HCVisR: Time Series Generation and H x C Visualisation

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.

Introduction to the H x C Plane

The $H \times C$ plane, representing Shannon Entropy (H) and Statistical Complexity (C), provides a way to understand the structure and predictability of time series. **Shannon Entropy (H)** measures the randomness in the series, while **Statistical Complexity (C)** assesses the complexity or structural richness. Together, they allow for an insightful visualization of different types of time series.

Installation

You can install the development version from GitHub:

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

Basic Usage

To start the Shiny app, use the following command:

```r
HCVisR::launchApp()
```

The app provides options for generating and uploading time series. Users can visualise the time series and see its position on the H x C plane based on the selected embedding dimension.

Using the Shiny App Interface

The app interface contains several main sections:

- **Time Series Generation**: Generate stochastic or deterministic time series by specifying parameters.

- **File Upload**: Upload your time series in CSV format (one column, numeric values only, and maximum of 1000 data points). There is an example data set `example_timeseries.csv` stored in the folder `inst/extdata`.
- **Embedding Dimension Selection**: Choose an embedding dimension (3 to 6) for H and C calculations.
- **Plot Visualizations**: View the time series plot and its corresponding H x C point based on selected parameters.

Example Workflow

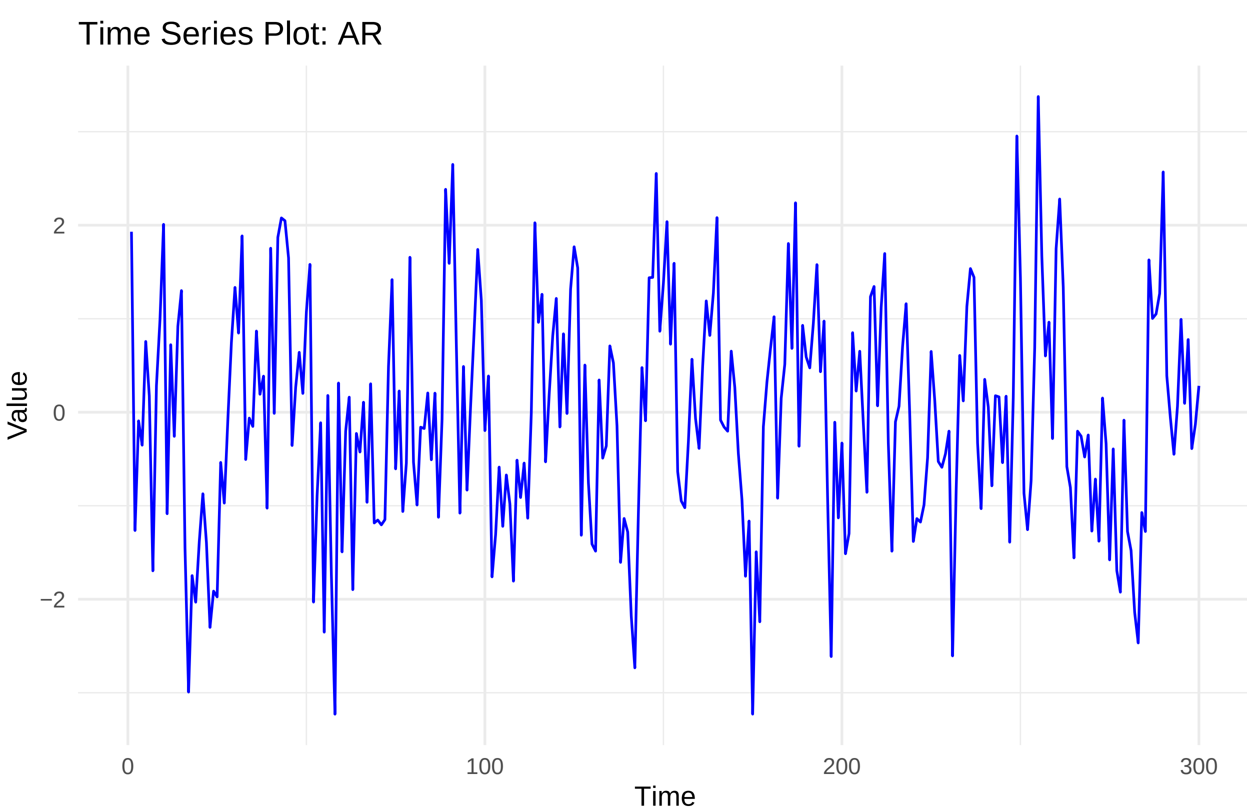
Here's an example of how to generate a stochastic AR(1) time series and visualise it:

```r
# Generate AR(1) Time Series
ts_ar <- new_stochastic_ts("AR", n = 300, phi = 0.5)

# Print the time series
print(ts_ar)
```

```
## Time Series Model:  AR
## First few values of the time series:
##  [1]  1.93022112 -1.26518307 -0.09260985 -0.35230443  0.75662745  0.17366595
##  [7] -1.69640827  0.28538675  0.99533208  2.00901970
## ...
```

```r
# Visualise the time series
plot(ts_ar)
```



```r
# Get a summary of the series
summary(ts_ar)
```

```
## Model:  AR
## Length of series:  300
## Summary statistics:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -3.22951 -0.95228 -0.08828 -0.05261  0.76187  3.37626
```

