Package ‘ctsfeatures’

January 29, 2024

Type Package

Title Analyzing Categorical Time Series

Version 1.2.2

Description An implementation of several functions for feature extraction in categorical time series datasets. Specifically, some features related to marginal distributions and serial dependence patterns can be computed. These features can be used to feed clustering and classification algorithms for categorical time series, among others. The package also includes some interesting datasets containing biological sequences. Practitioners from a broad variety of fields could benefit from the general framework provided by ‘ctsfeatures’.

License GPL-2

Encoding UTF-8

LazyData true

LazyDataCompression xz

Depends R (>= 4.0.0)

RoxygenNote 7.2.3

Imports ggplot2, astsa, latex2exp, Rdpack, Bolstad2, tibble

RdMacros Rdpack

NeedsCompilation no

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Author Angel Lopez-Oriona [aut, cre], Jose A. Vilar [aut]

Maintainer Angel Lopez-Oriona <oriona38@hotmail.com>

Repository CRAN

Date/Publication 2024-01-29 13:30:08 UTC
**binarization**

Constructs the binarized time series associated with a given categorical time series.

**Usage**

`binarization(series)`

**Arguments**

- `series` An object of type tsibble (see R package tsibble), whose column named Value contains the values of the corresponding CTS. This column must be of class factor and its levels must be determined by the range of the CTS.
Details
Given a CTS of length $T$ with range $V = \{1, 2, \ldots, r\}$, $X_t = \{X_1, \ldots, X_T\}$, the function constructs the binarized time series, which is defined as $Y_t = \{Y_1, \ldots, Y_T\}$, with $Y_k = (Y_{k,1}, \ldots, Y_{k,r})^T$ such that $Y_{k,i} = 1$ if $X_k = i$ ($k = 1, \ldots, T, i = 1, \ldots, r$). The binarized series is constructed in the form of a matrix whose rows represent time observations and whose columns represent the categories in the original series.

Value
The binarized time series.

Author(s)
Ángel López-Oriona, José A. Vilar

References

Examples
```r
sequence_1 <- GeneticSequences[which(GeneticSequences$Series==1),]
binarized_series <- binarization(sequence_1) # Constructing the binarized
# time series for the first CTS in dataset GeneticSequences
```

---

**calculate_features**

*Computes several features associated with a categorical time series*

**Description**

`calculate_features` computes several features associated with a categorical time series or between a categorical and a real-valued time series.

**Usage**

```r
calculate_features(series, n_series = NULL, lag = 1, type = NULL)
```

**Arguments**

- **series**: An object of type `tsibble` (see R package `tsibble`), whose column named `Value` contains the values of the corresponding CTS. This column must be of class `factor` and its levels must be determined by the range of the CTS.

- **n_series**: A real-valued time series.

- **lag**: The considered lag (default is 1).

- **type**: String indicating the feature one wishes to compute.
Details

Assume we have a CTS of length $T$ with range $\mathcal{V} = \{1, 2, \ldots, r\}$, $\mathbf{X}_t = \{\mathbf{X}_1, \ldots, \mathbf{X}_T\}$, with $\hat{p}_i$ being the natural estimate of the marginal probability of the $i$th category, and $\hat{p}_{ij}(l)$ being the natural estimate of the joint probability for categories $i$ and $j$ at lag $1$, $i, j = 1, \ldots, r$. Assume also that we have a real-valued time series of length $T$, $\mathbf{Z}_t = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_T\}$. The function computes the following quantities depending on the argument type:

