Title

stata.com

error — Display generic error message and exit

Description Syntax Remarks and examples Also see

Description

`error` displays the most generic form of the error message associated with expression and sets the return code to the evaluation of the expression. If expression evaluates to 0, `error` does nothing. Otherwise, the nonzero return code will force an `exit` from the program or `capture` block in which it occurs. `error` sets the return code to 197 if there is an error in using `error` itself.

Syntax

`error` *exp*

Remarks and examples

stata.com

Remarks are presented under the following headings:

Introduction
Summary
Other messages

Introduction

`error` is used in two ways inside programs. In the first case, you want to display a standard error message so that users can look up your message by using `search`:

```
if (‘nvals’>100) error 134
```

According to [R] **search**, return code 134 is associated with the message “too many values”. During program development, you can verify that by typing the error command interactively:

```
. error 134
too many values
r(134);
```

Below we list the individual return codes so that you can select the appropriate one for use with `error` in your programs.

`error` is also used when you have processed a block of code in a `capture` block, suppressing all output. If anything has gone wrong, you want to display the error message associated with whatever the problem was:

```
capture {
        code continues
}
local rc=_rc                    preserve return code from capture
cleanup code
error ‘rc’                      present error message and exit if necessary
code could continue
```

Usually, one hopes that the return code will be zero so that `error` does nothing.

You can interrogate the built-in variable _rc to determine the type of error that occurred and then take the appropriate action. Also see [U] **16.1.4 Error handling in do-files**.

The return codes are numerically grouped, which is a feature that you may find useful when you are writing programs. The groups are

Return codes	Meaning
1–99	sundry "minor" errors
100–199	syntax errors
300–399	failure to find previously stored result
400–499	statistical problems
500–599	matrix-manipulation errors
600–699	file errors
700–799	operating-system errors
900–999	insufficient-memory errors
1000–1999	system-limit-exceeded errors
2000–2999	nonerrors (continuation of 400–499)
3000–3999	Mata run-time errors; see [M-2] **Errors** for codes
4000–4999	class system errors
7100–7199	Python run-time errors; see *Error codes* in [P] **PyStata integration**
9000–9999	system-failure errors

Summary

1. You pressed *Break*. This is not considered an error.

2. `connection timed out -- see help r(2) for troubleshooting`
   An Internet connection has timed out. This can happen when the initial attempt to make a connection over the Internet has not succeeded within a certain time limit. You can change the time limit that Stata uses under this condition by typing `set timeout1` *#seconds*. Or, the initial connection was successful, but a subsequent attempt to send or receive data over the Internet has timed out.

3. `no dataset in use`
   You attempted to perform a command requiring data and have no data in memory.

4. `no; dataset in memory has changed since last saved`
   You attempted to perform a command that would substantively alter or destroy the data, and the data have not been `saved`, at least since the data were last changed. If you wish to continue anyway, add the `clear` option to the end of the command. Otherwise, `save` the data first.

5. `not sorted`
   `master data not sorted`
   `using data not sorted`
   The observations of the data are not in the order required. To solve the problem, use `sort` to sort the data then reissue the command; see [D] **sort**.

   In the second and third cases, both the dataset in memory and the dataset on disk must be sorted by the variables specified in the varlist of `merge` before they can be merged. `merge` automatically sorts the datasets for you, unless you specify the `sorted` option. You specified `sorted`, but your dataset is not sorted on the variables in varlist. Do not specify `sorted`.

6. Return code from `confirm existence` when *string* does not exist.

7. '_____' `found where` _____ `expected`
   You are using a program that is using the `confirm` command to verify that what you typed makes sense. The messages indicate what you typed and what the program expected to find instead of what you typed.

9. `assertion is false`
   `no action taken`
   Return code and message from `assert` when the assertion is false; see [D] **assert**.
   Or, you were using `mvencode` and requested that Stata change '.' to # in the specified varlist, but # already existed in the varlist, so Stata refused; see [D] **mvencode**.

18. you must start with an empty dataset
The command (for example, infile) requires that no data be in memory—you must drop _all first. You are probably using infile to append additional data to the data in memory. Instead, save the data in memory, drop _all, infile the new data, and then append the previously saved data; see [D] **append**.

100. varlist required
= exp required
using required
by() option required
Certain commands require a varlist or another element of the language. The message specifies the required item that was missing from the command you gave. See the command's syntax diagram. For example, merge requires using to be specified; perhaps you meant to type append. Or, ranksum requires a by() option; see [R] **ranksum**.

101. varlist not allowed
weights not allowed
in range not allowed
if not allowed
= exp not allowed
using not allowed
Certain commands do not allow an if qualifier or other elements of the language. The message specifies which item in the command is not allowed. See the command's syntax diagram. For example, append does not allow a varlist; perhaps you meant to type merge.

