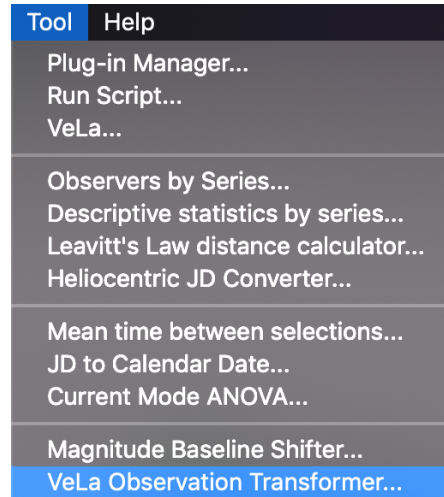# VeLa Observation Transformer Plug-In

This observation transformation plug-in is accessible from the `Tool` menu and allows observations in one or more series to be modified via a VeLa program.

In the following example, visual and Johnson V eta Aql observations have been loaded and a phase plot created. The plug-in has been selected from the `Tool` menu and both series selected in the dialog that opens.
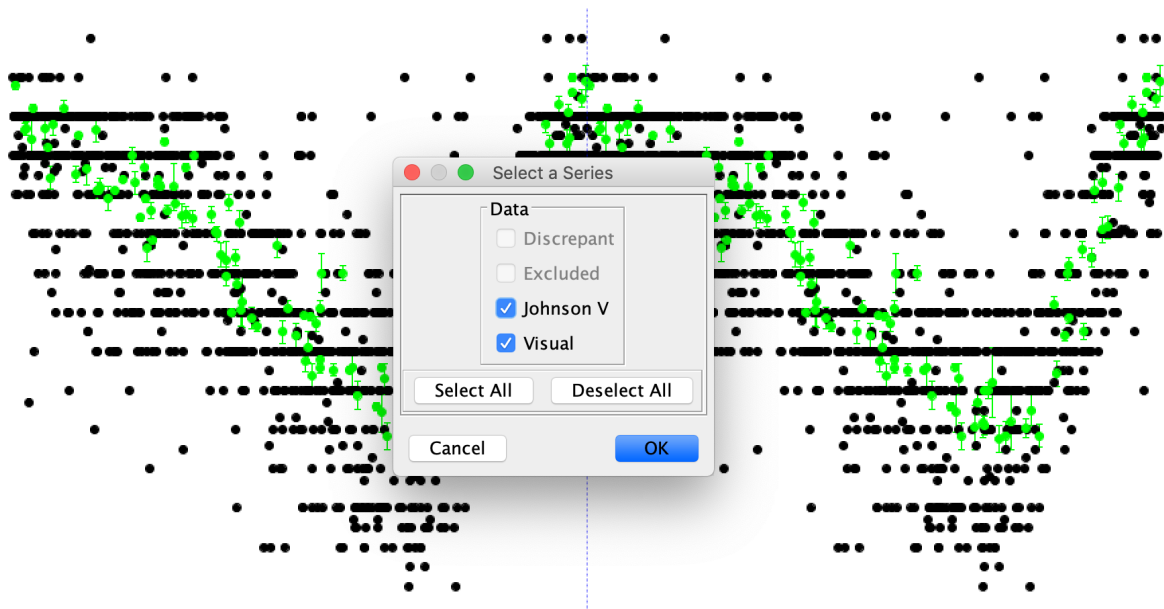
Clicking the `OK` button yields the general VeLa dialog, in which VeLa code can be entered tested:

Define VeLa function: do():list

VeLa Code

```
do() : list {
  err is uncertainty
  when
    obscode = "SAH" -> [mag+10 err]
    obscode = "BGZ" -> [mag+20 err]
    band = "Johnson V" -> [mag+30 err]
    true -> [mag err]
}
```

Output

| Cancel | Clear | Run | Load | Save | OK |

A parameterless function called `do` must be defined and it must return a list. This list must contain two values: magnitude and uncertainty. In future, the plug-in may be extended to allow other observation attributes to be returned. The function is applied to each observation and the returned magnitude and uncertainty values are used to modify the observation.