- If `type=gini_index`, the function computes the estimated gini index, $\hat{g} = \frac{T}{T-1} \left(1 - \sum_{i=1}^{r} \hat{p}_i^2\right)$.
- If `type=entropy`, the function computes the estimated entropy, $\hat{c} = \frac{1}{\ln(T)} \sum_{i=1}^{r} \hat{p}_i \ln \hat{p}_i$.
- If `type=chebycheff_dispersion`, the function computes the estimated chebycheff dispersion, $\hat{c} = \frac{1}{T-1} \sum_{i=1}^{r} \hat{p}_i$.
- If `type=gk_tau`, the function computes the estimated Goodman and Kruskal’s tau, $\hat{\tau}(l) = \frac{\sum_{i,j=1}^{r} \frac{\hat{p}_{ij}(l)^2 - \hat{p}_i \hat{p}_j}{\hat{p}_i \hat{p}_j}}{1 - \sum_{i=1}^{r} \hat{p}_i}$.
- If `type=gk_lambda`, the function computes the estimated Goodman and Kruskal’s lambda, $\hat{\lambda}(l) = \frac{\sum_{i,j=1}^{r} \max(\hat{p}_{ij}(l) - \max, \hat{p}_j)}{1 - \max_i \hat{p}_i}$.
- If `type=uncertainty_coefficient`, the function computes the estimated uncertainty coefficient, $\hat{\alpha}(l) = -\frac{\sum_{i,j=1}^{r} \hat{p}_{ij}(l) \ln \left(\frac{\hat{p}_{ij}(l)}{p_i p_j}\right)}{\sum_{i=1}^{r} \hat{p}_i \ln p_i}$.
- If `type=entropy`, the function computes the estimated entropy, $\hat{c} = \frac{1}{\ln(T)} \sum_{i=1}^{r} \hat{p}_i \ln \hat{p}_i$.
- If `type=chebycheff_dispersion`, the function computes the estimated chebycheff dispersion, $\hat{c} = \frac{1}{T-1} \sum_{i=1}^{r} \hat{p}_i$.
- If `type=phi2_measure`, the function computes the estimated Phi2 measure, $\hat{\phi}(l) = \frac{\sum_{i,j=1}^{r} \hat{p}_{ij}(l)}{\sum_{i=1}^{r} \hat{p}_i}$.
- If `type=spectral_envelope`, the function computes the estimated spectral envelope.
- If `type=total_correlation`, the function computes the total correlation given by $\hat{\psi}_1(l) = \frac{1}{r} \sum_{i=1}^{r} \hat{\psi}_i(l)^2$, where $\hat{\psi}_i(l) = \frac{1}{T-l} \sum_{t=1}^{T-l} \hat{c}_{i,t}$, and $\hat{c}_{i,t}$ is the estimated correlation $\hat{c}_{i,t}$.
- If `type=total_mixed_correlation_1`, the function computes the total mixed l-correlation given by $\hat{\Psi}_1(l) = \frac{1}{r} \sum_{i=1}^{r} \hat{\psi}_i(l)^2$, where $\hat{\psi}_i(l) = \hat{c}_{i,t}$.
- If `type=total_mixed_correlation_2`, the function computes the estimated total mixed q-correlation given by $\hat{\Psi}_2(l) = \frac{1}{r} \sum_{i=1}^{r} \int_0^1 \hat{\psi}_i^2(l) \hat{\rho}_l \, dl$, where $\hat{\rho}_l$ is the estimated correlation $\hat{c}_{i,t,l}$.
where $\hat{\psi}_\rho^t(l) = \hat{\text{Corr}}(Y_{t,i}, I(Z_{t-l} \leq q_{Zt}(\rho)))$, being $\mathbf{Y}_t = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_T\}$, with $\mathbf{Y}_k = (\mathbf{Y}_{k,1}, \ldots, \mathbf{Y}_{k,r})^\top$, the binarized time series of $\mathbf{X}_t$, $\rho \in (0,1)$ a probability level, $I(\cdot)$ the indicator function and $q_{Z_t}$ the quantile function of the corresponding real-valued process.

Value
The corresponding feature.

Author(s)
Ángel López-Oriona, José A. Vilar

References

Examples
```
sequence_1 <- GeneticSequences[which(GeneticSequences$Series==1),]
uc <- calculate_features(series = sequence_1, type = "uncertainty_coefficient")
# Computing the uncertainty coefficient
# for the first series in dataset GeneticSequences
se <- calculate_features(series = sequence_1, type = "spectral_envelope")
# Computing the spectral envelope
# for the first series in dataset GeneticSequences
```

**calculate_motifs**

*Computes the relative frequency of motifs in a categorical time series*

**Description**
calculate_motifs computes the motifs of a categorical time series

**Usage**
calculate_motifs(series, motif_length)

**Arguments**

- **series** An object of type tsibble (see R package tsibble), whose column named Value contains the values of the corresponding CTS. This column must be of class factor and its levels must be determined by the range of the CTS.
- **motif_length** The length of the motif.

