# Title

stata.com

**putpdf intro —** Introduction to generating PDF files

Description Remarks and examples Also see

## Description

The `putpdf` suite of commands creates PDF documents that include text, formatted images, and tables of Stata estimation results and summary statistics. The following commands are used to create, format, add content to, and save PDF documents.

**Create and save PDF files** (see [RPT] **putpdf begin**)

| | |
|---|---|
| `putpdf begin` | Creates a PDF file for export |
| `putpdf describe` | Describes contents of the active PDF file |
| `putpdf save` | Saves and closes the PDF file |
| `putpdf clear` | Closes the PDF file without saving the changes |

**Insert page breaks in a PDF file** (see [RPT] **putpdf pagebreak**)

| | |
|---|---|
| `putpdf pagebreak` | Adds a page break to the document |
| `putpdf sectionbreak` | Adds a new section to the document |

**Add paragraph with text and images** (see [RPT] **putpdf paragraph**)

| | |
|---|---|
| `putpdf paragraph` | Adds a new paragraph to the active document |
| `putpdf text` | Adds text to the active paragraph |
| `putpdf image` | Appends an image to the active paragraph |

**Add tables to a PDF file** (see [RPT] **putpdf table**)

| | |
|---|---|
| `putpdf table` | Creates a new table in the PDF file containing estimation results, summary statistics, or data in memory |

In this manual entry, we show you how to use the `putpdf` commands by walking you through an example that creates a simple report as a PDF file.

## Remarks and examples

stata.com

Remarks are presented under the following headings:

## Introduction

putpdf is a suite of commands used to write paragraphs, images, and tables to a PDF file. This allows you to create PDF files that include Stata results and graphs. With putpdf, you can also format the text, tables, and images in the document you create.

To get started with the putpdf commands, it is best to see them in action. Here we demonstrate how to create a PDF file, include text, add a graph, and incorporate an estimation table all from within Stata.

This example shows the basic tools you need to create your own document. However, this is only a starting point. You may want to create more extensive and more customized documents, and putpdf allows you to do that. We save the details of customizing text, tables, and images for the individual entries of the commands listed above.

## Create a PDF file

To demonstrate, we create a report on 1978 automobiles using auto.dta.

```
. use https://www.stata-press.com/data/r16/auto
(1978 Automobile Data)
```

Before we can add any content to the report, we first need to create an active .pdf document in memory. We do this with the putpdf begin command.

```
. putpdf begin
```

Because we did not include any options with putpdf begin, the document created uses the letter page size and the portrait orientation.

### Add a paragraph with text

Now that the document is created, we can add other objects such as paragraphs, images, and tables to it. We begin by adding a title to our report using the Courier font with size 20 and centering it.

```
. putpdf paragraph, font("Courier",20) halign(center)
. putpdf text ("Report on 1978 automobiles")
```

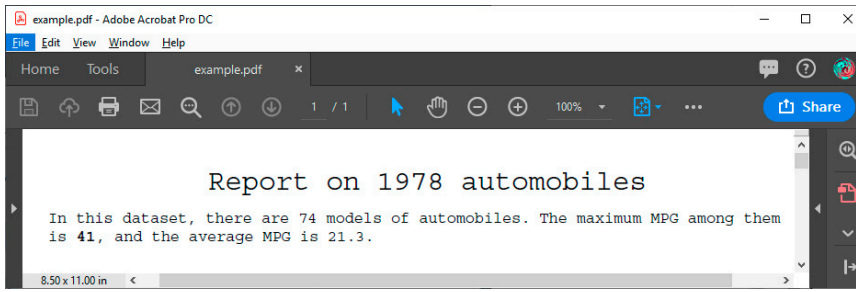
We added strings to this paragraph, but text can include any valid Stata expression. Below, we add text with summary statistics for our data by referring directly to the results stored after summarize. We type return list and see that r(N), r(mean), and r(max) store the number of automobiles, the average MPG, and the maximum MPG among those automobiles.

```
. summarize mpg
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
         mpg |         74     21.2973    5.785503         12         41

. return list

scalars:
                 r(N) =  74
             r(sum_w) =  74
              r(mean) =  21.2972972972973
               r(Var) =  33.47204738985561
                r(sd) =  5.785503209735141
              r(min) =  12
              r(max) =  41
              r(sum) =  1576
```

