**fvset** — Declare factor-variable settings

### Syntax

**Declare base settings**

```
fvset base base_spec varlist
```

**Declare design settings**

```
fvset design design_spec varlist
```

**Clear the current settings**

```
fvset clear varlist
```

**Report the current settings**

```
fvset report [ varlist ] [ , base(base_spec) design(design_spec) ]
```

<table>
<thead>
<tr>
<th>base_spec</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>default base</td>
</tr>
<tr>
<td>first</td>
<td>lowest level value; the default</td>
</tr>
<tr>
<td>last</td>
<td>highest level value</td>
</tr>
<tr>
<td>frequent</td>
<td>most frequent level value</td>
</tr>
<tr>
<td>none</td>
<td>no base</td>
</tr>
<tr>
<td>#</td>
<td>nonnegative integer value</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>design_spec</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>default</td>
<td>default base</td>
</tr>
<tr>
<td>asbalanced</td>
<td>accumulate using $1/k$, $k =$ number of levels</td>
</tr>
<tr>
<td>asobserved</td>
<td>accumulate using observed relative frequencies; the default</td>
</tr>
</tbody>
</table>

### Description

`fvset` declares factor-variable settings. Factor-variable settings identify the base level and how to accumulate statistics over levels.

`fvset base` specifies the base level for each variable in `varlist`. The default for factor variables without a declared base level is `first`. 
fvset — Declare factor-variable settings

fvset design specifies how to accumulate over the levels of a factor variable. The margins command is the only command aware of this setting; see [R] margins. By default, margins assumes that factor variables are asobserved, meaning that they are accumulated by weighting by the number of observations or the sum of the weights if weights have been specified.

fvset clear removes factor-variable settings for each variable in varlist. fvset clear _all removes all factor-variable settings from all variables.

fvset report reports the current factor-variable settings for each variable in varlist. fvset without arguments is a synonym for fvset report.

Options

base(base_spec) restricts fvset report to report only the factor-variable settings for variables with the specified base_spec.

design(design_spec) restricts fvset report to report only the factor-variable settings for variables with the specified design_spec.

Remarks and examples

Example 1

Using auto2.dta, we include factor variable i.rep78 in a regression:

```
. use http://www.stata-press.com/data/r13/auto2
(1978 Automobile Data)
. regress mpg i.rep78, baselevels
```

```
Source | SS       df       MS        Number of obs = 69
--------+----------------------------    F(   4,   64) = 4.91
Model   | 549.41578  4    137.35394    Prob > F = 0.0016
Residual| 1790.78712 64    27.98105     R-squared = 0.2348
         |          |           | Adj R-squared = 0.1869
Total   | 2340.2029 68  34.4147485    Root MSE = 5.2897

mpg     | Coef.  Std. Err.     t    P>|t|    [95% Conf. Interval]
--------+-------------------------------------------------    --------
rep78   |                      |          |          |                     |
    Poor | 0 (base)            |          | 0.655    | -10.22927 6.479274 |
    Fair | -1.875 4.181884     -0.45 0.655 | -10.22927 6.479274 |
    Average | -1.566667 3.863059  -0.41 0.686 | -9.284014 6.150681 |
    Good | .6666667 3.942718    0.17 0.866 | -7.209818 8.543152 |
    Excellent | 6.363636 4.066234  1.56 0.123 | -1.759599 14.48687 |
_cons   | 21 3.740391 5.61 0.000 | 13.52771 28.47229 |
```

We specified the baselevels option so that the base level would be included in the output. By default, the first level is the base level. We can change the base level to 2:
Let’s set `rep78` to have no base level and fit a cell-means regression:

```
. fvset base none rep78
. regress mpg i.rep78, noconstant
```

**Example 2**

By default, `margins` accumulates a margin by using the observed relative frequencies of the factor levels.

```
. reg mpg i.foreign
```