**Details**
Given a CTS of length $T$ with range $V = \{1,2,\ldots,r\}$, $\mathbf{X}_t = \{\mathbf{X}_1, \ldots, \mathbf{X}_T\}$, and a motif length $L$, the function returns an array of $r^L$ elements, with the element in the position $(i_1,i_2,\ldots,i_L)$ being the relative frequency of the motif “$i_1i_2\cdots i_L$” in the corresponding time series.
**Value**

Returns an array with the relative frequency of motifs in a categorical time series.

**Author(s)**

Ángel López-Oriona, José A. Vilar

**References**


**Examples**

```r
sequence_1 <- GeneticSequences[which(GeneticSequences$Series==1),]
calculate_motifs(sequence_1, motif_length = 3)
# Computing the relative frequencies of motifs of length 3 for the first
# series in dataset GeneticSequences
```

---

**calculate_subfeatures**  
*Computes several subfeatures associated with a categorical time series*

**Description**

calculate_features computes several subfeatures associated with a categorical time series or between a categorical and a real-valued time series.

**Usage**

calculate_subfeatures(series, n_series, lag = 1, type = NULL)

**Arguments**

- `series`  
  
  An object of type tsibble (see R package tsibble), whose column named Value contains the values of the corresponding CTS. This column must be of class factor and its levels must be determined by the range of the CTS.

- `n_series`  
  
  A real-valued time series.

- `lag`  
  
  The considered lag (default is 1).

- `type`  
  
  String indicating the subfeature one wishes to compute.
Details

Assume we have a CTS of length \( T \) with range \( \mathcal{V} = \{1, 2, \ldots, r\} \), \( \mathbf{X}_i = \{\mathbf{X}_{i1}, \ldots, \mathbf{X}_{iT}\} \), with \( \hat{p}_i \) being the natural estimate of the marginal probability of the \( i \)th category, and \( \hat{p}_{ij}(l) \) being the natural estimate of the joint probability for categories \( i \) and \( j \) at lag \( l \), \( i, j = 1, \ldots, r \). Assume also that we have a real-valued time series of length \( T \), \( \mathbf{Z}_t = \{\mathbf{Z}_t, \ldots, \mathbf{Z}_T\} \). The function computes the following subfeatures depending on the argument type:

- If type=entropy, the function computes the subfeatures associated with the estimated entropy, \( \hat{p}_i \ln(\hat{p}_i) \), \( i = 1, 2, \ldots, r \).
- If type=gk_tau, the function computes the subfeatures associated with the estimated Goodman and Kruskal's tau, \( \frac{\hat{p}_{ij}(l)^2}{\hat{p}_i \hat{p}_j} \), \( i, j = 1, 2, \ldots, r \).
- If type=gk_lambda, the function computes the subfeatures associated with the estimated Goodman and Kruskal's lambda, \( \max_i \hat{p}_{ij}(l) \), \( i = 1, 2, \ldots, r \).
- If type=uncertainty_coefficient, the function computes the subfeatures associated with the estimated uncertainty coefficient, \( \hat{p}_{ij}(l) \ln \left( \frac{\hat{p}_{ij}(l)}{\hat{p}_i \hat{p}_j} \right) \), \( i, j = 1, 2, \ldots, r \).
- If type=pearson_measure, the function computes the subfeatures associated with the estimated Pearson measure, \( \frac{(\hat{p}_{ij}(l) - \hat{p}_i \hat{p}_j)^2}{\hat{p}_i \hat{p}_j} \), \( i, j = 1, 2, \ldots, r \).
- If type=phi2_measure, the function computes the subfeatures associated with the estimated Phi2 measure, \( \frac{(\hat{p}_{ij}(l) - \hat{p}_i \hat{p}_j)^2}{\hat{p}_i \hat{p}_j} \), \( i, j = 1, 2, \ldots, r \).
- If type=sakoda_measure, the function computes the subfeatures associated with the estimated Sakoda measure, \( \frac{(\hat{p}_{ij}(l) - \hat{p}_i \hat{p}_j)^2}{\hat{p}_i \hat{p}_j} \), \( i, j = 1, 2, \ldots, r \).
- If type=cramers_vi, the function computes the subfeatures associated with the estimated Cramer's vi, \( \frac{(\hat{p}_{ij}(l) - \hat{p}_i \hat{p}_j)^2}{\hat{p}_i \hat{p}_j} \), \( i, j = 1, 2, \ldots, r \).
- If type=cohens_kappa, the function computes the subfeatures associated with the estimated Cohen's kappa, \( \hat{p}_i(l) - \hat{p}^o_i \), \( i = 1, 2, \ldots, r \).
- If type=total_correlation, the function computes the subfeatures associated with the total correlation, \( \hat{\psi}_{ij}(l), i, j = 1, 2, \ldots, r \) (see type='total_mixed_cor' in the function calculate_features).
- If type=total_mixed_correlation_1, the function computes the subfeatures associated with the total mixed l-correlation, \( \hat{\psi}_i(l), i = 1, 2, \ldots, r \) (see type='total_mixed_correlation_1' in the function calculate_features).
- If type=total_mixed_correlation_2, the function computes the subfeatures associated with the total mixed q-correlation, \( \int_0^1 \hat{\psi}_i^q(l)^2 d\rho, i = 1, 2, \ldots, r \) (see type='total_mixed_correlation_2' in the function calculate_features).

