Create and Export Tables Using Stata

MICHAEL N. MITCHELL

A Stata Press Publication
StataCorp LLC
College Station, Texas

Published by Stata Press, 4905 Lakeway Drive, College Station, Texas 77845
Typeset in LaTeX 2$_\varepsilon$
Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

Print ISBN-10: 1-59718-369-5
Print ISBN-13: 978-1-59718-369-7
ePub ISBN-10: 1-59718-370-9
ePub ISBN-13: 978-1-59718-370-3

Library of Congress Control Number: 2024945727

(Pages omitted)

Contents

(Pages omitted)

Preface

A key step in writing a manuscript for publication is creating properly formatted tables presenting statistical results. There are many ways to approach creating such tables. You might create the tables manually, typing in each result. You might write a program to help automate the process. Or you might use one of the community-contributed tools that help create publication-quality tables from statistical output.

Starting with Stata 17, you have new options. You can use the tools built into Stata for creating publication-quality tables. These tools include the reimagined `table` command, the new `dtable` command (which requires Stata 18), the new `etable` command, and the new `collect` suite of commands. Each of these is based on the `collect` system and can be used independently or collaboratively. These tools can gather statistical results, format them as a table, customize the table, and export the table in various file formats (for example, a Word document, an Excel file, a PDF file, an HTML file, or a LaTeX file).

The focus of this book is on the `dtable` command and the `collect` suite of commands. The `dtable` command allows you to create tables of descriptive statistics, such as those shown in "Table 1" of a journal article. The `collect` commands for creating and exporting tables are general—they can be used with any estimation command and any command that stores results. These tools are flexible—you can format the tables in many ways. They are also powerful—you can combine and customize results from multiple commands. However, any tool that is general, flexible, and powerful will naturally be complex. I have specifically designed this book to address the unique challenges of learning these tools.

- **This book includes numerous complete examples**. The commands to create a table can feel like a jigsaw puzzle. Even if you have all the pieces, it can be unclear how to assemble them to make the picture. I show how the commands combine to create a table and show how to export that table. In fact, chapter 2 includes 24 sample tables. These samples are designed to illustrate how you can create many common tables. I hope that these samples can act as templates that you can use as starting points for creating your tables, letting you succeed in creating complex tables with many customizations.
- **This book focuses on both the trees and the forest.** While I do focus on complete examples, I also focus on building those examples one step at a time. Once an example is completed, I bring the pieces together in sections I call *Recap*, which show the assembled puzzle showing you the complete picture.

- **This book goes under the hood.** I take you under the hood in chapter 5 to explain the inner workings of creating tables, illustrating the contents of collections and the inner workings of the `collect` system. This chapter will help build your skills in understanding and creating tables.

Writing this book, which illustrates and explains how to create and export tables, has been uniquely challenging, perhaps the most challenging book I have written. I hope it is useful to you, and I hope you like it!

Ventura, CA
September 2024

Michael N. Mitchell

(Pages omitted)

4 Creating and exporting regression tables

This chapter will illustrate how to create regression tables based on various regression models. At key junctures, I will have a section titled *Recap*, where I illustrate the entire process of creating and exporting a table. For this chapter, the examples will illustrate

how to export tables as Word documents. You can see more complete examples showing how to create and export eight different sample regression tables in chapter 2—see samples 1 to 8.

4.1 The big picture of creating regression tables

In this chapter, I provide examples that I hope clearly show how you can create tables of your regression results. The examples can sometimes be filled with so many details that you can lose sight of the big picture. I want to begin this chapter by emphasizing the major steps of creating tables of regression results. I describe this as a four-step process.