102. too few variables specified
The command requires more variables than you specified. For instance, stack requires at least two variables. See the syntax diagram for the command.

103. too many variables specified
The command does not allow as many variables as you specified. For example, tabulate takes only one or two variables. See the syntax diagram for the command.

104. nothing to input
You gave the input command with no varlist. Stata will input onto the end of the dataset, but there is no existing dataset here. You must specify the variable names on the input command.

106. variable _____ is _____ in master but _____ in using data
You have attempted to append two datasets, but there is a string or numeric mismatch for one of the variables. The first blank is filled in with a variable name, and the second and third blanks are filled in with a storage type (byte, int, long, float, double, str#, or strL). You could specify append's force option to ignore the mismatch. If the str# type is in the master data, the using variable would then be treated as if it contained "". If the str# type is in the using data, the using variable would then be treated as if it contained numeric missing value.

key variable _____ is strL in using data
You have attempted to merge two datasets, but one of the key variables is a strL. The blank is filled in with the variable name. The key variables—the variables on which observations are matched—can be str#, but they cannot be strLs.

107. not possible with numeric variable
You have requested something that is logically impossible with a numeric variable, such as encoding it. Perhaps you meant another variable or typed encode when you meant decode.

108. not possible with string variable
You have requested something that is logically impossible with a string variable, such as decoding it. Perhaps you meant another variable or typed decode when you meant encode.

109. type mismatch
In an expression, you attempted to combine a string and numeric subexpression in a logically impossible way. For instance, you attempted to subtract a string from a number or you attempted to take the substring of a number.

110. _____ already defined
A variable or a value label has already been defined, and you attempted to redefine it. This occurs most often with generate. If you really intend to replace the values, use replace. If you intend to replace a value label, specify the replace option with the label define command. If you are attempting to alter an existing label, specify the add or modify option with the label define command.

111. _____ not found
no variables defined
The variable does not exist. You may have mistyped the variable's name.

variables out of order
You specified a varlist containing *varname1*-*varname2*, yet *varname1* occurs after *varname2*. Reverse the order of the variables if you did not make some other typographical error. Remember, *varname1*-*varname2* is taken by Stata to mean *varname1*, *varname2*, and all the variables in *dataset order* in between. Type `describe` to see the order of the variables in your dataset.

_____ not found in using data
You specified a varlist with `merge`, but the variables on which you wish to merge are not found in the using dataset, so the `merge` is not possible.

_____ ambiguous abbreviation
You typed an ambiguous abbreviation for a variable in your data. The abbreviation could refer to more than one variable. Use a nonambiguous abbreviation, or if you intend all the variables implied by the ambiguous abbreviation, append a '`*`' to the end of the abbreviation.

119. statement out of context
This is the generic form of this message; more likely, you will see messages such as "may not streset after . . . ". You have attempted to do something that, in this context, is not allowed or does not make sense.

120. invalid %format
You specified an invalid %*fmt*; see **[U] 12.5 Formats: Controlling how data are displayed**.

Return codes 121–127 are errors that might occur when you specify a numlist. For details about *numlist*, see **[U] 11.1.8 numlist**.

121. invalid numlist

122. invalid numlist has too few elements

123. invalid numlist has too many elements

124. invalid numlist has elements out of order

125. invalid numlist has elements outside of allowed range

126. invalid numlist has noninteger elements

127. invalid numlist has missing values

130. expression too long
too many SUMs
In the first case, you specified an expression that is too long for Stata to process—the expression contains more than 249 pairs of nested parentheses or more than 800 dyadic operators. Break the expression into smaller parts. In the second case, the expression contains more than 5 `sum()` functions. This expression, too, will have to be broken into smaller parts.

131. not possible with test
You requested a `test` of a hypothesis that is nonlinear in the variables. `test` tests only linear hypotheses. Use `testnl`.

132. too many '(' or '['
too many ')' or ']'
You specified an expression with unbalanced parentheses or brackets.

133. unknown function _____()
You specified a function that is unknown to Stata; see *Stata Functions Reference Manual*. Or you may have meant to subscript a variable and accidentally used parentheses rather than square brackets; see **[U] 13.7 Explicit subscripting**.

134. too many values
1) You attempted to `encode` a string variable that takes on more than 65,536 unique values. 2) You attempted to `tabulate` a variable or pair of variables that take on too many values. If you specified two variables, try interchanging them. 3) You issued a `graph` command using the `by` option. The by-variable takes on too many different values to construct a readable chart.

135. not possible with weighted data
You attempted to predict something other than the prediction or residual, but the underlying model was weighted. Stata cannot calculate the statistic you requested using weighted data.

140. repeated categorical variable in term
At least one of the terms in your anova model or test statement has a repeated categorical variable, such as reg#div#reg. Either you forgot to specify that the variable is continuous or the second occurrence of the variable is unnecessary.