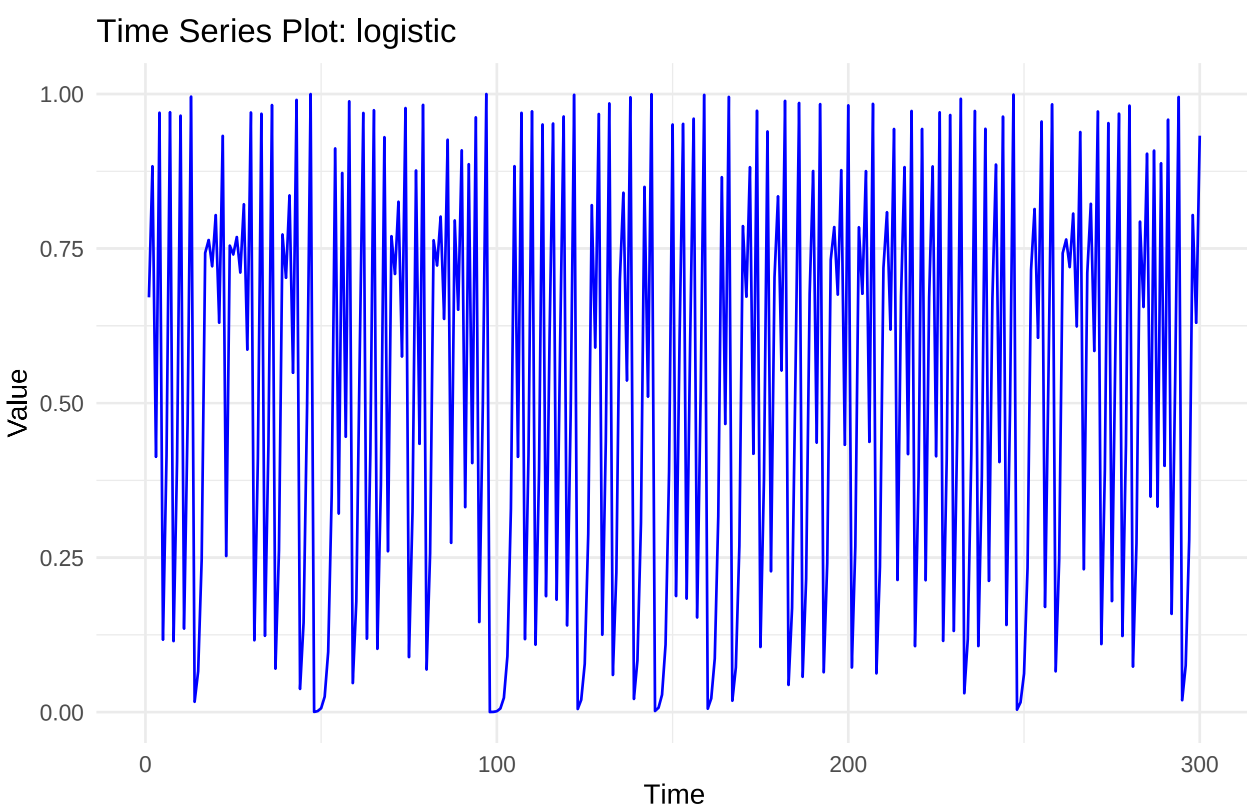
User can also generate a deterministic time series and explore it with the same object-oriented methods:

```r
# Generate logistic time series
ts_logistic <- new_deterministic_ts(model = "logistic", N = 300)

# Print the time series
print(ts_logistic)
```

```
## Time Series Model:  logistic
## First few values of the time series:
##  [1] 0.6709849 0.8830566 0.4130705 0.9697731 0.1172531 0.4140192 0.9704292
##  [8] 0.1147854 0.4064388 0.9649852
## ...
```

```r
# Visualise the time series
plot(ts_logistic)
```



```r
# Get a summary of the series
summary(ts_logistic)
```

```
## Model:  logistic
## Length of series:  300
## Summary statistics:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000911 0.1832670 0.5426592 0.5260780 0.8756810 0.9999772
```

