# The Stata Journal





The *Stata Journal*, electronic version (ISSN 1536-8734) is a publication of Stata Press, and Stata is a registered trademark of StataCorp LP.

The Stata Journal (2006)
**6**, Number 3, pp. 425–427

# Stata tip 34: Tabulation by listing

David A. Harrison
Intensive Care National Audit and Research Centre
London, UK
david.harrison@icnarc.org

The command `list` is often regarded as simply a data management tool for listing observations, but it has several little-used options that make it a useful tool for producing customized tables.

The dataset `auto.dta` contains the 1978 repair records (rated 1–5) for various makes of car. We can use `list` in its typical manner to look at some of the data:

```
. use make rep78 using http://www.stata-press.com/data/r9/auto
(1978 Automobile Data)
. drop if missing(rep78)
(5 observations deleted)
. list in 1/5
```

| | make | rep78 |
|---|---|---|
| 1. | AMC Concord | 3 |
| 2. | AMC Pacer | 3 |
| 3. | Buick Century | 3 |
| 4. | Buick Electra | 4 |
| 5. | Buick LeSabre | 3 |

Suppose that we wish to tabulate the makes of car according to repair record. There is no simple approach to produce tables containing strings; however, if we can modify the data so that the variables in the new dataset represent the columns of our desired table and the observations represent the rows, then we can produce the table with a `list` command. The `reshape` command (see [D] **reshape**) provides the ideal tool to do this:

```
. by rep78 (make), sort: gen row = _n
. reshape wide make, i(row) j(rep78)
(note: j = 1 2 3 4 5)
Data                               long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                       69   ->      30
Number of variables                   3   ->       6
j variable (5 values)             rep78   ->   (dropped)
xij variables:
                                   make   ->   make1 make2 ... make5
-----------------------------------------------------------------------------
```

The first command above generated a variable defining the rows of our table. Sorting on `make` will ensure that the makes of car appear in alphabetical order within the table. After this `reshape`, the results of a simple `list` will still not be ideal, as the columns

 

will be headed `make1`, `make2`, etc. We can change this by using the `subvarname` option, which substitutes the characteristic `varname` for each variable as the column heading. Characteristics (see [P] **char**) are named items of text that can be attached to any variable or to the entire dataset. We use `forvalues` to loop through the values of `rep78` creating these characteristics. A previous article in the *Stata Journal* has discussed more complex looping, including using the `levelsof` (previously `levels`) and `foreach` commands to cycle through all values of a variable (Cox 2003). Other options for `list` remove observation numbers, remove the default horizontal lines every five rows, insert dividers between the columns, and make the columns of equal width:

```
. forvalues i = 1/5 {
  2.     char make‘i’[varname] "Repair record ‘i’"
  3. }
. list make1-make3, noobs sep(0) divider nocompress subvarname
```

| Repair record 1 | Repair record 2 | Repair record 3 |
|---|---|---|
| Olds Starfire | Cad. Eldorado | AMC Concord |
| Pont. Firebird | Chev. Monte Carlo | AMC Pacer |
| | Chev. Monza | Audi Fox |
| | Dodge Diplomat | Buick Century |
| | Dodge Magnum | Buick LeSabre |
| | Dodge St. Regis | Buick Regal |
| | Plym. Volare | Buick Riviera |
| | Pont. Sunbird | Buick Skylark |
| | | Cad. Deville |
| | | Cad. Seville |
| | | Chev. Chevette |
| | | Chev. Malibu |
| | | Chev. Nova |
| | | Fiat Strada |

(*output omitted*)

Only the first three columns are displayed here because of the width of the page, but if you require more columns than can be displayed in your results window (and are logging your output), you can use the `linesize(#)` option to increase the available width.

Two-way tables can be achieved in a similar manner:

```
. use make rep78 foreign using http://www.stata-press.com/data/r9/auto
(1978 Automobile Data)
. drop if missing(rep78)
(5 observations deleted)
. by rep78 foreign (make), sort: gen row = _n
. qui reshape wide make, i(rep78 row) j(foreign)
. gen str1 rep78txt = string(rep78) if row == 1
(50 missing values generated)
. format rep78txt %-1s
. char rep78txt[varname] "Repair record"
. char make0[varname] "Domestic"
. char make1[varname] "Foreign"
```

```
. list rep78txt make0 make1, noobs sepby(rep78) div noc subvar abbrev(13)
```

| Repair record | Domestic | Foreign |
|---|---|---|
| 1 | Olds Starfire<br>Pont. Firebird | |
| 2 | Cad. Eldorado<br>Chev. Monte Carlo<br>Chev. Monza<br>Dodge Diplomat<br>Dodge Magnum<br>Dodge St. Regis<br>Plym. Volare<br>Pont. Sunbird | |
| 3 | AMC Concord<br>AMC Pacer<br>Buick Century<br>Buick LeSabre<br>Buick Regal | Audi Fox<br>Fiat Strada<br>Renault Le Car |

(*output omitted*)

Do not underestimate what can be achieved with a simple `list`!

## References

Cox, N. J. 2003. Speaking Stata: Problems with lists. *Stata Journal* 3: 185–202.