

Package Structure

Bernd Funovits

Contents

1	Functions in Main Script	1
1.1	Initial Values	1
1.1.1	Additional Functions needed for Initial Values	2
1.2	Generating Results	3
1.2.1	Likelihood Factory	4
1.2.1.1	Filling a template	5
2	Other Functions for Analysis	5
3	Whole Package	6

1 Functions in Main Script

The following functions are called on the highest level within the script in `scriptsP_whf_revision_sgt_bq/aa_Rscript4slurm` which is run on the HPC cluster of the University of Helsinki as batch job via `scriptsP_whf_revision_sgt_bq/b_slurm`.

- `tmpl_whf_rev()`
- `get_init_armamod_whf_random()`
- `create_results_list()`

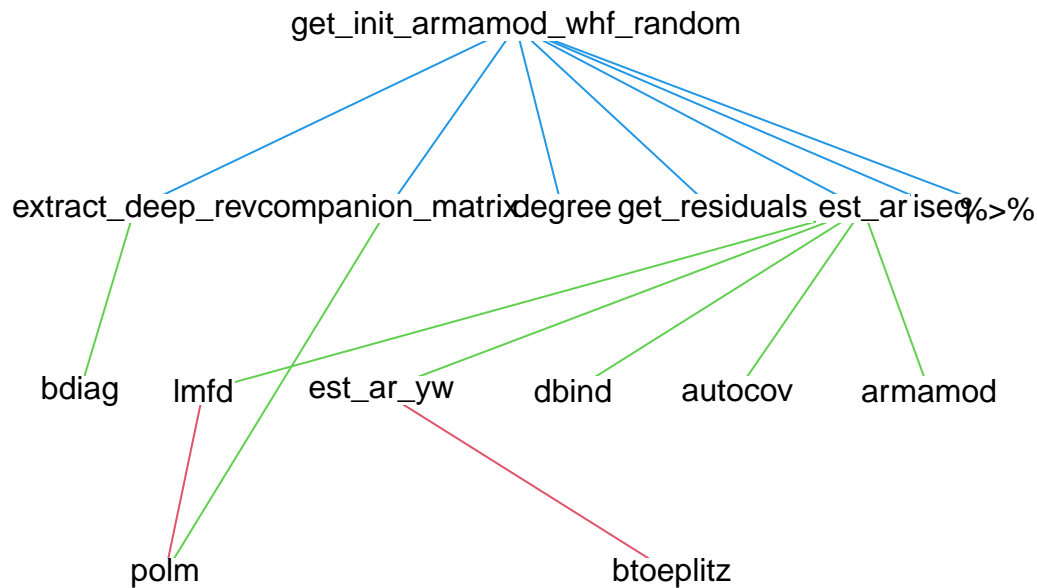
These functions are contained in the file starting with **a** in the package directory `svarmawhf/R/`.

There are no dependencies for **tmpl_whf_rev()** which just creates a template for a SVARMA-WHF model with given integer-valued parameters to be filled with the deep/free parameters.

1.1 Initial Values

The function `get_init_armamod_whf_random()`, which generates starting values for the deep parameters, has the following dependencies:

```
fw_get_init_armamod_whf_random =  
  foodweb(where = "package:svarmawhf", prune = "get_init_armamod_whf_random",  
          border = FALSE,  
          textcolor = "black",  
          cex = 1, charlim = 120, lwd = 1)
```