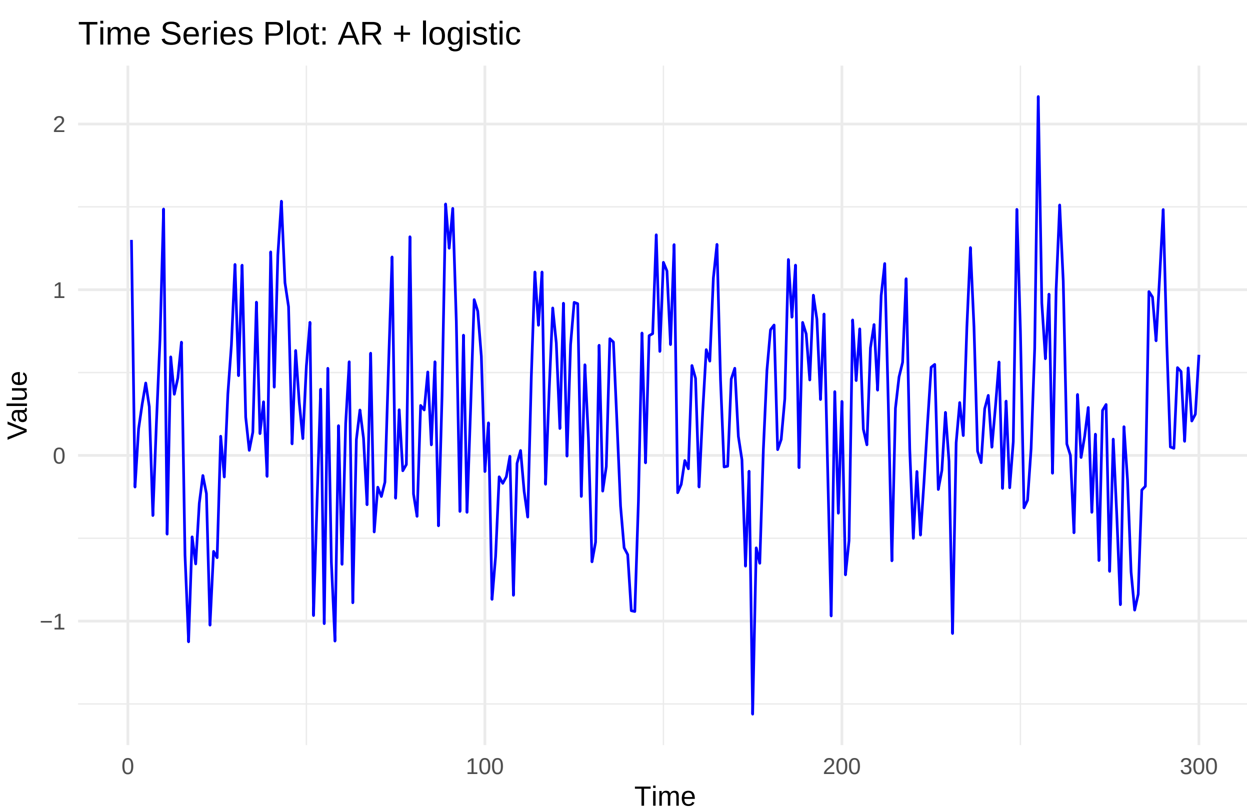
Users can then combine two time series via addition or multiplication. Below is an example of combining the previously generated AR(1) and the logistic time series:

```r
# Combine AR(1) and White Noise time series using addition
combined_add <- combine.TimeSeries(ts_ar, ts_logistic, method = "add", alpha = 0.5)

# Print the combined series
print(combined_add)
```

```
## Time Series Model:  AR + logistic
## First few values of the time series:
##  [1]  1.3006030 -0.1910632  0.1602303  0.3087343  0.4369403  0.2938426
##  [7] -0.3629895  0.2000861  0.7008855  1.4870025
## ...
```

```r
# Visualize the combined series
plot(combined_add)
```



```r
# Get a summary of the combined series
summary(combined_add)
```

```
## Model:  AR + logistic
## Length of series:  300
## Summary statistics:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.5618 -0.1744  0.2398  0.2367  0.6685  2.1659
```