141. repeated term
In the list of terms in your anova model or test statement is a duplicate of another term, although perhaps ordered differently. For instance, X#A#X and A#X#X. Remove the repeated term.

145. term contains more than 8 variables
One of the terms in your anova model test statement contains more than eight variables. Stata cannot fit such models.

147. term not in model
Your test command refers to a term that was not contained in your anova model.

148. too few categories
You attempted to run a command that required specifying the number of groups, and the number specified was too small. For instance, you attempted to run the brier command and specified group(#), where # is less than 2.

149. too many categories
You attempted to fit an mprobit or slogit model with a dependent variable that takes on more than 30 categories.

151. non r-class program may not set r()
Perhaps you specified return local in your program but forgot to declare the program rclass in the program define statement.

152. non e-class program may not set e()
Perhaps you specified estimates local in your program but forgot to declare the program eclass in the program define statement.

153. non s-class program may not set s()
Perhaps you specified sreturn local in your program but forgot to declare the program sclass in the program define statement.

161. ado-file has commands outside of program define ... end
All commands in ado-files must be part of Stata programs. That is, all commands must be between a program define that begins a program definition and an end that concludes a program definition. The command you typed automatically loaded an ado-file that violates this rule.

162. ado-file does not define command
xyz.ado is supposed to define xyz and, perhaps, subroutines for use by xyz, in which case file xyz.ado did not define anything named xyz.

170. unable to chdir
(*Unix and Mac.*) cd was unable to change to the directory you typed because it does not exist, it is protected, or it is not a directory.

175. factor level out of range
You specified an invalid value for the level of a factor variable.

180. invalid attempt to modify label
You are attempting to modify the contents of an existing value label by using the label define command. If you mean to completely replace the existing label, specify the replace option with the label define command. If you wish to modify the existing label, be sure to specify either the add option or the modify option on the label define command. add lets you add new entries but not change existing ones, and modify lets you do both. You will get this error if you specify add and then attempt to modify an existing entry. Then edit the command and substitute modify for the add option.

181. may not label strings
You attempted to assign a value label to a string variable, which makes no sense.

182. _____ not labeled
The indicated variable has no value label, yet your request requires a labeled variable. You may, for instance, be attempting to decode a numeric variable.

184. `options _____ and _____ may not be combined`
For instance, you issued the `regress` command and tried to specify both the `beta` and the `vce(cluster` *clustvar*`)` options.

190. `request may not be combined with by`
Certain commands may not be combined with `by`, and you constructed such a combination. See the syntax diagram for the command.
`in may not be combined with by`
`in` may never be combined with `by`. Use `if` instead; see **[U] 11.5 by varlist: construct**.

191. `request may not be combined with by() option`
Certain commands may not be combined with the `by()` option, and you constructed such a combination. See the syntax diagram for the command.

`in may not be combined with by`
`in` may never be combined with `by`. Use `if` instead; see **[U] 11.5 by varlist: construct**.

196. `could not restore sort order because variables were dropped`
You ran an ado-file program that has an error, and the program dropped the temporary marker variables that allow the sort order to be restored.

197. `invalid syntax`
This error is produced by `syntax` and other parsing commands when there is a syntax error in the use of the command itself rather than in what is being parsed.

198. `invalid syntax`
`option _____ incorrectly specified`
`option _____ not allowed`
`_____ invalid`
`range invalid`
`_____ invalid obs no`
`invalid filename`
`_____ invalid varname`
`_____ invalid name`
`multiple by's not allowed`
`_____ found where number expected`
`on or off required`
All items in this list indicate invalid syntax. These errors are often, but not always, due to typographical errors. Stata attempts to provide you with as much information as it can. Review the syntax diagram for the designated command.

In giving the message "invalid syntax", Stata is not helpful. Errors in specifying expressions often result in this message.

199. `unrecognized command`
Stata failed to recognize the command, program, or ado-file name, probably because of a typographical or abbreviation error.

301. `last estimates not found`
You typed an estimation command, such as `regress`, without arguments or attempted to perform a `test` or typed `predict`, but there were no previous estimation results.

302. `last test not found`
You have requested the redisplay of a previous `test`, yet you have not run a `test` previously.

303. `equation not found`
You referred to a coefficient or stored result corresponding to an equation or outcome that cannot be found. For instance, you estimated an `mlogit` model and the outcome variable took on the values 1, 3, and 4. You referred to `[2]_b[var]` when perhaps you meant `[#2]_b[var]` or `[3]_b[var]`.

304. `ml model not found`
You have used `mleval`, `mlsum`, or `mlmatsum` without having first used the other `ml` commands to define the model.

305. `ml model not found`
Same as 304.