1.1.1 Additional Functions needed for Initial Values

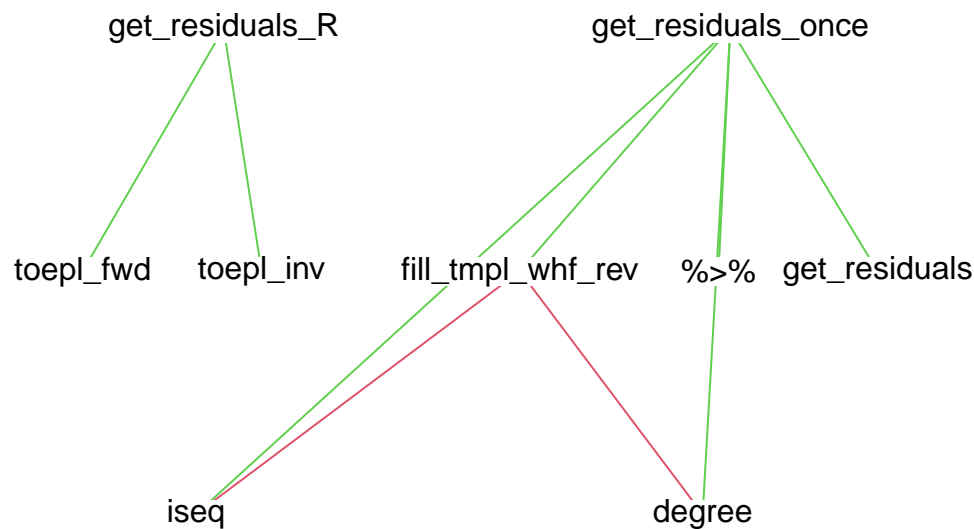
After obtaining AR estimates using the RLDM function `est_ar()`* and initial values for the forward and backward MA part, the parameters are extracted from the template with `extract_deep_rev()`.

Subsequently, we calculate the residuals/shocks with `get_residuals()`. This function is implemented in RcppArmadillo. However, an R, version which performs the same steps but is more easily readable, is available as well.

These functions are contained in the package directory `svarmawhf/R/b_*.R` as well as `svarmawhf/src/revision_armadillo_w`

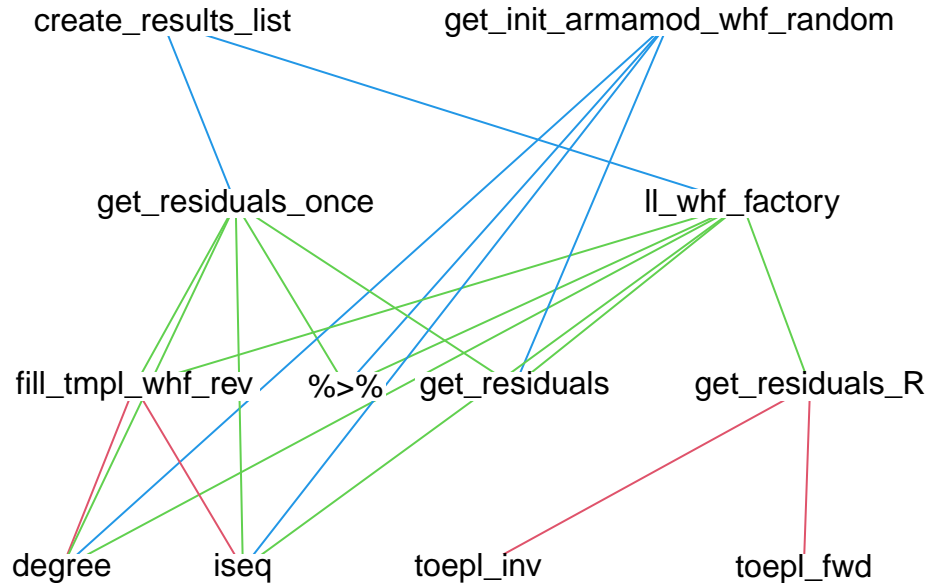
```

fw_get_residuals_noancestors =
  foodweb(
    where = "package:svarmawhf",
    prune = "get_residuals",
    border = FALSE,
    textcolor = "black",
    cex = 1,
    lwd = 1,
    ancestors = FALSE)
  
```



```
fw_get_residuals_noancestors =
  foodweb(
    where = "package:svarmawhf",
    prune = "get_residuals",
    border = FALSE,
    textcolor = "black",
    cex = 1, lwd = 1)