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>Number of obs = 74</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>378.153515</td>
<td>1</td>
<td>378.153515</td>
<td>F( 1, 72) = 13.18</td>
</tr>
<tr>
<td>Residual</td>
<td>2065.30594</td>
<td>72</td>
<td>28.6848048</td>
<td>Prob &gt; F = 0.0005</td>
</tr>
<tr>
<td>Total</td>
<td>2443.45946</td>
<td>73</td>
<td>33.4720474</td>
<td>R-squared = 0.1548</td>
</tr>
</tbody>
</table>

| `foreign` | Coef. | Std. Err. | t     | P>|t| | [95% Conf. Interval] |
|-----------|-------|-----------|-------|------|----------------------|
| Foreign   | 4.945804 | 1.362162 | 3.63  | 0.001| 2.230384 7.661225    |
| _cons     | 19.82692 | 0.7427186| 26.70 | 0.000| 18.34634 21.30751    |
Let's set `foreign` to always accumulate using equal relative frequencies:

```
.fvset design asbalanced foreign
.regress mpg i.foreign
```

Source | SS | df | MS  | Number of obs = 74  
--- | --- | --- | --- | ---
Model | 378.153515 | 1 | 378.153515 | F( 1, 72) = 13.18  
Residual | 2065.30594 | 72 | 28.6848048 | Prob > F = 0.0005 | R-squared = 0.1548  
Total | 2443.45946 | 73 | 33.4720474 | Adj R-squared = 0.1430 | Root MSE = 5.3558

mpg | Coef. | Std. Err. | t | P>|t| | [95% Conf. Interval]  
--- | --- | --- | --- | --- | ---
foreign | 4.945804 | 1.362162 | 3.63 | 0.001 | 2.230384 7.661225 |  
Foreign | 19.82692 | .7427186 | 26.70 | 0.000 | 18.34634 21.30751 |  
_cons | 22.29983 | .6810811 | 32.74 | 0.000 | 20.94211 23.65754 |  

Suppose that we issued the `fvset design` command earlier in our session and that we cannot remember which variables we set as `asbalanced`. We can retrieve this information by using the `fvset report` command:

```
.fvset report, design(asbalanced)
```

```
Variable | Base | Design
--- | --- | ---
foreign | | asbalanced
```

Technical note

`margins` is aware of a factor variable’s design setting only through the estimation results it is working with. The design setting is stored by the estimation command; thus changing the design setting between the estimation command and `margins` will have no effect. For example, the output from the following two calls to `margins` yields the same results:
. fvset clear foreign
. regress mpg i.foreign

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>Number of obs = 74</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>378.153515</td>
<td>1</td>
<td>378.153515</td>
<td>F( 1, 72) = 13.18</td>
</tr>
<tr>
<td>Residual</td>
<td>2065.30594</td>
<td>72</td>
<td>28.6848048</td>
<td>Prob &gt; F = 0.0005</td>
</tr>
<tr>
<td>Total</td>
<td>2443.45946</td>
<td>73</td>
<td>33.4720474</td>
<td>Adj R-squared = 0.1430</td>
</tr>
</tbody>
</table>

| mpg        | Coef.        | Std. Err. | t    | P>|t|  | [95% Conf. Interval] |
|------------|--------------|-----------|------|------|---------------------|
| foreign    | 4.945804     | 1.362162  | 3.63 | 0.001| 2.230384 7.661225   |
| _cons      | 19.82692     | 0.7427186 | 26.70| 0.000| 18.34634 21.30751   |

. margins
Predictive margins Number of obs = 74
Model VCE : OLS
Expression : Linear prediction, predict()

| Delta-method | Margin | Std. Err. | t    | P>|t|  | [95% Conf. Interval] |
|--------------|--------|-----------|------|------|---------------------|
| _cons        | 21.2973| 0.6226014 | 34.21| 0.000| 20.05616 22.53843   |

. fvset design asbalanced foreign
. margins
Predictive margins Number of obs = 74
Model VCE : OLS
Expression : Linear prediction, predict()

| Delta-method | Margin | Std. Err. | t    | P>|t|  | [95% Conf. Interval] |
|--------------|--------|-----------|------|------|---------------------|
| _cons        | 21.2973| 0.6226014 | 34.21| 0.000| 20.05616 22.53843   |

Stored results
fvset stores the following in r():

Macros
r(varlist) varlist
r(baselist) base setting for each variable in varlist
r(designlist) design setting for each variable in varlist