**Description**

`fvset base`, `fvset design`, and `fvset clear` manage factor-variable settings, which identify the base level and specify how to accumulate statistics over levels. `fvset base` declares the base level for each specified variable; the default for factor variables without a declared base level is the lowest value. `fvset design` specifies how the `margins` command is to accumulate over the levels of a factor variable. `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`.

**Quick start**

Set the base category of categorical variable `a1` to 3
```
fvset base 3 a1
```

Set the base category of `a2`, `a3`, and `a4` to each variable’s largest observed value
```
fvset base last a2 a3 a4
```

Set the base category of `a5` to the most frequent category
```
fvset base frequent a5
```

Set `a6` to have no base category
```
fvset base none a6
```

Restore the default base category (first) for `a5`
```
fvset base default a5
```

Specify that `margins` should treat `a2` as though it is balanced
```
fvset design asbalanced a2
```

Clear factor-variable settings for `a2` to `a4`
```
fvset clear a2-a4
```

List factor-variable settings for all factor variables
```
fvset report
```
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)
```

### base_spec

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

### design_spec

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

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 https://www.stata-press.com/data/r16/auto2
(1978 Automobile Data)
regress mpg i.rep78, baselevels
```

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>Number of obs =  69</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>549.415777</td>
<td>4</td>
<td>137.353944</td>
<td>F(4, 64) = 4.91</td>
</tr>
<tr>
<td>Residual</td>
<td>1790.78712</td>
<td>64</td>
<td>27.9810488</td>
<td>Prob &gt; F = 0.0016</td>
</tr>
<tr>
<td>Total</td>
<td>2340.2029</td>
<td>68</td>
<td>34.4147485</td>
<td>R-squared = 0.2348</td>
</tr>
</tbody>
</table>

```
mpg
Coef. Std. Err. t P>|t|  [95% Conf. Interval]
rep78
Poor       0 (base)
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       0.6666667 3.942718 0.17 0.866 -6.928014 8.261352
Excellent  6.363636 4.066234 1.56 0.123 -1.759599 14.48667
_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:

```
fvset base 2 rep78
regress mpg i.rep78, baselevels
```

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>Number of obs = 69</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>549.415777</td>
<td>4</td>
<td>137.353944</td>
<td>F(4, 64) = 4.91</td>
</tr>
<tr>
<td>Residual</td>
<td>1790.78712</td>
<td>64</td>
<td>27.9810488</td>
<td>Prob &gt; F = 0.0016</td>
</tr>
<tr>
<td>Total</td>
<td>2340.2029</td>
<td>68</td>
<td>34.4147485</td>
<td>Adj R-squared = 0.1869</td>
</tr>
</tbody>
</table>

```
mpg
Coef. Std. Err. t P>|t|  [95% Conf. Interval]
rep78
Poor       1.875 4.181884 0.45 0.655 -6.479274 10.22927
Fair       0 (base)
Average    0.308333 2.104836 0.15 0.884 -1.948621 7.031954
Good       2.541667 2.247695 1.13 0.262 -1.948621 7.031954
Excellent  8.238636 2.457918 3.35 0.001 3.32838 13.14889
_cons      19.125 1.870195 10.23 0.000 15.38886 22.86114
```

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
```

```
<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>Number of obs = 69</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>31824.2129</td>
<td>5</td>
<td>6364.84258</td>
<td>F(5, 64) = 227.47</td>
</tr>
<tr>
<td>Residual</td>
<td>1790.78712</td>
<td>64</td>
<td>27.9810488</td>
<td>Prob &gt; F = 0.0000</td>
</tr>
<tr>
<td>Total</td>
<td>33615</td>
<td>69</td>
<td>487.173913</td>
<td>R-squared = 0.9467</td>
</tr>
</tbody>
</table>

<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>

| mpg     | Coef.     | Std. Err. | t    | P>|t| | [95% Conf. Interval] |
|---------|-----------|-----------|------|------|----------------------|
| rep78   |           |           |      |      |                      |
| Poor    | 21        | 3.740391  | 5.61 | 0.000| 13.52771 28.47229    |
| Fair    | 19.125    | 1.870195  | 10.23| 0.000| 15.38886 22.86114    |
| Average | 19.43333  | .9657648  | 20.12| 0.000| 17.504 21.36267     |
| Good    | 21.6667   | 1.246797  | 17.38| 0.000| 19.1759 24.15743    |
| Excellent | 27.36364  | 1.594908  | 17.16| 0.000| 24.17744 30.54983   |

Example 2

By default, `margins` assumes that factor variables are to be treated as observed and accumulates a margin by using the observed relative frequencies of the factor levels or the sum of the weights if weights have been specified.

```
. 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>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  | .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

. margins

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)
```

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:

```plaintext
. fvset clear foreign
. regress mpg i.foreign
```

```
Source | SS   df    MS
----------|-------|------
Model    | 378.153515 | 1   | 378.153515
Residual | 2065.30594  | 72  | 28.6848048
Total    | 2443.45946  | 73  | 33.4720474

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

```
. margins
```

```
Predictive margins Number of obs = 74
Expression : Linear prediction, predict()

<table>
<thead>
<tr>
<th>Delta-method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Margin</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>_cons</td>
</tr>
</tbody>
</table>
```

```
. fvset design asbalanced foreign
. margins
```

```
Predictive margins Number of obs = 74
Expression : Linear prediction, predict()

<table>
<thead>
<tr>
<th>Delta-method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Margin</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>_cons</td>
</tr>
</tbody>
</table>
```

Stored results

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

Macros

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