

# *Apollo* versions: timeline, changes and backwards compatibility

## Version 0.0.6 (13 March 2019)

This is the first fully functioning release of *Apollo*.

## Version 0.0.7 (8 May 2019)

### *Changes:*

#### **General**

Minor improvements to efficiency, stability and reporting of user errors.

#### **Bayesian estimation produces `model$estimate`**

Functions affected: `apollo_estimate`, `apollo_prediction`, `apollo_llCalc`

Detailed description: until version 0.0.6, Bayesian estimation in *Apollo* did not produce a `model$estimate` output. We have retained the various existing outputs, but in addition, `model$estimate` is now produced, combining non-random parameters with individual specific posteriors for random parameters. This now allows the user to use `apollo_prediction` and `apollo_llCalc` on such outputs, where care is of course required in interpretation of outputs based on posterior means.

Backwards compatibility of code: no backwards compatibility issues for existing functions

Backwards compatibility of examples: Prediction component added to `apollo_example_26.r` which only works from version 0.0.7 onwards

#### **Scaling of parameters during model estimation**

Functions affected: `apollo_estimate`

Detailed description: scaling of model parameters can be used during estimation

Backwards compatibility of code: no backwards compatibility issues for existing functions

Backwards compatibility of examples: Scaling implemented in `apollo_example_12.r` which only works from version 0.0.7 onwards

#### **Inputs changed for `apollo_choiceAnalysis`**

Functions affected: `apollo_choiceAnalysis`

Detailed description: inputs changed so function can be called prior to `apollo_validateInputs`

Backwards compatibility of code: function call changed from version 0.0.7 onwards

Backwards compatibility of examples: Function used in `apollo_example_1.r` and `apollo_example_2.r` where this part of the examples only works from version 0.0.7 onwards

#### **Constraints for classical estimation**

Functions affected: `apollo_estimate`

Detailed description: *Apollo* now allows the user to include a list called `constraints` in `estimate_settings` for use with BFGS for classical model estimation.

Backwards compatibility of code: no backwards compatibility issues for existing functions

Backwards compatibility of examples: no backwards compatibility issues for examples

### **Validation output**

Functions affected: `apollo_estimate`

Detailed description: *Apollo* no longer reports that all pre-estimation checks were passed for a model component and instead only reports if there are an issues.

Backwards compatibility of code: no backwards compatibility issues for existing functions

Backwards compatibility of examples: no backwards compatibility issues for examples

### **Outputs changed for `apollo_choiceAnalysis`**

Functions affected: `apollo_choiceAnalysis`

Detailed description: outputs changed so that t-test value is reported instead of p-value, and order of outputs is changed

Backwards compatibility of code: outputs changed from version 0.0.8 onwards, but function called in the same way

Backwards compatibility of examples: no backwards compatibility issues for examples

### ***Bug fixes:***

***apollo\_speedTest***

This function was unintentionally hidden from users in previous versions

## **Version 0.0.8 (9 September 2019)**

### ***Changes:***

#### **General**

Minor improvements to efficiency, stability and reporting of user errors.

### **Outputs changed for `apollo_choiceAnalysis`**

Functions affected: `apollo_choiceAnalysis`

Detailed description: outputs changed so that t-test value is reported instead of p-value, and order of outputs is changed

Backwards compatibility of code: outputs changed from version 0.0.8 onwards, but function called in the same way

Backwards compatibility of examples: no backwards compatibility issues for examples

### **Ability to define estimation/validation subsets for `apollo_outOfSample`**

Functions affected: `apollo_outOfSample`

Detailed description: the user can now provide a matrix or data.frame describing which observations are to be used in the estimation and validation subsets

Backwards compatibility of code: optional argument added from version 0.0.8 onwards, but function called in the same way

Backwards compatibility of examples: no backwards compatibility issues for examples

### **Bootstrap estimation added**

Functions affected: `apollo_bootstrap`, `apollo_estimate`

Detailed description: the user can now perform bootstrap estimation. This can also be called directly with `apollo_estimate` during estimation

Backwards compatibility of code: new function from version 0.0.8 onwards, new optional arguments for `apollo_estimate`, but function called in the same way

Backwards compatibility of examples: no backwards compatibility issues for examples

### **Chosen unavailable alternatives have a likelihood of zero**

Functions affected: `apollo_mnl`, `apollo_mdcev`, `apollo_mdcnev`

Detailed description: MNL, MDCEV and MDCNEV now return a likelihood equal to zero for chosen alternatives that are not available. This change is only relevant if `apollo_control$noValidation` is TRUE.

