



Input for Hierarchical Construction

Repository Name: arviz,

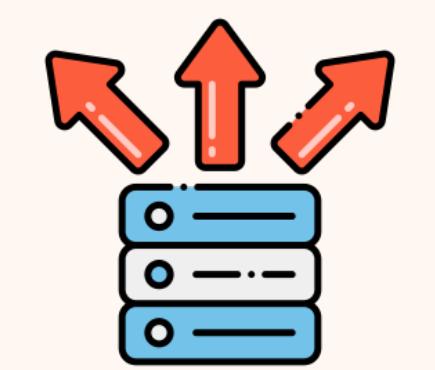
Repository Info: The ArviZ repository is a comprehensive Python package designed for the exploratory analysis of Bayesian models. ...

Parsed Semantic Features:

```
{  
    "visualize posterior predictive checks": ["configure  
    animations for posterior predictive checks", "visualize  
    posterior predictive statistical checks"],  
    "mcmc_diagnostic_tools": ["compute simulation standard  
    error accounting for autocorrelation", "compute rank-  
    normalized R-hat diagnosis", "..."],  
}
```

Functional Areas: ["DataProcessing", "DataConversion",
"ModelIntegration", "StatisticalDiagnostics",
"Visualization", "BayesianEvaluation",
"ConfigurationManagement", "UtilityFunctions"]

Task: Please analyze the above information and propose
how to reorganize the top-level feature groups



LLM Response

<think>

I mapped each parsed feature group to the 8 functional areas by its primary intent and output: visualization - centric groups (PPC, rank diagnostics, Bayesian diagnostic plots) were consolidated under Visualization with a “plot generation/statistical visualizations” sub-bucket, while time-oriented plots were separated into “plot generation/time series analysis” to keep concerns distinct. ...

</think>

<action>

```
{ "Visualization/plot generation/statistical visualizations":  
    ["visualize posterior predictive checks", "visualize rank  
    diagnostics", "visualize bayesian diagnostics"],  
    "Visualization/plot generation/time series analysis":  
    ["visualize temporal trends"],  
    "StatisticalDiagnostics/model evaluation/convergence  
    diagnostics": "mcmc_diagnostic_tools", "ecdf confidence  
    band computation"]}
```

</action>

Response →