Value

The corresponding subfeature

Author(s)

Ángel López-Oriona, José A. Vilar
conditional_probabilities

References


Examples

```r
sequence_1 <- GeneticSequences[which(GeneticSequences$Series==1),]
suc <- calculate_subfeatures(series = sequence_1, type = 'uncertainty_coefficient')
# Computing the subfeatures associated with the uncertainty coefficient
# for the first series in dataset GeneticSequences
scv <- calculate_subfeatures(series = sequence_1, type = 'cramers_vi')
# Computing the subfeatures associated with the cramers vi
# for the first series in dataset GeneticSequences
```

conditional_probabilities

*Computes the conditional probabilities of a categorical time series*

Description

conditional_probabilities returns a matrix with the conditional probabilities of a categorical time series

Usage

```r
conditional_probabilities(series, lag = 1)
```

Arguments

- **series**: An object of type tsibble (see R package tsibble), whose column named Value contains the values of the corresponding CTS. This column must be of class factor and its levels must be determined by the range of the CTS.
- **lag**: The considered lag (default is 1).

Details

Given a CTS of length \( T \) with range \( \mathcal{V} = \{1, 2, \ldots, r\} \), \( \mathbf{X}_t = \{\mathbf{X}_1, \ldots, \mathbf{X}_T\} \), the function computes the matrix \( \hat{P}_c(l) = (\hat{p}_{ij}(l))_{1 \leq i, j \leq r} \), with \( \hat{p}_{ij}(l) = \frac{T N_{ij}(l)}{\sum_{i} N_i} \), where \( N_i \) is the number of elements equal to \( i \) in the realization \( \mathbf{X}_t \) and \( N_{ij}(l) \) is the number of pairs \((\mathbf{X}_t, \mathbf{X}_{t-l}) = (i, j)\) in the realization \( \mathbf{X}_t \).

Value

A matrix with the conditional probabilities.

Author(s)

Ángel López-Oriona, José A. Vilar
References


Examples

```r
sequence_1 <- GeneticSequences[which(GeneticSequences$Series==1),]
matrix_cp <- conditional_probabilities(series = sequence_1) # Computing the matrix of
# joint probabilities for the first series in dataset GeneticSequences
```

Description

Categorical time series (CTS) of DNA sequences from different viruses

Usage

```r
data(GeneticSequences)
```

Format

A `tsibble` with four columns, which are:

- **Value**: The categorical values of the time series in the dataset.
- **Series**: Integer values indicating the considered time series (there are 32 time series in the dataset).
- **Time**: Integer values indicating the temporal indexes of the observations.
- **Class**: Integer values indicating the class of each time series.

Details

The column **Value** is the concatenation of 32 time series taking four categorical values (DNA bases). The column **Class** is formed by integers from 1 to 4, indicating that there are 4 different classes in the database. Each class is associated with a different family of viruses. For more information, see López-Oriona et al. (2023).

References

joint_probabilities  Computes the joint probabilities of a categorical time series

Description

joint_probabilities returns a matrix with the joint probabilities of a categorical time series

Usage

joint_probabilities(series, lag = 1)

Arguments

series  An object of type tsibble (see R package tsibble), whose column named Value contains the values of the corresponding CTS. This column must be of class factor and its levels must be determined by the range of the CTS.

lag  The considered lag (default is 1).

