HS3 · 30 OCTOBER 2026

HS3 pull requests to merge for the next release

Seven pull requests, what they do, what was discussed, and where each one stands.

Why now

EARLY NEXT WEEK, 2 TO 4 NOV

Spec PRs agreed and merged

13 NOV

ROOT implementation done, one week of validation

20 NOV

ATLAS Higgs models uploaded

On the list

#129 histfactory function · #126 axes refactor · #124 HistFactory sync · #119 combined data sets · #109 array constraints · #134 and #135 v0.3 fixes

For each: **what it does**, **what was discussed**, **where it stands**.

PR #129

The histfactory function

DOES

- Factors the HistFactory prediction out of histfactory_dist into a standalone histfactory function
- Moves the HistFactory text into an appendix

DISCUSSED

- Nothing open
- Needed by the derived-data example in discussion #88 and by issue #136

CURRENT STATE

- No changes requested so far; first in the merge order, #124 rebases on it

PR #126

Axes refactor

DOES

- Axis definitions move to an appendix with three self-contained types: continuous, binned, categorical
- Every place that uses axes says which types it allows
- Carries the data-chapter fixes from #134 and #135: bin-boundary rule, flattening order, order of categorical values, structural rule

DISCUSSED

- Can edges alone replace min and max on a binned axis? Split out as issue #137
- Should the bin-count distributions allow categorical axes at all?

CURRENT STATE

- With min and max still required, pending #137
- #137 decided afterwards
- The categorical-axes question answered in the text first

PR #124

HistFactory chapter sync

DOES

- Aligns histfactory_dist with the rest of the standard and with ROOT
- Named parameters and constraints, a custom modifier, optional errors
- Clarifies what a sample's data means

DISCUSSED

- How staterror and shapesys constraints are encoded after the physics-statistics split: issue #136, options A to E
- Benched for now: not part of this merge

CURRENT STATE

- After #135 and #129, rebased
- staterror and shapesys as they are; revisited when #136 is picked up again

PR #119 · 1 of 4 · the problem

Selecting by the value of a categorical variable

WHAT WAS MISSING

- A multi-channel fit pairs one model per region with one data set per region. HS3 had no way to bundle several data sets into one and to say which sub-model goes with which
- Discrete profiling needs a distribution that switches between alternative shapes by a discrete parameter
- Without a common layout, every export encoded regions differently

WHAT THE PR ADDS

- Categorical variables, declared once in the domains
- A **key**: one label per categorical variable in the order of columns, selecting a component; combinations that do not occur are simply absent
- Four structures that all use the same components-and-key layout: combined data set, named_joint_dist, category_conditional_dist, categorical function

Next: the combined data set, the named joint distribution, the two selectors by parameter, then discussion and status.

PR #119 · 2 of 4 · the combined data set

One data set made of several

- type combined, with columns: the names of the categorical variables
- components: one entry per key, each naming a sub-data set
- The sub-data sets are ordinary data sets of any type, declared alongside; they may share axes or not
- The combined data set itself carries no axes: the key is the only thing it adds
- Not every key needs an entry. Here (VR, global) is absent

```
"data":[
  {
    "name":"combined_data",
    "type":"combined",
    "columns":["regions","observables"],
    "components":[
      {"key":["SR","physics"], "dataset":"data_sr"},
      {"key":["CR","physics"], "dataset":"data_cr"},
      {"key":["VR","physics"], "dataset":"data_vr"},
      {"key":["SR","global"],
"dataset":"global_obs_sr"},
      {"key":["CR","global"], "dataset":"global_obs_cr"}
    ]
  },
  {"name":"data_sr", "type":"binned", ...},
  {"name":"data_cr", "type":"unbinned", ...},
  ...
]
```

PR #119 · 3 of 4 · the named joint distribution

One sub-model per key

- Same columns as the paired combined data set, in the same order
- components maps each key to a sub-distribution, evaluated on the sub-data set with that key
- Sub-distributions may have different variates, or share them: the key is part of the outcome, so each event belongs to exactly one component
- If every component is extended: a joint distribution over the variates and the categorical variables, whose values are drawn in proportion to the expected counts
- Otherwise: a conditional density per key; the categorical values are given by the data, not sampled
- A key with no sub-data set contributes nothing; an empty sub-data set still contributes

```
{
  "name":"channels",
  "type":"named_joint_dist",
  "columns":["regions","observables"],
  "components":[
    {"key":["SR","physics"], "dist":"sr"},
    {"key":["CR","physics"], "dist":"cr"},
    {"key":["VR","physics"], "dist":"vr"},
    {"key":["SR","global"], "dist":"global_sr"},
    {"key":["CR","global"], "dist":"global_cr"}
  ]
}
```

Paired in a likelihood with combined_data from the previous slide: sr is evaluated on data_sr, cr on data_cr, and so on.

PR #119 · 4 of 4 · selecting by a parameter

Category-conditional distribution and categorical function

category_conditional_dist

- Selector is a categorical **parameter**, not an observable: categorical_parameter
- Evaluates the sub-distribution for the current value, conditional on it; never a distribution over the parameter
- All sub-distributions describe the same variate: alternative shapes of one thing
- The use case: discrete profiling over background functions

```
{
  "name":"bkg_choice",
  "type":"category_conditional_dist",
  "categorical_parameter":["bkg_function"],
  "components":[
    {"key":["poly1"], "dist":"bkg_poly1"},
    {"key":["poly2"], "dist":"bkg_poly2"},
    {"key":["exp1"], "dist":"bkg_exp1"}
  ]
}
```

categorical function

- The same idea for functions: categorical_variable names the selector, components maps each key to a function
- Lets a yield, a scale factor or a nuisance mapping differ per region without duplicating the model
- Together the four structures use one vocabulary: columns or a selector field, then components with key

```
{
  "name":"lumi_per_region",
  "type":"categorical",
  "categorical_variable":["regions"],
  "components":[
    {"key":["SR"], "function":"lumi_sr"},
    {"key":["CR"], "function":"lumi_cr"}
  ]
}
```