In the example shown above, if the observer code is SAH, the observation's magnitude will have 10 added to it. If the observer code is BGZ, the observation's magnitude will have 20 added to it. If the band is Johnson V, the observation's magnitude will have 30 added to it. Otherwise, the magnitude and uncertainty will be unchanged.
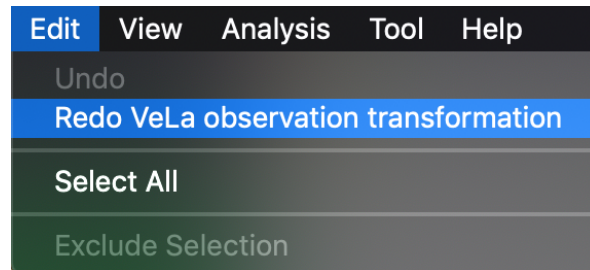
The result of running this code by clicking the `OK` button is as follows:

Phase Plot for eta Aql
06/10/2019 (database), period: 7.17679, epoch: 2436084.656
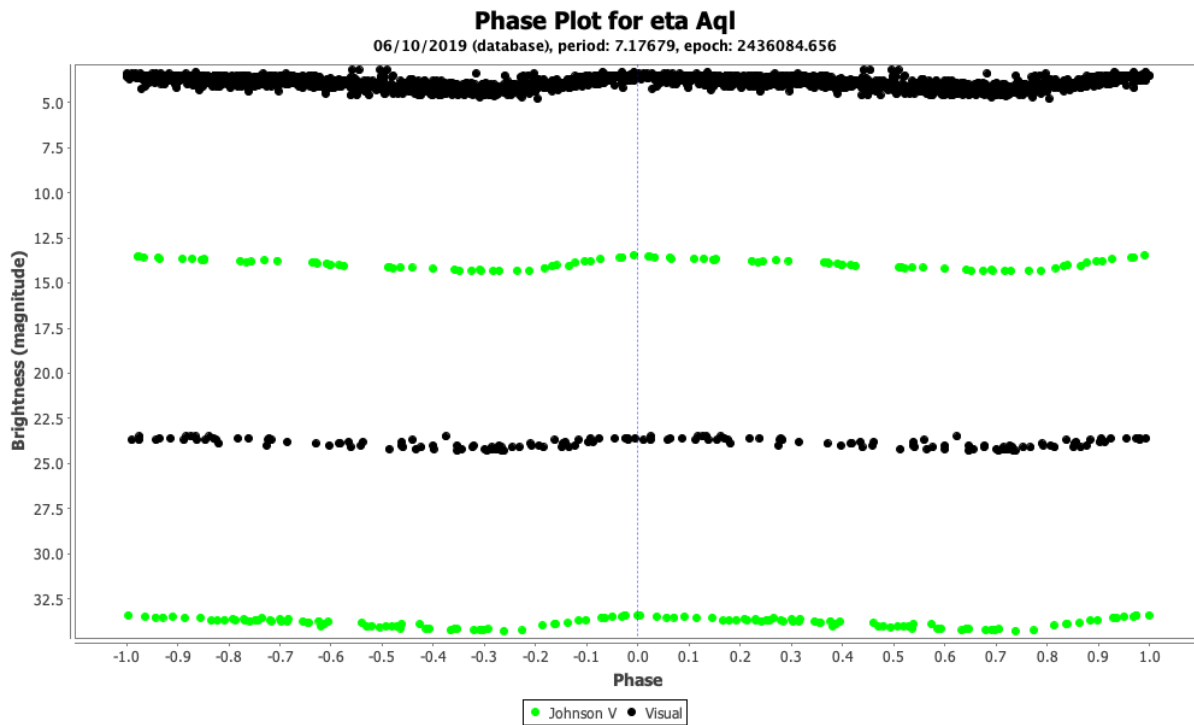
Selecting `Undo` from the `Edit` menu undoes this operation:
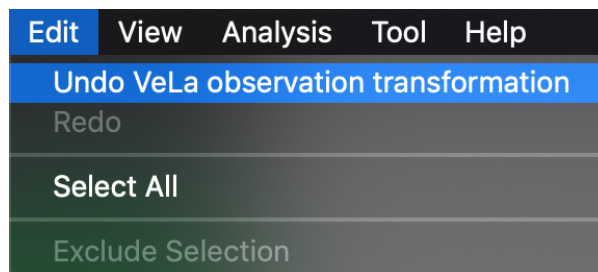
Selecting `Redo` from the `Edit` menu applies the original operation again.