310. `not possible because object(s) in use`
This can occur with `mata describe` and `mata drop` and indicates that the objects referred to cannot be described or eliminated because an earlier iteration of Mata is currently using them.

321. `requested action not valid after most recent estimation command`
This message can be produced by `predict` or `test` and indicates that the requested action cannot be performed.

322. `something that should be true of your estimation results is not`
This error is used by prefix commands and postestimation commands to indicate that the estimation command returned an unexpected result and that the prefix or postestimation command does not know how to proceed.

399. `may not drop constant`
You issued a `logistic` or `logit` command and the constant was dropped. Your model may be underidentified; try removing one or more of the independent variables.

401. `may not use noninteger frequency weights`
You specified an `fweight` frequency weight with noninteger weights, telling Stata that your weights are to be treated as replication counts. Stata encountered a weight that was not an integer, so your request made no sense. You probably meant to specify `aweight` analytic weights; see **[U] 11.1.6 weight**.

402. `negative weights encountered`
`negative weights not allowed`
You specified a variable that contains negative values as the weighting variable, so your request made no sense. Perhaps you meant to specify another variable.

404. `not possible with pweighted data`
You requested a statistic that Stata cannot calculate with `pweighted` data, either because of a shortcoming in Stata or because the statistics of the problem have not been worked out. For example, perhaps you requested the standard error of the Kaplan–Meier survival curve, and you had previously specified `pweight` when you `stset` your data (a case where no one has worked out the statistics).

406. `not possible with analytic weights`
You specified a command that does not allow analytic weights. See the syntax diagram for the command to see which types of weights are allowed.

407. `weights must be the same for all observations in a group`
`weights not constant for same observation across repeated variables`
For some commands, weights must be the same for all observations in a group for statistical or computational reasons. For the `anova` command with the `repeated()` option, weights must be constant for an observation across the repeated variables.

409. `no variance`
You were using `lnskew0` or `bcskew0`, for instance, but the *exp* that you specified has no variance.

411. `nonpositive values encountered`
`_____ has negative values`
`time variable has negative values`
For instance, you have used `ztest` and specified a negative value for the standard deviation in option `sd()`. Or perhaps you were using `ltable` and specified a time variable that has negative values.

412. `redundant or inconsistent constraints`
For instance, you are estimating a constrained model with `mlogit`. Among the constraints specified is at least one that is redundant or inconsistent. A redundant constraint might constrain a coefficient to be zero that some other constraint also constrains to be zero. An inconsistent constraint might constrain a coefficient to be 1 that some other constraint constrains to be zero. List the constraints, find the offender, and then reissue the `mlogit` command omitting it.

416. `missing values encountered`
You specified a variable with missing values in a place where Stata does not allow missing values.

420. `_____ groups found, 2 required`
You used a command (such as `ttest`), and the grouping variable you specified does not take on two unique values.

421. `could not determine between-subject error term; use bse() option`
You specified the `repeated()` option to `anova`, but Stata could not automatically determine certain terms that are needed in the calculation; see [R] **anova**.

422. `could not determine between-subject basic unit; use bseunit() option`
You specified the `repeated()` option to `anova`, but Stata could not automatically determine certain terms that are needed in the calculation; see [R] **anova**.

430. `convergence not achieved`
You have estimated a maximum likelihood model, and Stata's maximization procedure failed to converge to a solution; see [R] **Maximize**. Check if the model is identified.

450. _____ is not a 0/1 variable
invalid number of successes
invalid success probability
_____ takes on _____ values, not 2
You have used a command, such as bitest, that requires the variable take on only the values 0, 1, or missing, but the variable you specified does not meet that restriction. (You can also get this message from, for example, bitesti, when you specify a number of successes greater than the number of observations or a probability not between 0 and 1.)

451. invalid values for time variable
For instance, you specified mytime as a time variable, and mytime contains noninteger values.

452. invalid values for factor variable
You specified a variable that does not meet the factor-variable restrictions. Factor variables are assumed to take on only nonnegative integer values.

459. something that should be true of your data is not
data have changed since estimation
This is the generic form of this message; more likely, you will see messages such as "y must be between 0 and 1" or "x not positive". You have attempted to do something that, given your data, does not make sense.

460. fpc must be >= 0
There is a problem with your fpc variable; see [SVY] **svyset**.

461. fpc for all observations within a stratum must be the same
There is a problem with your fpc variable; see [SVY] **svyset**.

462. fpc must be <= 1 if a rate, or >= no. sampled PSUs per stratum if PSU totals
There is a problem with your fpc variable; see [SVY] **svyset**.

463. sum of weights equals zero
sum of weights for subpopulation equals zero
When weights sum to zero, the requested statistic cannot be computed.