We add a new paragraph and then refer to these scalars directly as we add text.

```
. putpdf paragraph, font("Courier")
. putpdf text ("In this dataset, there are `r(N)'")
. putpdf text (" models of automobiles. The maximum MPG among them is ")
. putpdf text (r(max)), bold
. putpdf text (", and the average MPG is ")
. putpdf text (r(mean)), nformat("%4.1f")
. putpdf text ("."), linebreak(2)
```

We formatted `r(max)` as bold and `r(mean)` with one decimal place. We also inserted two line breaks at the end of our text to create some space between this content and the next element we export. If we were to save our document now, it would contain the following:




## Add an image to a paragraph

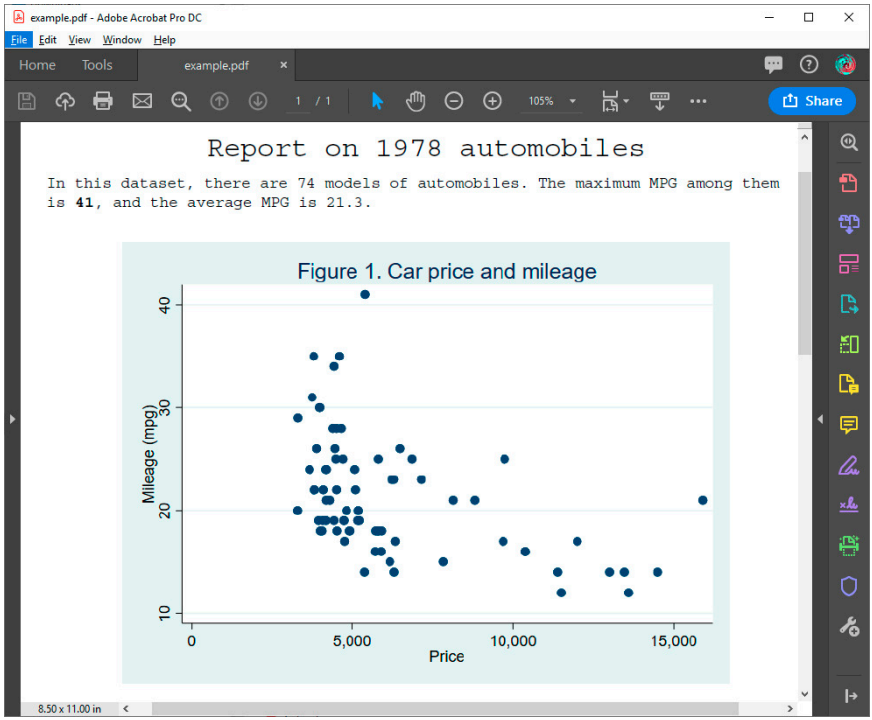
Next we add a scatterplot to our document, showing how mileage (`mpg`) correlates with the price (`price`) of cars. We use `scatter` to create our scatterplot, specifying a title for the graph. We must convert the graph to one of the supported image formats, `.png` or `.jpg`. We save it as a `.png` file with `graph export`.

```
. scatter mpg price, title("Figure 1. Car price and mileage")
. graph export auto.png
(file auto.png written in PNG format)
```

Now we use `putpdf image` to append it to the active paragraph. To center the image, we specify the alignment of the paragraph. We resize the image by setting the width at 6 inches and add a line break after the image.

```
. putpdf paragraph, halign(center)
. putpdf image auto.png, linebreak width(6)
```

If we were to save the document now, it would contain the following:




## Add table of estimation results

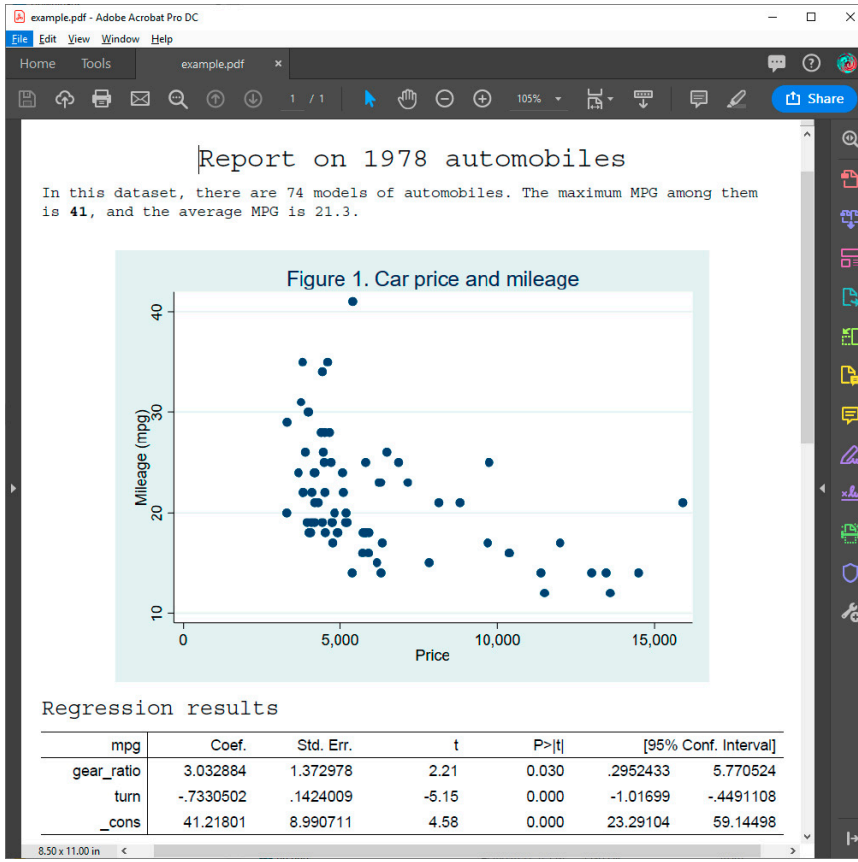
Next we will add a table with regression results after modeling fuel efficiency as a function of the car's gear ratio (`gear_ratio`) and turn radius (`turn`). We export a table named `tbl1` with all the statistics shown in the regression output below. This is done effortlessly by specifying the `etable` output type with `putpdf table`.

```
. putpdf paragraph, font("Courier",16)
. putpdf text ("Regression results")
. regress mpg gear_ratio turn, noheader
```

| mpg | Coef. | Std. Err. | t | P>\|t\| | [95% Conf. | Interval] |
|---|---|---|---|---|---|---|
| gear_ratio | 3.032884 | 1.372978 | 2.21 | 0.030 | .2952433 | 5.770524 |
| turn | -.7330502 | .1424009 | -5.15 | 0.000 | -1.01699 | -.4491108 |
| _cons | 41.21801 | 8.990711 | 4.58 | 0.000 | 23.29104 | 59.14498 |

```
. putpdf table tbl1 = etable
. putpdf save example.pdf
```

After saving our document, `example.pdf` appears as follows:




We will consider this our final report. However, you will likely want to create PDF files with more content and perhaps more customization. See [RPT] **putpdf begin** for information on formatting the document as a whole, including specifying page size, page layout, and font. See [RPT] **putpdf paragraph** for information on adding text to a document; modifying the style, font, alignment, and other formatting of a paragraph; and customizing the size and location of an image. See [RPT] **putpdf table** for information on creating tables from stored results, matrices, data, and even images and for information on customizing these tables. Finally, see [RPT] **putpdf pagebreak** for information on adding page breaks and section breaks to your document.

## Also see

[RPT] **putpdf begin** — Create a PDF file

[RPT] **putpdf pagebreak** — Add breaks to a PDF file

[RPT] **putpdf paragraph** — Add text or images to a PDF file

[RPT] **putpdf table** — Add tables to a PDF file