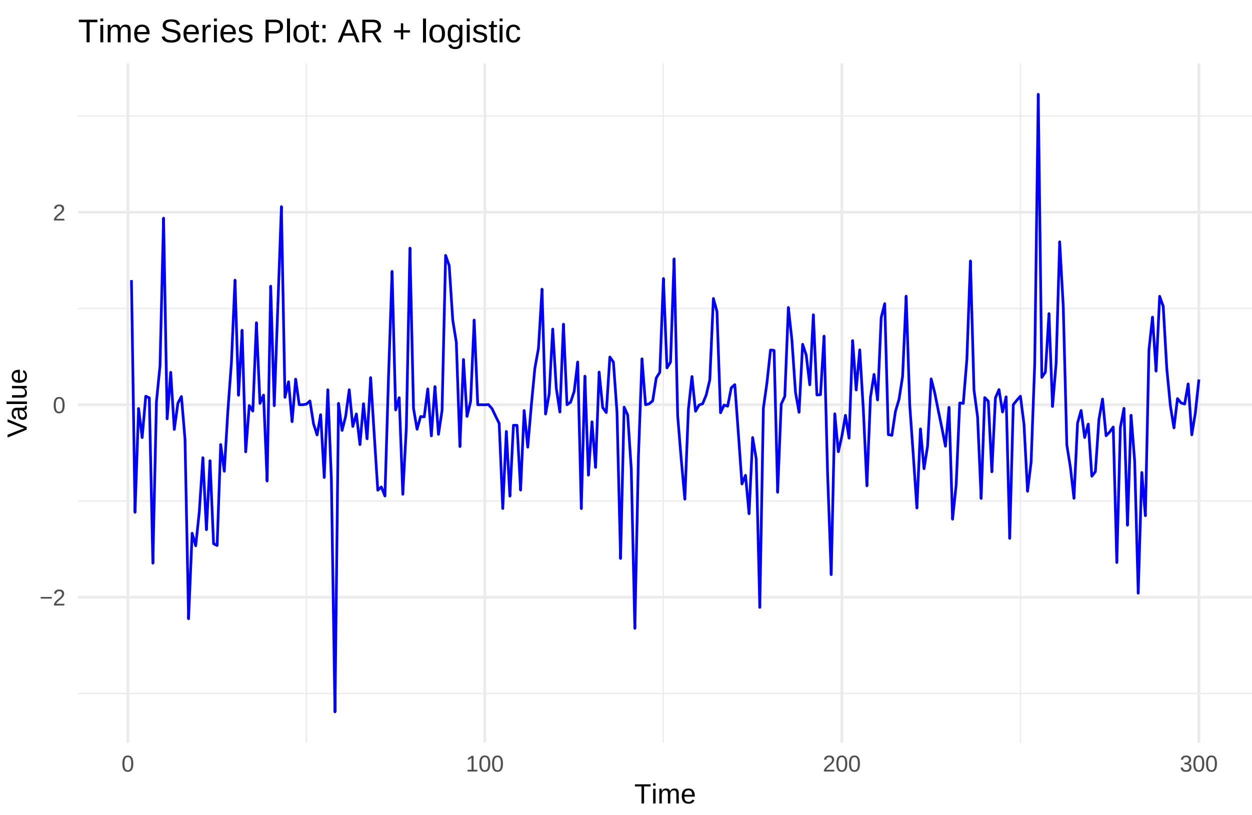
You can also combine them using multiplication:

```r
# Combine AR(1) and White Noise time series using multiplication
combined_multiply <- combine.TimeSeries(ts_ar, ts_logistic, method = "multiply")

# Print the combined series
print(combined_multiply)
```

```
## Time Series Model:  AR + logistic
## First few values of the time series:
##  [1]  1.29514928 -1.11722828 -0.03825440 -0.34165534  0.08871691  0.07190104
##  [7] -1.64624416  0.03275823  0.40454162  1.93867436
## ...
```

```r
# Visualize the combined series
plot(combined_multiply)
```



```r
# Get a summary of the combined series
summary(combined_multiply)
```

```
## Model:  AR + logistic
## Length of series:  300
## Summary statistics:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -3.19116 -0.34017 -0.01663 -0.05620  0.26061  3.22598
```

Exploring Embedding Dimensions

The embedding dimension affects the calculation of H and C. Users can select an embedding dimension (3 to 6) in the app to explore different structural views. Higher dimensions provide a richer view of complexity, though with increased computation.

Viewing the H x C plot

To view the H $\times$ C visualization:

1. Select a time series or upload your data.
2. Choose the desired embedding dimension (3-6).
3. The plot will display the computed entropy-complexity point in the H x C plane.

This tool is valuable for examining how different types of time series—stochastic or deterministic—position themselves within the entropy-complexity landscape.

Additional Resources and References

- For more on entropy and complexity calculations, refer to the StatOrdPattHxC package.
- Read more about the application of the H $\times$ C plane in the context of time series in the Bandt and Pompe (2002) paper.