464. poststratum weights must be constant within poststrata
You have svyset your data and specified the poststrata() and postweight() options. The variable containing poststratum population sizes must be constant within each poststratum to be valid.

465. poststratum weights must be >= 0
You have svyset your data and specified the postweight() option. Poststratum population sizes cannot be negative.

466. standardization weights must be constant within standard strata
You are using the mean, proportion, or ratio command, and you specified the stdweight() option. The weight variable for standardization must be constant within each standard stratum.

467. standardization weights must be >= 0
You are using the mean, proportion, or ratio command, and you specified the stdweight() option. The standardization weights cannot be negative.

471. esample() invalid
This concerns ereturn post. The *varname* variable specified by the esample(*varname*) option must contain exclusively 0 and 1 values (never, for instance, 2 or missing). *varname* contains invalid values.

480. starting values invalid or some RHS variables have missing values
You were using nl and specified starting values that were infeasible, or you have missing values for some of your independent variables.

481. equation/system not identified
cannot calculate derivatives
You were using reg3, for instance, and the system that you have specified is not identified.

You specified an nl *fcn* for which derivatives cannot be calculated.

482. nonpositive value(s) among _____, cannot log transform
You specified an lnlsq option in nl that attempts to take the log of a nonpositive value.

491. could not find feasible values
You are using ml and it could not find starting values for which the likelihood function could be evaluated. You could try using ml search with the repeat() option to randomly try more values, or you could use ml init to specify valid starting values.

498. *various messages*
The statistical problem described in the message has occurred. The code 498 is not helpful, but the message is supposed to be. Return code 498 is reserved for messages that are unique to a particular situation.

499. *various messages*
The statistical problem described in the message has occurred. The code 499 is not helpful, but the message is supposed to be. Return code 499 is reserved for messages that are unique to a particular situation.

501. `matrix operation not found`
You have issued an unknown `matrix` subcommand or used `matrix define` with a function or operator that is unknown to Stata.

503. `conformability error`
You have issued a `matrix` command attempting to combine two matrices that are not conformable, for example, multiplying a 3×2 matrix by a 3×3 matrix. You will also get this message if you attempt an operation that requires a square matrix and the matrix is not square.

504. `matrix has missing values`
This return code is now infrequently used because, beginning with version 8, Stata now permits missing values in matrices.

505. `matrix not symmetric`
You have issued a `matrix` command that can be performed only on a symmetric matrix, and your matrix is not symmetric. While fixing their code, programmers are requested to admire our choice of the "symmetric" number 505—it is symmetric about the zero—for this error.

506. `matrix not positive definite`
You have issued a `matrix` command that can be performed only on a positive-definite matrix, and your matrix is not positive definite.

507. `name conflict`
You have issued a `matrix post` command, and the variance–covariance matrix of the estimators does not have the same row and column names, or if it does, those names are not the same as for the coefficient vector.

508. `matrix has zero values`
`matrix has zero values on diagonal`
`matrix has zero or negative values`
`matrix has zero or negative values on diagonal`
A matrix is being used or produced that has zero or negative values where it should not. For instance, you used the `matrix sweep()` function, but the matrix had zero values on the diagonal.

509. `matrix operators that return matrices not allowed in this context`
Expressions returning nonmatrices, such as those in `generate` and `replace`, may use matrix functions returning scalars, such as `trace(A)`, but may not include subexpressions evaluating to matrices, such as `trace(A+B)`, which requires evaluating the matrix expression `A + B`. (Such subexpressions are allowed in the context of expressions returning matrices, such as those in `matrix`.)

601. `file` _____ `not found`
The filename you specified cannot be found. Perhaps you mistyped the name, or it may be on another CD or directory. If you are a Mac user, perhaps you had an unintentional blank at the beginning or ending of your filename when it was created. In Finder, click on the file to blacken the name. If you see anything other than a thin, even space on each side of the name, rename the file to eliminate the leading and trailing space characters.

602. `file` _____ `already exists`
You attempted to write over a file that already exists. Stata will never let you do this accidentally. If you really intend to overwrite the previous file, reissue the last command, specifying the `replace` option.

603. `file` _____ `could not be opened`
The file, although found, could not be opened. Check to see if it is currently open in another application or, if it is a file on your network, it is being used by another person. If it is not in use, check to see if the file is in a directory where you are allowed to create files.

604. `log file already open`
You attempted to open a `log` file when one is already open. Perhaps you forgot that you have the file open or forgot to close it.

606. `no log file open`
You have attempted to `close`, turn `on`, or turn `off` logging when no log file was open. Perhaps you forgot to open the log file.

607. `no cmdlog file open`
You have attempted to `close`, turn `on`, or turn `off` logging when no cmdlog file was open. Perhaps you forgot to open the cmdlog file.