Details

Given a CTS of length $T$ with range $V = \{1, 2, \ldots, r\}$, $\mathbf{X}_t = \{\mathbf{X}_1, \ldots, \mathbf{X}_T\}$, the function computes the matrix $\hat{P}(l) = (\hat{p}_{ij}(l))_{1 \leq i,j \leq r}$, with $\hat{p}_{ij}(l) = \frac{N_{ij}(l)}{T-l}$, where $N_{ij}(l)$ is the number of pairs $(\mathbf{X}_t, \mathbf{X}_{t-l}) = (i, j)$ in the realization $\mathbf{X}_t$.

Value

A matrix with the joint probabilities.

Author(s)

Ángel López-Oriona, José A. Vilar

References


Examples

sequence_1 <- GeneticSequences[which(GeneticSequences$Series==1),]
matrix_jp <- joint_probabilities(series = sequence_1) # Computing the matrix of # joint probabilities for the first series in dataset GeneticSequences
marginal_probabilities

Computes the marginal probabilities of a categorical time series

Description

marginal_probabilities returns a vector with the marginal probabilities of a categorical time series.

Usage

marginal_probabilities(series)

Arguments

series An object of type tsibble (see R package tsibble), whose column named Value contains the values of the corresponding CTS. This column must be of class factor and its levels must be determined by the range of the CTS.

Details

Given a CTS of length \( T \) with range \( V = \{1, 2, \ldots, r\} \), \( X_t = \{X_1, \ldots, X_T\} \), the function computes the vector \( \hat{p} = (\hat{p}_1, \ldots, \hat{p}_r) \), with \( \hat{p}_i = \frac{N_i}{T} \), where \( N_i \) is the number of elements equal to \( i \) in the realization \( X_t \).

Value

A vector with the marginal probabilities.

Author(s)

Ángel López-Oriona, José A. Vilar

References


Examples

```r
sequence_1 <- GeneticSequences[which(GeneticSequences$Series==1),]
vector_mp <- marginal_probabilities(series = sequence_1) # Computing the vector of marginal probabilities for the first series in dataset GeneticSequences```
plot_ccc  Constructs a control chart for the cycle lengths of a categorical series

Description
plot_ccc constructs a control chart for the cycle lengths of a categorical series

Usage
plot_ccc(
  series,
  mu_t,
  lcl_t,
  ucl_t,
  plot = TRUE,
  title = "Control chart (cycles)",
  ...
)

Arguments
  series     An object of type tsibble (see R package tsibble), whose column named
              Value contains the values of the corresponding CTS. This column must be of
              class factor and its levels must be determined by the range of the CTS.
  mu_t       The mean of the process measuring the cycle lengths.
  lcl_t      The lower control limit.
  ucl_t      The upper control limit.
  plot       Logical. If plot = TRUE (default), returns the control chart. Otherwise, returns
              the standardized statistic.
  title      The title of the graph.
  ...        Additional parameters for the function.

Details
Constructs a control chart of a CTS based on cycle lengths. The chart is based on the standardized
statistic \( T_t = T_t^{(L)} + T_t^{(U)} \), with
\[ T_t^{(L)} = \min \left( 0, \frac{Z_t - \mu_t}{\mid LCL_t - \mu_t \mid} \right) \]
and
\[ T_t^{(U)} = \max \left( 0, \frac{Z_t - \mu_t}{\mid UCL_t - \mu_t \mid} \right), \]
where \( Z_t \) expresses the length of a cycle ending with a specific category, \( \mu_t \) denotes the mean of
\( Z_t \) and \( LCL_t \) and \( UCL_t \) are lower and upper individual control limits, respectively. Note that an
out-of-control alarm is signalled if \( T_t < -1 \) or \( T_t > 1 \).

Value
If plot = TRUE (default), represents the control chart for the cycle lengths. Otherwise, the function
returns a matrix with the values of the standardized statistic for each time \( t \)
Author(s)
Ángel López-Oriona, José A. Vilar

References

Examples
sequence_1 <- SyntheticData1[which(SyntheticData1$Series==1),]
cycle_cc <- plot_ccc(series = sequence_1, mu_t = c(1, 1.5, 1),
lcl_t = rep(10, 600), ucl_t = rep(10, 600)) # Representing
# a control chart for the cycle lengths
cycle_cc <- plot_ccc(series = sequence_1, mu_t = c(1, 1.5, 1),
lcl_t = rep(10, 600), ucl_t = rep(10, 600), plot = FALSE) # Computing the
# corresponding standardized statistic

plot_cohen

Constructs a serial dependence plot based on Cohen’s kappa

Description
plot_cohen constructs a serial dependence plot of a categorical time series based on Cohen’s kappa

Usage
plot_cohen(
  series,
  max_lag = 10,
  alpha = 0.05,
  plot = TRUE,
  title = "Serial dependence plot",
  bar_width = 0.12,
  ...
)

Arguments
series An object of type tsibble (see R package tsibble), whose column named Value contains the values of the corresponding CTS. This column must be of class factor and its levels must be determined by the range of the CTS.
max_lag The maximum lag represented in the plot (default is 10).
alpha The significance level for the corresponding hypothesis test (default is 0.05).
plot Logical. If plot = TRUE (default), returns the serial dependence plot. Otherwise, returns a list with the values of Cohens’s kappa, the critical value and the corresponding p-values.
plot_cramer

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>title</td>
<td>The title of the graph.</td>
</tr>
<tr>
<td>bar_width</td>
<td>The width of the corresponding bars.</td>
</tr>
<tr>
<td>...</td>
<td>Additional parameters for the function.</td>
</tr>
</tbody>
</table>

Details

Constructs a serial dependence plot based on Cohens’s kappa, \( \hat{\kappa}(l) \), for several lags. A dashed lined is incorporated indicating the critical value of the test based on the following asymptotic approximation (under the i.i.d. assumption):

\[
\sqrt{\frac{T}{V(\hat{p})}} \left( \hat{\kappa}(l) + \frac{1}{T} \right) \sim N(0, 1),
\]

where \( T \) is the series length, \( \hat{p} = (\hat{p}_1, \ldots, \hat{p}_r) \) is the vector of estimated marginal probabilities for the \( r \) categories of the series and \( V(\hat{p}) = 1 - \frac{1+2\sum_{i=1}^{r} \hat{p}_i^3 - 3\sum_{i=1}^{r} \hat{p}_i^2}{(1-\sum_{i=1}^{r} \hat{p}_i)^2} \).

Value

If plot = TRUE (default), returns the serial dependence plot based on Cohens’s kappa. Otherwise, the function returns a list with the values of Cohens’s kappa, the critical value and the corresponding p-values.

Author(s)

Ángel López-Oriona, José A. Vilar

References


Examples