This undo/redo cycle can be repeated. Using the plug-in again for a new VeLa observation transformation operation starts a new undo/redo cycle.

Here is an example of how a `do` function can be tested before being used for observation transformation:
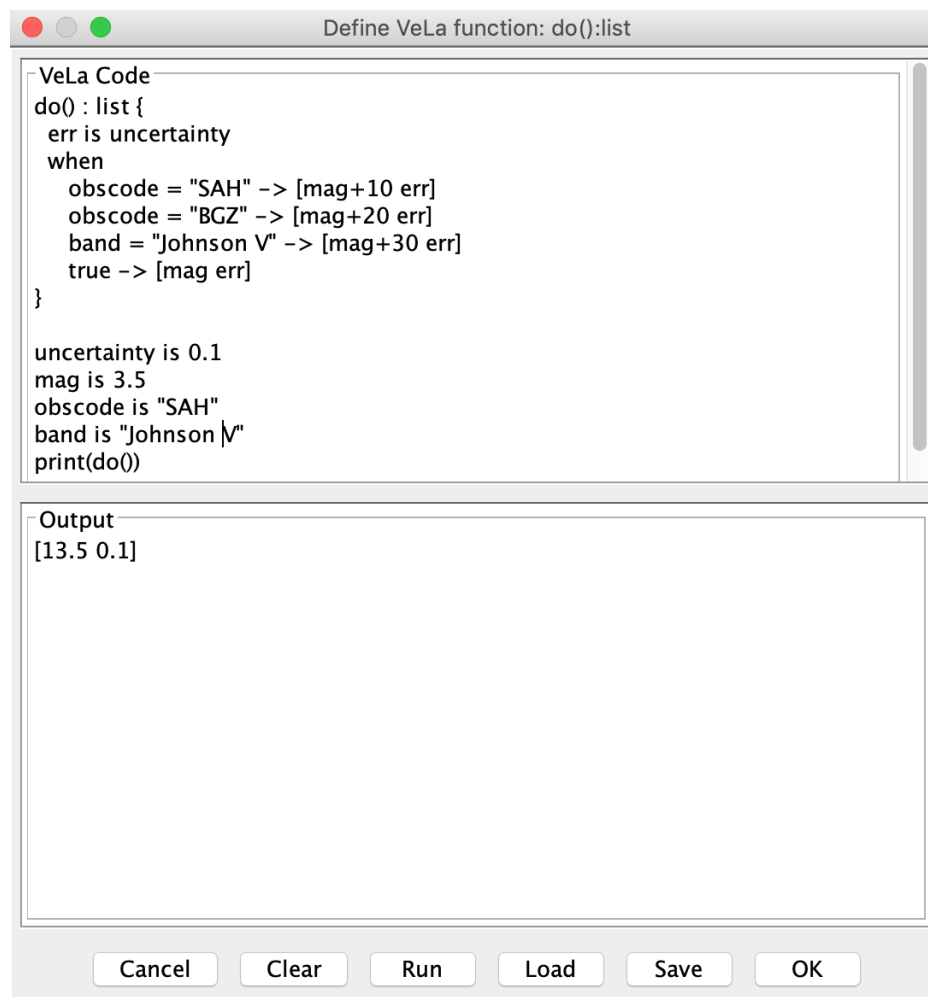
The `Run` button (instead of `OK`) is clicked to execute the code, generating the output shown.