608. `file is read-only; cannot be modified or erased`
The operating system has the file marked as read-only, meaning that changes cannot be made.

609. `file xp format`
The designated file is stored in an unsupported cross-product format.

610. `file _____ not Stata format`
The designated file is not a Stata-format file. This occurs most often with `use`, `append`, and `merge`. You probably typed the wrong filename.

611. `record too long`
You have attempted to process a record that exceeds 524,275 characters by using formatted `infile` (that is, `infile` with a dictionary). When reading formatted data, records may not exceed this maximum. If the records are not formatted, you can read these data by using the standard `infile` command (that is, without a dictionary). There is no maximum record length for unformatted data.

612. `unexpected end of file`
You used `infile` with a dictionary, and the file containing the dictionary ended before the '}' character. Perhaps you forgot to type the closing brace, or perhaps you are missing a hard return at the end of your file. You may also get this message if you issued the command `#delimit ;` in a do-file and then subsequently forgot to use ';' before the '`end`' statement.

613. `file does not contain dictionary`
You used `infile` with a dictionary, yet the file you specified does not begin with the word '`dictionary`'. Perhaps you are attempting to `infile` data without using a dictionary and forgot to specify the varlist on the `infile` command. Or you forgot to include the word `dictionary` at the top of the dictionary file or typed `DICTIONARY` in uppercase.

614. `dictionary invalid`
You used `infile` with a dictionary, and the file appears to contain a dictionary. Nevertheless, you have made some error in specifying the dictionary, and Stata does not understand your intentions. The contents of the dictionary are listed on the screen, and the last line is the line that gave rise to the problem.

616. `wrong number of values in checksum file`
The checksum file being used to verify integrity of another file does not contain values in the expected checksum format.

621. `already preserved`
You specified `preserve`, but you have already `preserved` the data.

622. `nothing to restore`
You issued the `restore` command, but you have not previously specified `preserve`.

Return codes 630–696 are all messages that you might receive when executing any command with a file over the network.

631. `host not found`

632. `web filename not supported in this context`

633. `may not write files over Internet`

639. `file transmission error (checksums do not match)`

640. `package file too long`

641. `package file invalid`

651. `may not seek past end of file`
`may not seek in write-append file`
You may not seek past the end of a file; if your desire is to increase the file's length, you must seek to the end and then write.

660. `proxy host not found`
The host name specified as a proxy server cannot be mapped to an IP address. Type `query` to determine the host you have set.

662. `proxy server refused request to send`
Stata was able to contact the proxy server, but the proxy server refused to send data back to Stata. The proxy host or port specified may be incorrect. Type `query` to determine your settings.

663. `remote connection to proxy failed -- see help r(663) for troubleshooting`
Although you have set a proxy server, it is not responding to Stata. The likely problems are that you specified the wrong port, you specified the wrong host, or the proxy server is down. Type `query` to determine the host and port that you have set.

665. `could not set socket nonblocking`

667. `wrong version winsock.dll`

668. `could not find a valid winsock.dll`

669. `invalid URL`

670. `invalid network port number`

671. `unknown network protocol`

672. `server refused to send file`
If your computer is on a network, then more than likely your computer is behind a firewall. To get Internet access from within Stata, you will have to contact your network administrator and get the network proxy address and port. Once you have the proxy information, open Stata, and in your Stata menu bar click on **Prefs** and then **General Preferences**. Under the **Internet Prefs** tab, check the box labeled *Use HTTP Proxy* and fill in the appropriate IP and port settings. If you have to enter a username and password to get Internet access, check the box labeled *Use HTTP Proxy Authorization* and fill in your username and password.

If your proxy information is entered into Stata correctly and you are still having troubles updating Stata, make sure that your firewall is caching the Stata website correctly. Sometimes at large corporate sites, there are firewalls and caching proxy servers that can interfere with some of the download operations of Stata. The error 672 in Stata is "server refused to send file", which can come if the proxy server is caching information locally and not directly forwarding the packets on to our web server. Ask your network administrators if they can trace whether your update requests from Stata are making it to our web server or if they are stopping at your firewall.

673. `authorization required by server`

674. `unexpected response from server`

675. `server reported server error`

676. `server refused request to send`

677. `remote connection failed -- see help r(677) for troubleshooting`
You requested that something be done over the web, but Stata could not contact the specified host. Perhaps the host is down; try again later.

If all your web access results in this message, perhaps your network connection is via a proxy server. If it is, you must tell Stata. Contact your system administrator and ask for the name and port of the "http proxy server". See https://www.stata.com/support/tech-support/contact/ for Stata contact information.

678. `could not open local network socket`

681. `too many open files`

682. `could not connect to odbc dsn`
This typically occurs because of incorrect permissions, such as a bad *User Name* or *Password*. Use `set debug on` to display the actual error message generated by the ODBC driver.