Step 1: Collect the regression results

The first step is to use `collect` in conjunction with an estimation command (like `regress`, `logistic`, `mlogit`, or `mixed`).

```
. use nhanes2-cetus
. collect clear
. collect: regress tcresult bmi age
```

Source	SS	df	MS	Number of obs	=	10,351
				F(2, 10348)	=	1047.76
Model	4251212.07	2	2125606.04	Prob > F	=	0.0000
Residual	20993163.6	10,348	2028.71701	R-squared	=	0.1684
				Adj R-squared	=	0.1682
Total	25244375.7	10,350	2439.07011	Root MSE	=	45.041

tcresult	Coefficient	Std. err.	t	P>|t|	[95% conf.	interval]
bmi	1.055147	.0914279	11.54	0.000	.8759302	1.234363
age	1.087691	.0261034	41.67	0.000	1.036523	1.138858
_cons	138.9718	2.489416	55.83	0.000	134.0921	143.8516

Step 2: Lay out the results into rows and columns

I next use the `collect composite define` command to create a composite result named `coefse`, containing the coefficient and standard error (SE). Next I use the `collect layout` command to lay out the structure of the table, specifying the rows and columns.

```
. collect composite define coefse = _r_b _r_se, trim
. collect layout (colname) (result[coefse])
Collection: default
      Rows: colname
   Columns: result[coefse]
   Table 1: 3 x 1
```

	Coefficient Std. error
Body mass index	1.055147 .0914279
Age	1.087691 .0261034
Intercept	138.9718 2.489416

Step 3: Customize the table

In this step, I format the results, specifying three digits after the decimal. I also place parentheses around the SEs. Then I customize the labels. Lastly, I use the `collect stars` command to create and add stars that indicate the statistical significance, showing the stars after the coefficient.[1]

```
01: * Format results
02: collect style cell, nformat(%5.3f)
03: collect style cell result[_r_se], sformat("(%s)")
04: * Customize labels
05: collect label levels colname bmi "BMI", modify
06: collect label levels result coefse "Coef (SE)", modify
07: * Additional customizations
08: collect stars _r_p 0.001 "***" 0.01 "**" 0.05 "*", attach(_r_b) shownote
```

The `collect preview` command shows the customized table.

```
. collect preview
```

	Coef (SE)
BMI	1.055*** (0.091)
Age	1.088*** (0.026)
Intercept	138.972*** (2.489)

*** p<.001, ** p<.01, * p<.05

1. You could also customize the table using a style file as described in section 4.10.11 on page 365.

Step 4: Export the table

At this stage, you might want to export the table. In this example, I want to export the table as a `.docx` file. I first use the `collect style` command to specify that the table width should automatically fit the table contents. I then use the `collect export` command to export the table as a Word document. This exports the document in memory to `Reg1.docx`, which is shown in figure 4.1.

```
. collect style putdocx, layout(autofitcontents)
. collect export Reg1.docx
(collection default exported to file Reg1.docx)
```

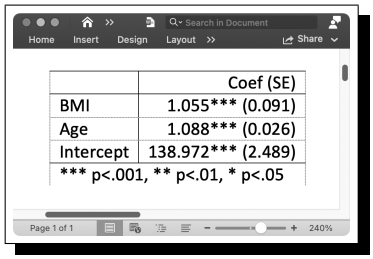

Figure 4.1. `Reg1.docx`

Recap

I gathered the commands from the prior four steps and have repeated them below so that you can see all the commands at once.

```
01: *** Step 1: Collect the regression results
02: use nhanes2-cetus
03: collect clear
04: collect: regress tcresult bmi age
05: *** Step 2: Lay out the results into rows and columns
06: collect composite define coefse = _r_b _r_se, trim
07: collect layout (colname) (result[coefse])
08: *** Step 3: Customize the table
09: * Format results
10: collect style cell, nformat(%5.3f)
11: collect style cell result[_r_se], sformat("(%s)")
12: * Customize labels
13: collect label levels colname bmi "BMI", modify
14: collect label levels result coefse "Coef (SE)", modify
15: * Additional customizations
16: collect stars _r_p 0.001 "***" 0.01 "**" 0.05 "*", attach(_r_b) shownote
17: *** Step 4: Export table as Word document
18: collect style putdocx, layout(autofitcontents)
19: collect export Reg1.docx
```