

Test Plan for HCVisR Package

Overview

The HCVisR package provides an interactive Shiny app for generating and visualizing time series data in the $H \times C$ plane. The app supports stochastic and deterministic time series generation, allows “mixing” two time series with addition or multiplication, and visualizes points in the $H \times C$ plane using Shannon Entropy (H) and Statistical Complexity (C), relying on the StatOrdPattHxC package for ordinal pattern analysis.

This test plan covers installation, functionality, input validation, error handling, and performance testing to ensure robustness, usability, and accuracy.

Installation

The steps to download the package and install it are the following:

```
install.packages('remotes')
library(remotes)
remotes::install_github("alexkzw/HCVisR")
```

```
library(HCVisR)
```

```
## Warning: replacing previous import 'shiny::runExample' by 'shinyjs::runExample'
## when loading 'HCVisR'
```

```
## Warning: replacing previous import 'dplyr::filter' by 'stats::filter' when
## loading 'StatOrdPattHxC'
```

Usage

Time Series Generation

Confirm that the app correctly generates time series data for different models

```
ts_wn <- new_stochastic_ts("WN", n = 300)
ts_ar <- new_stochastic_ts("AR", n = 300, phi = 0.5)
ts_arma <- new_stochastic_ts("ARMA", n = 300, phi = 0.5, theta = 0.4)
ts_logistic <- new_deterministic_ts("logistic", N = 300)
ts_deterministic <- new_deterministic_ts("henon", N = 300)
```

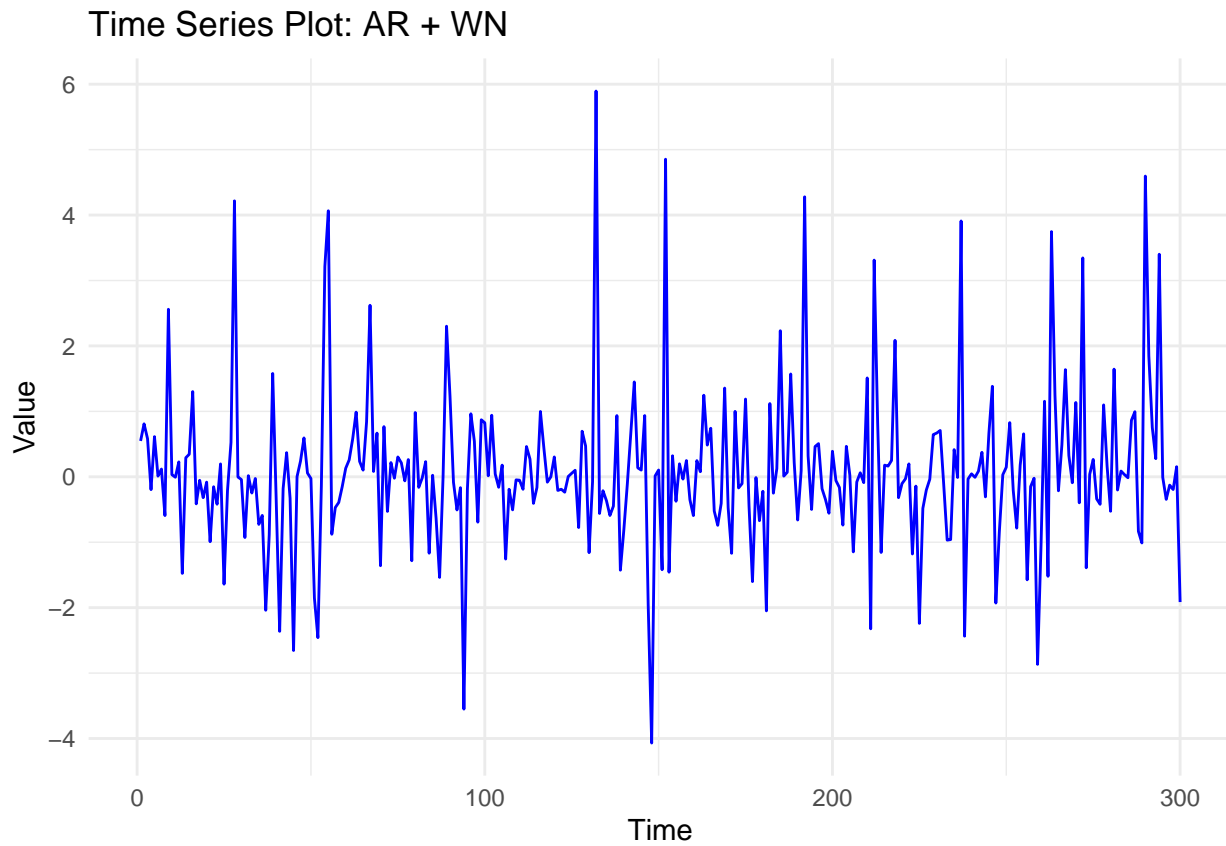
Check that the time series data generated for each model matches the specified parameters.

Mixing Time Series

Validate the correct combination of two time series using addition and multiplication.

```
combined_add <- combine.TimeSeries(ts_ar, ts_logistic, method = "add", alpha = 0.5)
combined_multiply <- combine.TimeSeries(ts_ar, ts_wn, method = "multiply")

plot(combined_multiply)
```



Check that the combined time series is correctly generated with the appropriate naming conventions, and the length matches the original series.

File Upload Validation

Verify correct handling of uploaded files and appropriate error responses for invalid files.

Tests:

- Upload a CSV file with non-numeric values and check that no series is added.
- Upload a CSV with multiple columns and check that no series is added.
- Upload a valid CSV file to confirm successful processing. There is an example data set `example_timeseries.csv` stored in the folder `inst/extdata`.

Expected Results:

- Non-numeric and multi-column files produce error messages, and no series is added.

- Valid files are processed correctly, and the resulting time series matches the uploaded data.

Performance Testing

This package leverages Rcpp for enhanced speed in time series operations. Below is a simple comparison when “mixing” (addition or multiplication) of a stochastic ARMA(1,1) time series with a deterministic logistic time series.

```
library(HCVisR)
library(microbenchmark)
library(ggplot2)
set.seed(123)
ts_arma <- new_stochastic_ts("ARMA", n = 300, phi = 0.5, theta = 0.4)
ts_logistic <- new_deterministic_ts("logistic", N = 300)
alpha <- 0.5
ts_list <- list(ts_arma$series, ts_logistic$series)

# Vectorized R function
R_vectorized_add <- function(ts1, ts2, alpha) alpha * ts1 + (1 - alpha) * ts2
R_vectorized_multiply <- function(ts1, ts2) ts1 * ts2

# C++ functions from HCVisR package
Cpp_add <- function() time_series_operations_cpp(ts_list, "add", alpha)$result
Cpp_multiply <- function() time_series_operations_cpp(ts_list, "multiply", alpha)$result

# Run benchmark
benchmark_results <- microbenchmark(
  R_vectorized_add = R_vectorized_add(ts_arma$series, ts_logistic$series, alpha),
  Cpp_add = Cpp_add(),

  R_vectorized_multiply = R_vectorized_multiply(ts_arma$series, ts_logistic$series),
  Cpp_multiply = Cpp_multiply(),

  times = 10000
)

# Plot the benchmark results
autoplot(benchmark_results)
```