Backwards compatibility of code: new likelihood values for unavailable chosen alternatives on MNL, MDCEV and MDCNEV models from version 0.0.8 onwards

Backwards compatibility of examples: no backwards compatibility issues for examples

### **Allow user to specify number of outliers to report**

Functions affected: `apollo_modelOutput`, `apollo_saveOutput`

Detailed description: In addition to specifying TRUE/FALSE for `printOutliers`, the user can provide the number of outliers to report (instead of the default of 20).

Backwards compatibility of code: optional argument added from version 0.0.8 onwards, but function called in the same way

Backwards compatibility of examples: no backwards compatibility issues for examples

### **Allow user to use subset of rows for analysis of choices**

Functions affected: `apollo_choiceAnalysis`

Detailed description: An additional `rows` argument can be entered into `choiceAnalysis_settings`.

Backwards compatibility of code: optional argument added from version 0.0.8 onwards, but function called in the same way

Backwards compatibility of examples: no backwards compatibility issues for examples

### **Allow user to change name and location of outside good in MDCEV and MDCNEV**

Functions affected: `apollo_mdcev` and `apollo_mdcnev`

Detailed description: An additional `outside` argument can be entered into `mdcev_settings` and `mdcnev_settings` with the name of the outside good which can now differ from `outside`. It also no longer needs to be in first position in the list of alternatives.

Backwards compatibility of code: optional argument added from version 0.0.8 onwards, but function called in the same way

Backwards compatibility of examples: no backwards compatibility issues for examples

### **No need to define superfluous $\gamma$ for outside good in MDCEV and MDCNEV**

Functions affected: `apollo_mdcev` and `apollo_mdcnev`

Detailed description: The user no longer needs to create a `gamma` term for the outside good.

Backwards compatibility of code: function called in the same way

Backwards compatibility of examples: no backwards compatibility issues for examples

### **Individual IDs and choice scenario numbers added in predictions**

Functions affected: `apollo_prediction`

Detailed description: The output from `apollo_prediction` now includes the IDs and choice scenario numbers as the first two columns.

Backwards compatibility of code: function called in the same way

Backwards compatibility of examples: no backwards compatibility issues for examples, except for shift in some columns

### ***Bug fixes:***

*`apollo_firstRow`*

This function mistakenly replicated the first row for each person  $T_n$  times

### **HB estimation**

HB estimation failed in earlier versions for models without any random parameters

### ***apollo\_combineResults***

This function failed in earlier versions when using only a single model

## **Version 0.0.9 (23 October 2019)**

### ***Changes:***

#### **General**

Minor improvements to efficiency, stability and reporting of user errors.

#### **Pre-estimation tests to ensure all parameters affect likelihood function**

Functions affected: `apollo_estimate`

Detailed description: unless `apollo_control$noDiagnostics==TRUE`, a pre-estimation check is used to ensure that there are no parameters in `apollo_beta` for which changes do not lead to changes in the model likelihood

Backwards compatibility of code: function called in the same way

Backwards compatibility of examples: no backwards compatibility issues for examples

#### **Additional diagnostic message for HB estimation**

Functions affected: `apollo_estimate`

Detailed description: the RSGHB package used for Bayesian estimation left censors likelihood values at the individual level to avoid numerical issues. This has the undesired side effect of mis-specified models still running, and a warning message is now displayed when censoring has been used

Backwards compatibility of code: function called in the same way

Backwards compatibility of examples: no backwards compatibility issues for examples

### ***Bug fixes:***

#### **HB estimation**

HB estimation failed in earlier versions for models without any random parameters if using scaling

### ***apollo\_deltaMethod***

This function had a small error in the calculation for standard errors for logistic transforms

## **Version 0.1.0 (under development)**

### ***Changes:***

#### **General**

Minor improvements to efficiency, stability and reporting of user errors.

#### **Ability to sort results by date**

Functions affected: `apollo_combineResults`

Detailed description: an additional option `sortByDate` has been included. When set to `TRUE`, the models in the summary file will be sorted by the date when the model was estimated (default set to `TRUE`)

Backwards compatibility of code: function called in the same way

Backwards compatibility of examples: no backwards compatibility issues for examples

***Bug fixes:***

**Prediction from latent class**

Predictions from latent class were missing the names of the alternatives in the output

**Model validation**

Earlier versions of the code still performed model validation even if `apollo_control$noValidation==TRUE`

***apollo\_combineResults***

When `combineResults_settings$modelNames` was not provided, other settings were ignored

***apollo\_mdcev***

When the `rows` option was used, any pre-estimation checks still included all rows in the data.

In addition, some failures could occur in estimation.

***apollo\_estimate***

The pre-estimation diagnostic tests would fail in case the starting values were the same for multiple classes in a latent class model