```r
sequence_1 <- GeneticSequences[which(GeneticSequences$Series==1),]
plot_ck <- plot_cohen(series = sequence_1, max_lag = 3) # Representing
# the serial dependence plot
list_ck <- plot_cohen(series = sequence_1, max_lag = 3, plot = FALSE) # Obtaining
# the values of Cohens's kappa, the critical value and the p-values
```

plot_cramer

Constructs a serial dependence plot based on Cramer’s vi

Description

plot_cramer constructs a serial dependence plot of a categorical time series based on Cramer’s vi
plot_cramer

Usage

plot_cramer(
  series,
  max_lag = 10,
  alpha = 0.05,
  plot = TRUE,
  title = "Serial dependence plot",
  bar_width = 0.12,
  ...
)

Arguments

series An object of type tsibble (see R package tsibble), whose column named Value contains the values of the corresponding CTS. This column must be of class factor and its levels must be determined by the range of the CTS.

max_lag The maximum lag represented in the plot (default is 10).

alpha The significance level for the corresponding hypothesis test (default is 0.05).

plot Logical. If plot = TRUE (default), returns the serial dependence plot. Otherwise, returns a list with the values of Cramer’s vi, the critical value and the corresponding p-values.

title The title of the graph.

bar_width The width of the corresponding bars.

... Additional parameters for the function.

Details

Constructs a serial dependence plot based on Cramer’s vi, \( \hat{v}(l) \), for several lags. A dashed line is incorporated indicating the critical value of the test based on the following asymptotic approximation (under the i.i.d. assumption):

\[
T(r - 1)\hat{v}(l)^2 \sim \chi^2_{(r-1)^2},
\]

where \( T \) is the series length and \( r \) is the number of categories in the time series.

Value

If plot = TRUE (default), returns the serial dependence plot based on Cramer’s vi. Otherwise, the function returns a list with the values of Cramer’s vi, the critical value and the corresponding p-values.

Author(s)

Ángel López-Oriona, José A. Vilar

References

Examples