```



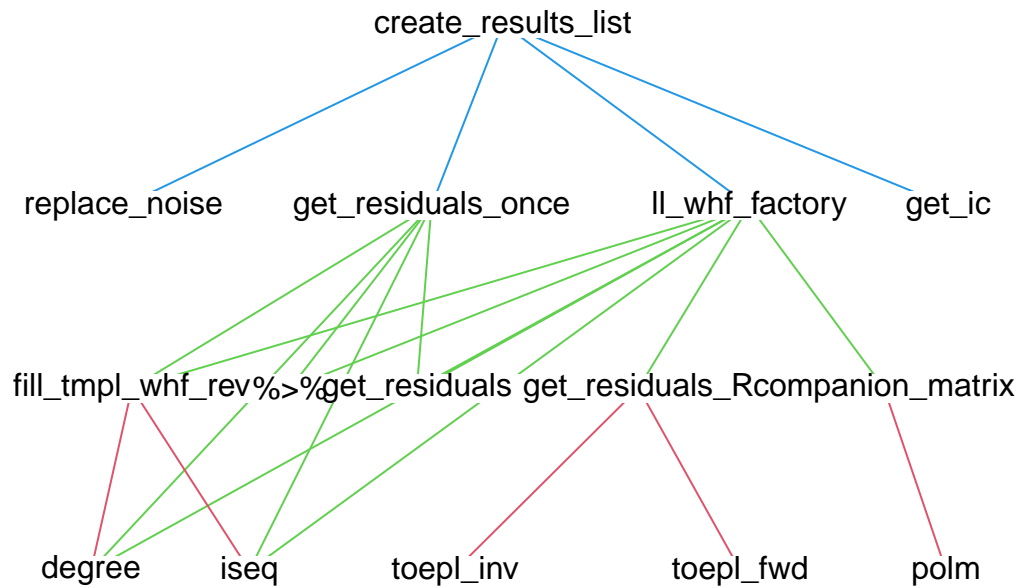
1.2 Generating Results

The quite complex function `create_results_list()` performs the heavy lifting and writes all necessary outputs in a tibble which is saved as an rds-file. In turn, the main work within `create_results_list()` is done by the likelihood factory `ll_whf_factory()`: A function which creates log-likelihood functions for the desired density and whose enclosing environment contains the dataset.

The other two functions (`get_ic()` and `replace_noise()`) are used to rotate and update the noise parameters to independence, if desired.

```
fw_create_results =
  foodweb(
    where = "package:svarmawhf",
    prune = "create_results_list",
    border = FALSE,
    textcolor = "black",
    cex = 1, lwd = 1)

```



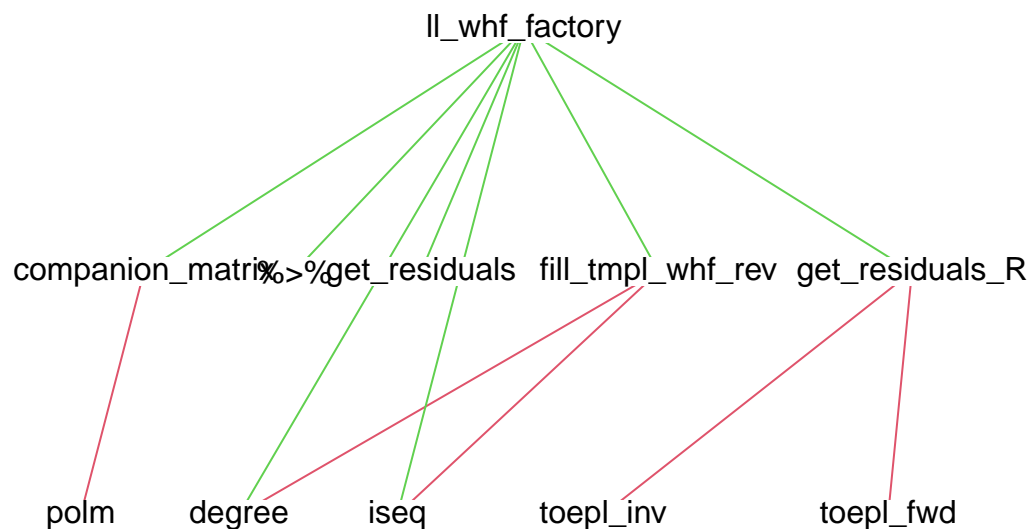
1.2.1 Likelihood Factory

The log-likelihood factory generates log-likelihood functions (closures, see <https://adv-r.hadley.nz/functions.html?q=closure#first-class-functions>) for the indicated type (laplace, Gaussian mixtures, normal distribution, SGT) and a particular SVARMA-WHF template. The data set is contained in the enclosing environment.

The likelihood function only depends on the deep parameters which are filled into the SVARMA-WHF template with `fill_template_whf()`. Subsequently, we obtain the residuals/shocks with `get_residuals()` and evaluate them with the provided density.

