**Description**

clonevar generates `newvar` as an exact copy of an existing variable, `varname`, with the same storage type, values, and display format as `varname`. `varname`’s variable label, value labels, notes, and characteristics will also be copied.

**Quick start**

Copy contents, label, and value label of `v1` to `newv1`
```
clonevar newv1 = v1
```

Copy observations from `v2` to `newv2` where `v2` is less than 30
```
clonevar newv2 = v2 if v2 < 30
```

Copy the first 20 observations of `v3` to `newv3`
```
clonevar newv3 = v3 in f/20
```

Same as above
```
clonevar newv3 = v3 in 1/20
```

**Menu**

Data > Create or change data > Other variable-creation commands > Clone existing variable

**Syntax**

```plaintext
clonevar newvar = varname [if] [in]
```

**Remarks and examples**

`clonevar` has various possible uses. Programmers may desire that a temporary variable appear to the user exactly like an existing variable. Interactively, you might want a slightly modified copy of an original variable, so the natural starting point is a clone of the original.
Example 1

We have a dataset containing information on modes of travel. These data contain a variable named `mode` that identifies each observation as a specific mode of travel: air, train, bus, or car.

```
. use http://www.stata-press.com/data/r15/travel
. describe mode
```

<table>
<thead>
<tr>
<th>variable name</th>
<th>type</th>
<th>format</th>
<th>label</th>
<th>variable label</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>byte</td>
<td>%8.0g</td>
<td>travel</td>
<td>travel mode alternatives</td>
</tr>
</tbody>
</table>

To create an identical variable identifying only observations that contain air or train, we could use `clonevar` with an `if` qualifier.

```
. clonevar airtrain = mode if mode == 1 | mode == 2
(420 missing values generated)
```

```n
. describe mode airtrain
```

<table>
<thead>
<tr>
<th>variable name</th>
<th>type</th>
<th>format</th>
<th>label</th>
<th>variable label</th>
</tr>
</thead>
<tbody>
<tr>
<td>mode</td>
<td>byte</td>
<td>%8.0g</td>
<td>travel</td>
<td>travel mode alternatives</td>
</tr>
<tr>
<td>airtrain</td>
<td>byte</td>
<td>%8.0g</td>
<td>travel</td>
<td>travel mode alternatives</td>
</tr>
</tbody>
</table>

. list mode airtrain in 1/5

<table>
<thead>
<tr>
<th>mode</th>
<th>airtrain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>air</td>
</tr>
<tr>
<td>2.</td>
<td>train</td>
</tr>
<tr>
<td>3.</td>
<td>bus</td>
</tr>
<tr>
<td>4.</td>
<td>car</td>
</tr>
<tr>
<td>5.</td>
<td>air</td>
</tr>
</tbody>
</table>

The new `airtrain` variable has the same storage type, display format, value label, and variable label as `mode`. If `mode` had any characteristics or notes attached to it, they would have been applied to the new `airtrain` variable, too. The only differences in the two variables are their names and values for bus and car.

Technical note

The `if` qualifier used with the `clonevar` command in example 1 referred to the values of `mode` as 1 and 2. Had we wanted to refer to the values by their associated value labels, we could have typed

```
. clonevar airtrain = mode if mode == "air":travel | mode == "train":travel
```

For more details, see [U] 13.11 Label values.
Acknowledgments

clonavar was written by Nicholas J. Cox of the Department of Geography at Durham University, UK, and coeditor of the *Stata Journal* and author of *Speaking Stata Graphics*. He in turn thanks Michael Blasnik of Nest Labs and Ken Higbee of StataCorp for very helpful comments on a precursor of this command.

Also see

[D] generate — Create or change contents of variable

[D] separate — Create separate variables