683. `could not fetch variable in odbc table`
This error usually occurs when a requested variable is not found in the current ODBC data table. Other scenarios can generate this error, however, so use `set debug on` to display the error message generated by the ODBC driver.

688. `file is corrupt`

691. `I/O error`
A filesystem error occurred during input or output. This typically indicates a hardware or operating system failure, although it is possible that the disk was merely full and this state was misinterpreted as an I/O error.

692. `file I/O error on read`

693. `file I/O error on write`

694. `could not rename file`
The file is in a directory that is marked by the operating system as read-only, and therefore files in that directory cannot be modified.

695. `could not copy file`
You tried to perform an `update swap` but Stata could not make a backup copy of the Stata executable, so the update was not performed.

696. `_____ is temporarily unavailable`

699. `insufficient disk space`
You ran out of disk space while writing a file to disk. The file is now closed and is probably erased. Review your operating system documentation to determine how to proceed.

702. `op. sys. refused to start new process`

703. `op. sys. refused to open pipe`

791. `system administrator will not allow you to change this setting`

900. `no room to add more variables`
Stata just attempted to exceed the maximum number of variables allowed. If you are using Stata/SE or Stata/MP, you can reset this maximum number; see [D] **memory**. For Stata/BE, the maximum number is fixed at 2,048.

901. `no room to add more observations`
Stata just attempted to exceed the maximum number of observations allowed. This maximum number is 1,099,511,627,775 for Stata/MP and 2,147,483,619 for Stata/SE and Stata/BE.

902. `no room to add more variables because of width`
Width refers to the number of bytes required to store a single observation; it is the sum of the widths of the individual variables. You just attempted to exceed the maximum width. Try typing `compress`; see [D] **compress**.

903. `no room to promote variable (e.g., change int to float) because of width`
Width refers to the number of bytes required to store a single observation; it is the sum of the widths of the individual variables. You just attempted to exceed the maximum width. Try typing `compress`; see [D] **compress**.

907. `maxvar too small`
You have attempted to use an interaction with too many levels or attempted to fit a model with too many variables. You need to increase `maxvar`. Use `set maxvar`; see [D] **memory**.

If you are using factor variables and included an interaction that has numerous missing cells, either increase `maxvar` or `set emptycells drop` to reduce the required matrix size; see [R] **set emptycells**.

If you are using factor variables, you might have accidentally treated a continuous variable as a categorical, resulting in lots of categories. Use the `c.` operator on such variables.

909. `op. sys. refuses to provide memory`
The message above can vary.
Stata was unable to allocate more memory, either because the operating system refused or because of Stata's `max_memory` setting (see [D] **memory**). The message will provide the details.

910. `value too small`
You attempted to change the size of memory but specified values for memory, maximum observations, maximum width, or maximum variables that are too small. Stata wants to allocate a minimum of 300 K.

912. `value too large`
You attempted to change the size of memory but specified values for memory, maximum observations, maximum width, or maximum variables that are too large.

913. `op. sys. refuses to provide sufficient memory`
The message above can vary.
You attempted to `set segmentsize`, and the operating system was unable to provide sufficient memory. The message will provide the details.

914. `op. sys. refused to allow Stata to open a temporary file`
To honor your request for memory, Stata needed to open a temporary disk file, and the operating system said that it could not do so. This most often occurs under Unix, and then the text of the error message provided more information on how to repair the problem.

915. `unable to allocate matrix`
You have attempted to create a matrix with too many rows or columns or attempted to fit a model with too many variables.

You are using Stata/BE which supports matrices with up to 800 rows or columns. See limits for how many more rows and columns Stata/SE and Stata/MP can support.

If you are using factor variables and included an interaction that has lots of missing cells, try `set emptycells drop` to reduce the required matrix size; see help `set emptycells`.

If you are using factor variables, you might have accidentally treated a continuous variable as a categorical, resulting in lots of categories. Use the `c.` operator on such variables.

916. `unable to allocate matrix`
You have attempted to create a matrix with too many rows or columns or attempted to fit a model with too many variables.

Assuming that you are not playing with `set max_memory`, your system administrator froze the `max_memory` setting at its current value. Contact your system administrator if you need to change this setting.

If you are using factor variables and included an interaction that has lots of missing cells, try `set emptycells drop` to reduce the required matrix size; see help `set emptycells`.

If you are using factor variables, you might have accidentally treated a continuous variable as a categorical, resulting in lots of categories. Use the `c.` operator on such variables.

920. `too many macros`
You specified a line containing recursive macro substitutions. An example of single-level recursion is referring to `"$this"` when `$this` contains `"$that"` and `$that` contains `"result"`. The result of evaluating `"$this"` is to produce `"result"`. Double-level recursion would be when `$this` contains `"$that"` and `$that` contains `"$what"` and `$what` contains `"result"`. Error 920 arises when the recursion level is greater than 20.