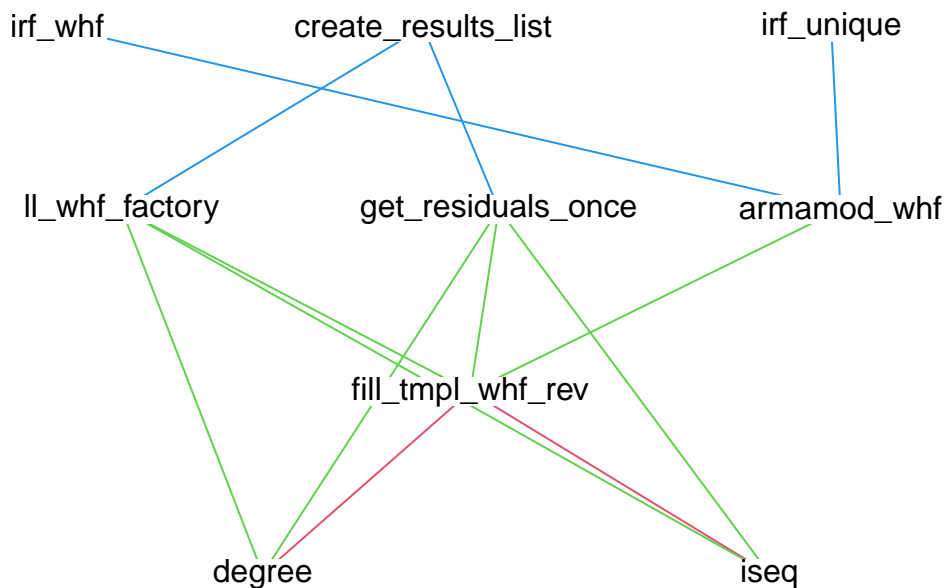
```

fw_ll_whf_factory =
  foodweb(
    where = "package:svarmawhf",
    prune = "ll_whf_factory",
    border = FALSE,
    textcolor = "black",
    cex = 1,
    charlim = 200,
    lwd = 1,
    ancestors = FALSE)
  
```

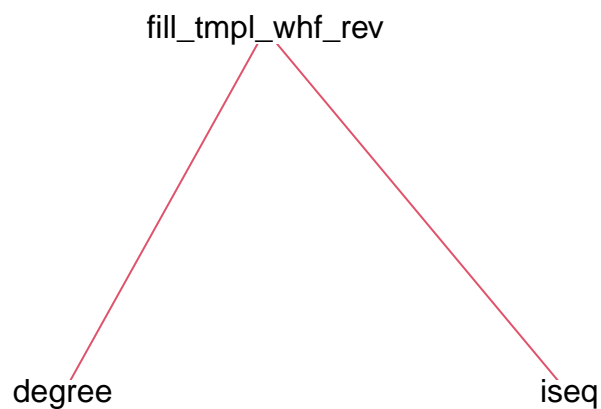


1.2.1.1 Filling a template The function `fill_tmpl_whf()` fills a given template with deep parameters. It is contained in the file starting with `c_`.

```
foodweb(where = "package:svarmawhf", rprune = "fill_tmpl_whf",
        border = FALSE,
        textcolor = "black",
        cex = 1, lwd= 1)
```



```
foodweb(where = "package:svarmawhf", rprune = "fill_tmpl_whf",
        border = FALSE,
        textcolor = "black",
        cex = 1, lwd= 1,
        ancestors = FALSE)
```



2 Other Functions for Analysis

Some other functions are only needed for subsequent IRF analysis:

- `armamod_whf()`: Transform the WHF to a usual $b(z)$ polynomial matrix
- `irf_whf()`: Obtain IRF
- `irf_unique()`: Permute and scale B such that the identifying restriction in Section 3 regarding (B, Σ) holds.

- `irf_change()`: Permute and scale B as desired.

Some are small helpers:

- Converting between the forward polynomial $f(z)$ and $g(z) = s(z)f(z)$:
 - `f2g()` and `g2f()`
- Toeplitz calculations for obtaining residuals in R (instead of Rcpp):
 - `toeplitz_fwd()` and `toeplitz_inv()`

3 Whole Package

Due to the inclusion of functions from the quite sophisticated packages **rationalmatrices** and **RLDM**, visualising leads to a complex graph.

```
fw_all =
  foodweb(
    where = "package:svarmawhf",
    border = FALSE,
    textcolor = "black",
    cex = 1, charlim = 120, lwd = 1)

```

