

inspect — Display simple summary of data's attributes

[Description](#)

[Remarks and examples](#)

[Quick start](#)

[Stored results](#)

[Menu](#)

[Also see](#)

[Syntax](#)

Description

The `inspect` command provides a quick summary of a numeric variable that differs from the summary provided by `summarize` or `tabulate`. It reports the number of negative, zero, and positive values; the number of integers and nonintegers; the number of unique values; and the number of *missing*; and it produces a small histogram. Its purpose is not analytical but is to allow you to quickly gain familiarity with unknown data.

Quick start

Summary of all numeric variables in the dataset

```
inspect
```

Summary of `v1` for each level of `catvar`

```
bysort catvar: inspect v1
```

Summary of `v1` if `v2` is greater than 30

```
inspect v1 if v2 > 30
```

Menu

Data > Describe data > Inspect variables

Syntax

```
inspect [varlist] [if] [in]
```

`by` and `collect` are allowed; see [\[U\] 11.1.10 Prefix commands](#).

Remarks and examples

Typing `inspect` by itself produces an inspection for all the variables in the dataset. If you specify a *varlist*, an inspection of just those variables is presented.

► Example 1

`inspect` is not a replacement or substitute for `summarize` and `tabulate`. It is instead a data management or information tool that lets us quickly gain insight into the values stored in a variable.

For instance, we receive data that purport to be on automobiles, and among the variables in the dataset is one called `mpg`. Its variable label is `Mileage (mpg)`, which is surely suggestive. We `inspect` the variable,

```
. use https://www.stata-press.com/data/r18/auto
(1978 automobile data)
```

```
. inspect mpg
```

```
mpg: Mileage (mpg)
```

					Number of observations				
					Total	Integers	Nonintegers		
#	#				Negative	-	-	-	
#	#				Zero	-	-	-	
#	#				Positive	74	74	-	
#	#	#			Total	74	74	-	
#	#	#	#	.	Missing	-			
12				41	74				

(21 unique values)

and we discover that the variable is never *missing*; all 74 observations in the dataset have some value for `mpg`. Moreover, the values are all positive and are all integers, as well. Among those 74 observations are 21 unique (different) values. The variable ranges from 12 to 41, and we are provided with a small histogram that suggests that the variable appears to be what it claims.

◀

► Example 2

Bob, a coworker, presents us with some census data. Among the variables in the dataset is one called `region`, which is labeled `Census region` and is evidently a numeric variable. We `inspect` this variable:

```
. use https://www.stata-press.com/data/r18/bobsdata
(1980 Census data by state)
```

```
. inspect region
```

```
region: Census region
```

					Number of observations				
					Total	Integers	Nonintegers		
#	#				Negative	-	-	-	
#	#				Zero	-	-	-	
#	#	#			Positive	50	50	-	
#	#	#			Total	50	50	-	
#	#	#	#	.	Missing	-			
1				5	50				

(5 unique values)

region is labeled but 1 value is NOT documented in the label.

In this dataset something may be wrong. `region` takes on five unique values. The variable has a value label, however, and one of the observed values is not documented in the label. Perhaps there is a typographical error.

◀

▷ Example 3

There was indeed an error. Bob fixes it and returns the data to us. Here is what `inspect` produces now:

```
. use https://www.stata-press.com/data/r18/census
(1980 Census data by state)
. inspect region
region: Census region
```

				Number of observations		
				Total	Integers	Nonintegers
#	#	#	#	Negative	-	-
#	#	#	#	Zero	-	-
#	#	#	#	Positive	50	50
#	#	#	#	Total	50	50
#	#	#	#	Missing	-	-
1				50		
4						

(4 unique values)

region is labeled and all values are documented in the label.

◀

▷ Example 4

We receive data on the climate in 956 U.S. cities. The variable `tempjan` records the Average January temperature in degrees Fahrenheit. The results of `inspect` are

```
. use https://www.stata-press.com/data/r18/citytemp
(City temperature data)
. inspect tempjan
tempjan: Average January temperature
```

				Number of observations		
				Total	Integers	Nonintegers
#	#	#	#	Negative	-	-
#	#	#	#	Zero	-	-
#	#	#	#	Positive	954	78
#	#	#	#	Total	954	78
#	#	#	#	Missing	2	876
2.2				956		
72.6						

(More than 99 unique values)

In two of the 956 observations, `tempjan` is *missing*. Of the 954 cities that have a recorded `tempjan`, all are positive, and 78 of them are integer values. `tempjan` varies between 2.2 and 72.6. There are more than 99 unique values of `tempjan` in the dataset. (Stata stops counting unique values after 99.)

◀

Stored results

`inspect` stores the following in `r()`:

Scalars

<code>r(N)</code>	number of observations
<code>r(N_neg)</code>	number of negative observations
<code>r(N_0)</code>	number of observations equal to 0
<code>r(N_pos)</code>	number of positive observations
<code>r(N_negint)</code>	number of negative integer observations
<code>r(N_posint)</code>	number of positive integer observations
<code>r(N_unique)</code>	number of unique values or . if more than 99
<code>r(N_undoc)</code>	number of undocumented values or . if not labeled

Also see

[D] **codebook** — Describe data contents

[D] **compare** — Compare two variables

[D] **describe** — Describe data in memory or in a file

[D] **ds** — Compactly list variables with specified properties

[D] **isid** — Check for unique identifiers

[R] **lv** — Letter-value displays

[R] **summarize** — Summary statistics

[R] **table** — Table of frequencies, summaries, and command results

[R] **tabulate oneway** — One-way table of frequencies

[R] **tabulate, summarize()** — One- and two-way tables of summary statistics

[R] **tabulate twoway** — Two-way table of frequencies

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on [citing Stata documentation](#).