Error 920 also arises if macro substitution would result in text longer than the maximum number of characters allowed in a macro. See [R] **Limits**.

950. `insufficient memory`
There is insufficient memory to fulfill the request. Type `discard`, press *Return*, and try the command again. If that fails, consider dropping value labels, variable labels, or macros.

1000. `system limit exceeded - see manual`
See [R] **Limits**.

1001. `too many values`
You have attempted to create a table that has too many rows or columns. For a one-way table, the maximum number of rows is 12,000 for Stata/MP and Stata/SE and 3,000 for Stata/BE. For a two-way table, the maximum number of rows and columns is 1,200 by 80 for Stata/MP and Stata/SE and 300 by 20 for Stata/BE. Thus `tabulate y x` may not result in too many values even if `tabulate x y` does.

1002. `too many by variables`
The number of `by` variables exceeded 32,766 for Stata/MP or Stata/SE, or 2,048 for Stata/BE. You cannot exceed these maximums.

1003. `too many options`
The number of options specified exceeded 256. You cannot exceed this maximum.

1004. `command too long`
You attempted to issue a Stata command in a do-file, ado-file, or program, and the command exceeded 264,408 characters for Stata/BE. For Stata/MP and Stata/SE, the limit is 33*`c(max_k_theory)` + 216, which for the default setting of 5,000 is 165,216.

1400. `numerical overflow`
You have attempted something that, in the midst of the necessary calculations, has resulted in something too large for Stata to deal with accurately. Most commonly, this is an attempt to estimate a model (say, with `regress`) with too many effective observations. This effective number could be reached with far fewer observations if you were running a frequency-weighted model.

2000. `no observations`
You have requested some statistical calculation and there are no observations on which to perform it. Perhaps you specified `if` or `in` and inadvertently filtered all the data.

2001. `insufficient observations`
You have requested some statistical calculation, and although there are some observations, the number is not sufficient to carry out your request.

3000–3999. Mata run-time errors; see [M-2] **Errors** for codes.

4000–4999. Class system errors; see [P] **class** for information on the class system.

7100–7199. Python run-time errors; see *Error codes* in [P] **PyStata integration**.

9xxx. Various messages, all indicating an unexpected system failure. You should never see such a message. If one occurs, `save` your data, and `exit` Stata immediately. Please email tech-support@stata.com to report the problem.

Other messages

`no observations`
`insufficient observations`
You have requested something when there are either no observations or insufficient observations in memory to carry forth your request.

`(_____ not found)`
You referred to the indicated value name in an expression, and no such value label existed. A missing value was substituted.

`(eof before end of obs)`
`infile` was reading your data and encountered the end-of-file marker before it had completed reading the current observation. Missing values are filled in for the remaining variables. This message indicates that the dataset may contain more or fewer variables than you expected.

`(_____ missing values generated)`
The command created the indicated number of missing values. Missing values occur when a mathematical operation is performed on a missing value or when a mathematical operation is infeasible.

`(file _____ not found)`
You specified the `replace` option on a command, yet no such file was found. The file was saved anyway.

`(variable _____ was _____, now _____ to accommodate using data's values)`
Occurs during `append` or `merge` when there is a type mismatch between the data in memory and the data on disk. The first blank is filled in with a variable name, and the second and third blanks with a storage type (`byte`, `int`, `long`, `float`, `double`, or `str#`, or `strL`). For instance, you might receive the message "variable myvar was str5, now strL to accommodate using data's values". This means that `myvar` is of type `str5` in the *master dataset* and of type `strL` in the *using dataset*.

`(label _____ already defined)`
Occurs during `append` or `merge`. The *using* dataset has a label definition for one of its variables. A label with the same name exists in the *master* dataset. Thus you are warned that the label already exists, and the previous definition (the one from the *master* dataset) is retained.

`(note: hascons false)`
You specified the `hascons` option on `regress`, yet an examination of the data revealed that there is no effective constant in your varlist. Stata added a constant to your regression.

`_____ real changes made`
You used `replace`. This is the actual number of changes made to your data, not counting observations that already contained the replaced value.

`_____ was _____ now _____`
Occurs during `replace`, `append`, or `merge`. The first blank is filled in with a variable name, and the second and third blanks are filled in with a numeric storage type (`byte`, `int`, `long`, `float`, or `double`). For instance, you might receive the message "myvar was byte now float". Stata automatically promoted `myvar` to a `float` to prevent truncation.

Also see

[P] **break** — Suppress Break key

[P] **capture** — Capture return code

[P] **exit** — Exit from a program or do-file

[R] **search** — Search Stata documentation and other resources

[U] **16.1.4 Error handling in do-files**