A brief description of Stata

Stata is a statistical package for managing, analyzing, and graphing data.

Stata is available for a variety of platforms. Stata may be used either as a point-and-click application or as a command-driven package.

Stata’s GUI provides an easy interface for those new to Stata and for experienced Stata users who wish to execute a command that they seldom use.

The command language provides a fast way to communicate with Stata and to communicate more complex ideas.

Here is an extract of a Stata session using the GUI:

(Throughout the Stata manuals, we will refer to various datasets. These datasets are all available from https://www.stata-press.com/data/r16/. For easy access to them within Stata, type `webuse dataset_name`, or select File > Example datasets... and click on Stata 16 manual datasets.)

```
. webuse lbw
(Hosmer & Lemeshow data)
```

We select Data > Describe data > Summary statistics and choose to summarize variables lw, age, and smoke, whose names we obtained from the Variables window. We click on OK.
Stata shows us the command that we could have typed in command mode—`summarize low age smoke`—before displaying the results of our request.

Next we fit a logistic regression model of `low` on `age` and `smoke`. We select **Statistics > Binary outcomes > Logistic regression**, fill in the fields, and click on **OK**.

![Logistic regression dialogue box](image)

```
. logistic low age smoke
Logistic regression
Number of obs = 189
LR chi2(2)    =  7.40
Prob > chi2   = 0.0248
Log likelihood = -113.63815  Pseudo R2 = 0.0315

low | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-----+---------------------------------------------------------------------
age |    0.95144    0.03042    -1.55  0.119     0.89365    1.01297
smoke|    1.99741    0.64278    2.15  0.032     1.06303    3.75308
     _cons|    1.06279    0.80488    0.08  0.936     .240890    4.68903
```

Note: `_cons` estimates baseline odds.

Here is an extract of a Stata session using the command language:
. use https://www.stata-press.com/data/r16/auto
(1978 Automobile Data)
. summarize mpg weight

<table>
<thead>
<tr>
<th>Variable</th>
<th>Obs</th>
<th>Mean</th>
<th>Std. Dev.</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>mpg</td>
<td>74</td>
<td>21.2973</td>
<td>5.785503</td>
<td>12</td>
<td>41</td>
</tr>
<tr>
<td>weight</td>
<td>74</td>
<td>3019.46</td>
<td>777.1936</td>
<td>1760</td>
<td>4840</td>
</tr>
</tbody>
</table>

The user typed summarize mpg weight and Stata responded with a table of summary statistics. Other commands would produce different results:

. generate gp100m = 100/mpg
. label var gp100m "Gallons per 100 miles"
. format gp100m %5.2f
. correlate gp100m weight
(obs=74)

<table>
<thead>
<tr>
<th></th>
<th>gp100m</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>gp100m</td>
<td>1.0000</td>
<td>0.8544</td>
</tr>
<tr>
<td>weight</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

. regress gp100m weight gear_ratio

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>Number of obs = 74</th>
<th>F(2, 71) = 96.65</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>87.4543721</td>
<td>2</td>
<td>43.7271861</td>
<td>Prob &gt; F = 0.0000</td>
<td></td>
</tr>
<tr>
<td>Residual</td>
<td>32.1218868</td>
<td>71</td>
<td>.452420967</td>
<td>R-squared = 0.7314</td>
<td></td>
</tr>
<tr>
<td>Total</td>
<td>119.576261</td>
<td>73</td>
<td>1.63803097</td>
<td>Adj R-squared = 0.7238</td>
<td></td>
</tr>
</tbody>
</table>

| gp100m | Coef.       | Std. Err. | t     | P>|t|  | [95% Conf. Interval] |
|--------|-------------|-----------|-------|------|----------------------|
| weight | .0014769    | .0001556  | 9.49  | 0.000| .0011665 .0017872   |
| gear_ratio | .1566091 | .2651131  | 0.59  | 0.557| -.3720115 .6852297  |
| _cons  | .0878243    | 1.198434  | 0.07  | 0.942| -2.301786 2.477435  |

. scatter gp100m weight, by(foreign)

The user-interface model is type a little, get a little, etc., so that the user is always in control.
Stata’s model for a dataset is that of a table—the rows are the observations and the columns are the variables:

```
. list mpg weight gp100m in 1/10

    mpg | weight  gp100m
----------+------------------
       1. |  22     2,930   4.55
       2. |  17     3,350   5.88
       3. |  22     2,640   4.55
       4. |  20     3,250   5.00
       5. |  15     4,080   6.67
       6. |  18     3,670   5.56
       7. |  26     2,230   3.85
       8. |  20     3,280   5.00
       9. |  16     3,880   6.25
      10. |  19     3,400   5.26
```

Observations are numbered; variables are named.

Stata is fast. That speed is due partly to careful programming, and partly because Stata keeps the data in memory. Stata’s file model is that of a word processor: a dataset may exist on disk, but the dataset in memory is a copy. Datasets are loaded into memory, where they are worked on, analyzed, changed, and then perhaps stored back on disk.

Working on a copy of the data in memory makes Stata safe for interactive use. The only way to harm the permanent copy of your data on disk is if you explicitly save over it.

Having the data in memory means that the dataset size is limited by the amount of computer memory. Stata stores the data in memory in an efficient format—you will be surprised how much data can fit. Nevertheless, if you work with extremely large datasets, you may run into memory constraints. You will want to learn how to store your data as efficiently as possible; see [D] compress.