

Statistical and Data Analysis Package in SWIFT

Claude A. Pruneau
Department of Physics and Astronomy
Wayne State University

What is SWIFT? Why use Swift?

- New flagship language of APPLE Computer,
- Compiled object-oriented language similar to C++ but with the **coding simplicity of a scripting language**.
- Interoperability with C, Objective-C, and C++ code,
- Truly comprehensive debugging and documentation features.
- Many language features that make for **rapid / effective development of robust, easily debugable & maintainable software libraries + APPs**.
- SWIFT is **open source** & available under **APACHE** license 2.0 for development on **APPLE and LINUX** platforms.
- Extension Construct enables User extension of any existing classes.

Basic Design and Components

- All components defined as Abstract Interfaces implemented as **SWIFT Protocol**.
- **Single class inheritance but multiple protocol implementation** provide for **integration of all basic features**.
 - E.g., PDFs evaluation, moments, differentiation, integration, random number generation.
- Functions
 - Basic and Special Functions
 - Function Derivatives
 - Numerical Integration
 - Numerical Interpolation
 - Root Finding
- Function Fitting and Optimization
- Random Number Generation
- Probability
 - Wide set of PDFs and Tools (for statistical tests)
- Statistics
 - Robust Moments, Histograms (1D-nD), Profile Histograms, Statistical Tests
 - Basic Bayesian Statistic Components
- Linear Algebra Module
- Task and Work Flow Control
- Logging Module
- Particle Physics Module
- Note: NOT ALL classes and components of the package can be represented on this poster.

Linear Algebra

- Vectors
- Matrices
- Rotation2D, Rotation3D, Square Matrices, *Sparse Matrices*,
- Jacobi Transformation,
- LUP Decomposition

Data Representation

- Data Points
- N-Tuples
- Trees
- Maps
- Dictionaries

Statistics

- Robust Moments Calculators
- Histogram (1D, nD)
- Profile Histogram (1D, nD)
- Correlation Functions
- Statistic Tests (Frequentist)

Tasks and Workflow Control

Error Logging

Fitting, Optimization

- ```

Filter
var nDimensions: Int { get }
var nParameters: Int { get }
var nFreeParameters: Int { get }
var nDataPoints: Int { get }
var nUsedDataPoints: Int { get }
var dof: Int { get }
var dataPoints: DataPointCollection { get set }
var fitFunction: ModelFunction { get set }
var objectiveFunction: ObjectiveFunction { get set }
var modelParameters: ModelParameters { get }
var status: FitStatus { get }
var covMatrixStatus: CovarianceMatrixStatus { get }
var hasCovMatrix: Bool { get }
var covMatrix: Matrix? { get }
var chi2: Double { get }
var iterations: Int { get }
var nFunctionCalls: Int { get }
var hessianMatrix: SquareMatrix { get }

```

- ```
Optimizer
• var nDimensions : Int { get }
• var function : ModelFunction { get set }
• var modelParameters : ModelParameters { get }
• var status : OptimizerStatus { get }
```

-

- OptimizerBracketFinder

- OptimizerHillClimbing

- Supervisor's Signature _____

-
- ```

graph LR
 A[OptimizerSimplex] --> B[OptimizerSimplex]

```

- 11

- Generic**

- more

## Abstract

- We present the design and implementation of an Object-Oriented Math + Statistics package designed for rapid and robust data analysis, with a comprehensive suite of statistical tools, fitting tools, and modeling functions. The package design is centered on abstract interfaces (Swift Protocols) and comprehensive + carefully designed class structures. The package features single/multi-dimensional functions, including common functions such as Bessel functions, Laguerre, and Legendre polynomials, classes for vectors, matrices and related linear algebra tools, a limited set of physics tools including rotations, Lorentz vectors, etc, multi-dimensional histograms, fast and robust moments calculation, calculation of correlation functions, frequentist statistical tests, maximum likelihood and least square fits, and extensible random number generation tools, as well as basic plotting capabilities.
- Given SWIFT's interoperability with other languages, the presented package should be easy to integrate within existing computing environments such as ROOT.

## Functions, PDFs, Random Numbers