PR #119 · discussion and status

Combined data sets, selection by categorical variables

DISCUSSED

- When is named_joint_dist a joint distribution over the categorical variables? Only if every component is extended. Otherwise their values are conditioned on, cannot be sampled, and the mixed case is a valid likelihood but not a distribution
- Each extended component keeps its own Poisson term; an empty sub-data set still contributes, a missing key does not
- Same field name, columns, in data set and distribution
- Name: named_joint_dist or simultaneous_dist

CURRENT STATE

- The open points raised in review are being addressed shortly
- Merges with that review round
- Name still open; default named_joint_dist

PR #109 · 1 of 4 · the problem

Constraint terms, by the hundred

WHAT WAS MISSING

- Nuisance parameters are constrained by auxiliary measurements: one Gaussian or Poisson term per parameter, with the measured value as a fixed global observable
- A realistic model has hundreds of them, all of the same shape
- Today each one is a separate distribution plus a separate point data set, or they are multiplied into the channel model where their multiplicity is ambiguous

WHAT THE PR ADDS

- array_dist: one distribution standing for n independent one-dimensional distributions of one type, one per named observable
- Paired with one unbinned data set that holds all global observables as one entry
- An appendix, Auxiliary measurements and global observables, comparing the three ways to write constraints and recommending this one

Next: the structure, the paired data set, then discussion and status.

PR #109 · 2 of 4 · the structure

array_dist

The density is a product over the components: each factor is the distribution of type **of**, evaluated at its own observable x with its own parameters.

- **of**: the type of every factor, for example gaussian_dist or poisson_dist. It fixes which parameters each component needs
- **components**: one entry per factor, with x, the name of its observable, plus the parameters of that type
- **default_parameters**: values or names applied to every component that does not set them itself, for example a common sigma
- The observables must be pairwise distinct, and a component's parameters may not depend on another component's observable; shared model parameters are fine
- Equivalent to a product_dist of n single-type factors, but flat and explicit so engines can vectorize

```
{
  "name":"gauss_constraints",
  "type":"array_dist",
  "of":"gaussian_dist",
  "default_parameters":{"sigma":1.0},
  "components":[
    {"x":"nom_alpha_1", "mean":"alpha_1"},
    {"x":"nom_alpha_2", "mean":"alpha_2"},
    {"x":"nom_alpha_3", "mean":"alpha_3",
     "sigma":"alpha_3_sigma"}
  ]
}
```

Three unit Gaussians centred on the nuisance parameters alpha_1 to alpha_3, the third with its own width. Reads as: nom_alpha_i is Gaussian around alpha_i.

PR #109 · 3 of 4 · the paired data set

All global observables in one entry

- One unbinned data set, one continuous axis per component, named like the component's x
- Exactly one entry: the measured values of the global observables
- The likelihood pairs the array distribution with this data set, next to the channel model and its data
- One entry means each constraint term enters the likelihood exactly once; no per-event multiplicity question
- Poisson constraints work the same way: of poisson_dist, the observed counts as the entry, and the means given as functions of the nuisance parameters

```
"data":[
  {
    "name":"global_observables",
    "type":"unbinned",
    "entries":[[0.0, 0.0, 0.0]],
    "axes":[
      {"name":"nom_alpha_1", "min":-5, "max":5},
      {"name":"nom_alpha_2", "min":-5, "max":5},
      {"name":"nom_alpha_3", "min":-5, "max":5}
    ]
  }
],
"likelihoods":[
  {
    "name":"likelihood1",
    "distributions":["channel1_pdf", "gauss_constraints"],
    "data":["channel1_data", "global_observables"]
  }
]
```

PR #109 · 4 of 4 · discussion and status

Compact constraint terms

DISCUSSED

- The first design indexed the constraints by a categorical axis; that index is gone, components are listed by observable name
- Independence has to be explicit: distinct observable names alone do not make the factors independent when parameters may reference other components' observables
- Does a constraint with a fixed global observable survive normalization inside a product_dist? Now stated: factors are normalized one by one, so constant factors stay
- Name: array_dist or homogeneous_product_dist; the field of, element_type or factor_type

CURRENT STATE

- The open points raised in review are being addressed shortly
- Merges with that review round
- Name and field name as array_dist and of so far

PR #134 and #135

The v0.3 bundle

DOES

- #134 editorial, changes no valid document
- #135 normative fixes: plural parameters_of_interest and domains, hs3_version as a string, optional domain bounds, Crystal Ball field sets, string values for categorical parameters, identifier grammar, optional errors, no implicit interpolation, shapefactor is not shapesys

DISCUSSED

- Identifier grammar: ROOT-generated names already violate it, so a must only for names used in expressions
- The non-normative label on the introduction
- Edges without min and max, moved to #137

CURRENT STATE

- #134 with its data-chapter items moved to #126
- #135 with those three trims, before #124

Open points

Where	Question	Default
#109	Name: array_dist or homogeneous_product_dist; field of, element_type or factor_type	array_dist with of
#119	Name: named_joint_dist or simultaneous_dist	named_joint_dist
#126 / #137	May edges replace min and max on a binned axis?	min and max stay required; #137 later
#126	Categorical axes on the bin-count distributions?	Answered in the text before merge
#135	Identifier grammar: must for every name, or only for names in expressions?	Must only in expressions, should elsewhere

THE ASK

Anything you disagree with? Comment on the PR in the coming days!