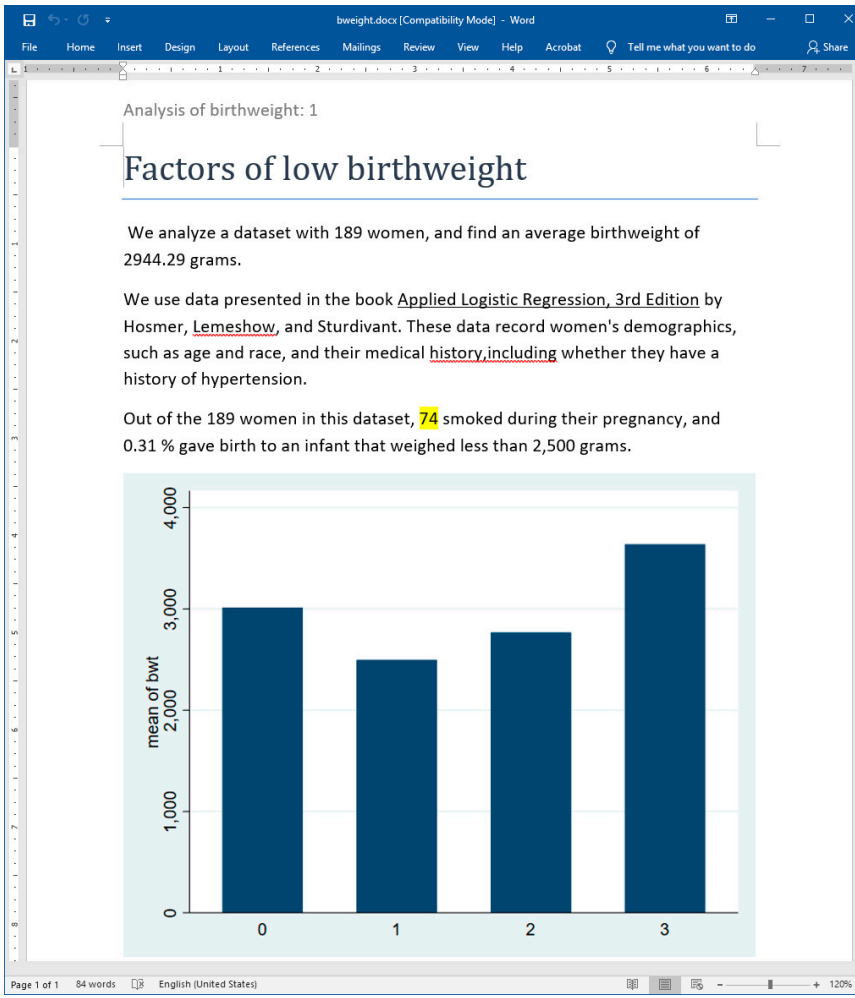



```
. graph export ptl.png
(file ptl.png written in PNG format)
```

Now we use `putdocx image` to add the `.png` file to a document. There is currently no active paragraph, so we declare a new paragraph and specify the `halign()` option to center our image; by default, paragraphs are left-aligned.

```
. putdocx paragraph, halign(center)
. putdocx image ptl.png
. putdocx save bweight, append
successfully appended to "C:/mypath/bweight.docx"
```

We append our active document, which includes the two text blocks along with this graph, to the existing content of `bweight.docx`. Here is what the file contains:



Inserting text files in the document

There may be instances where you are building a document in pieces or where you are collaborating with others to complete a project. For these situations, it may be useful to include previously created text files in your `.docx` file with `putdocx textfile`. You can even specify the portion of the text file that you want to include in your `.docx` file.

For example, let's say our classmate wrote a section about the discovery of the effects of smoking during pregnancy. We want to incorporate this in a birthweight report we have saved as `bweight2.docx`. However, the text file he sent also includes a section on hypertension, which we do not want to include. To add his text file to our document, excluding the section on hypertension, we type the following in Stata:

```
putdocx begin
putdocx textfile smoke.txt, stopat(hypertension)
putdocx save bweight2, append
```

This tells Stata that whenever it comes across a line in the text file that contains the word "hypertension", it should exclude that line and anything that follows.

Also see

[RPT] **putdocx intro** — Introduction to generating Office Open XML (.docx) files

[RPT] **putdocx begin** — Create an Office Open XML (.docx) file

[RPT] **putdocx pagebreak** — Add breaks to an Office Open XML (.docx) file

[RPT] **putdocx table** — Add tables to an Office Open XML (.docx) file

[RPT] **Appendix for putdocx** — Appendix for putdocx entries

[RPT] **set docx** — Format settings for blocks of text