```r
sequence_1 <- SyntheticData1[which(SyntheticData1$Series==1),]
plot_cv <- plot_cramer(series = sequence_1, max_lag = 3) # Representing
# the serial dependence plot
list_cv <- plot_cramer(series = sequence_1, max_lag = 3, plot = FALSE) # Obtaining
# the values of Cramer's vi, the critical value and the p-values
```

---

**plot_cts**

Constructs a categorical time series plot

Description

plot_cts constructs a categorical time series plot

Usage

```r
plot_cts(series, title = "Time series plot")
```

Arguments

- `series`: An object of type tsibble (see R package tsibble), whose column named Value contains the values of the corresponding CTS. This column must be of class factor and its levels must be determined by the range of the CTS.
- `title`: The title of the graph.

Details

Constructs a categorial time series plot for a given CTS.

Value

The categorical time series plot.

Author(s)

Ángel López-Oriona, José A. Vilar

References


Examples

```r
sequence_1 <- GeneticSequences[which(GeneticSequences$Series==1),]
time_series_plot <- plot_cts(series = sequence_1) # Constructs a categorical
# time series plot for the first 50 observations of the first time series in
# dataset GeneticSequences
```
plot_ifsct

Description

plot_ifsct constructs the IFS circle transformation of a categorical time series.

Usage

plot_ifsct(series, alpha, beta, title = "IFS circle transformation", ...)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>series</td>
<td>An object of type tsibble (see R package tsibble), whose column named Value contains the values of the corresponding CTS. This column must be of class factor and its levels must be determined by the range of the CTS.</td>
</tr>
<tr>
<td>alpha</td>
<td>Parameter alpha in the circle transformation.</td>
</tr>
<tr>
<td>beta</td>
<td>Parameter beta in the circle transformation.</td>
</tr>
<tr>
<td>title</td>
<td>The title of the graph.</td>
</tr>
<tr>
<td>...</td>
<td>Additional parameters for the function.</td>
</tr>
</tbody>
</table>

Details

Constructs the IFS circle transformation for a given CTS, which is useful to identify cycles of arbitrary length.

Value

The IFS circle transformation.

Author(s)

Ángel López-Oriona, José A. Vilar

References


Examples

```r
sequence_1 <- GeneticSequences[which(GeneticSequences$Series==1),]
cf <- plot_ifsct(sequence_1, alpha = 0.1, beta = 0.1) # Constructing the IFS circle transformatation # for the first CTS in dataset GeneticSequences
```
plot_mcc

Constructs a control chart for the marginal distribution of a categorical series

Description

plot_mcc constructs a control chart for the marginal distribution of a categorical series

Usage

plot_mcc(
  series,
  c,
  sigma,
  lambda = 0.99,
  k = 3.3,
  min_max = FALSE,
  plot = TRUE,
  title = "Control chart (marginal)",
  ...
)

Arguments

series  An object of type tsibble (see R package tsibble), whose column named Value contains the values of the corresponding CTS. This column must be of class factor and its levels must be determined by the range of the CTS.

c        The hypothetical marginal distribution.

sigma    A matrix containing the variances for each category (columns) and each time t (rows).

lambda   The constant lambda to construct the EWMA estimator.

k         The constant k to construct the k sigma limits.

min_max   Logical. If min_max = FALSE (default), the standard control chart for the marginal distribution is plotted. Otherwise, the reduced control chart is plotted, i.e., only the minimum and maximum values of the standardized statistics (with respect to the set of categories) are considered.

plot      Logical. If plot = TRUE (default), returns the control chart. Otherwise, returns the standardized statistics or their maximum and minimum value for each time t.

title     The title of the graph.

...       Additional parameters for the function.
Details
Constructs a control chart of a CTS with range $V = \{1, \ldots, r\}$ based on the marginal distribution. The chart relies on the standardized statistic $T_{t,i} = \frac{\hat{\pi}_{\lambda,t,i} - p_i}{k \cdot \sigma_{t,i}}$, where the $\hat{\pi}_{\lambda,t,i}$, $i = 1, \ldots, r$, are the components of the EWMA estimator of the marginal distribution, $p_i$ is the marginal probability of category $i$, $\sigma_{t,i}$ is the variance of $\hat{\pi}_{\lambda,t,i}$ and $k$ is a constant set by the user. If $\text{min\_max} = \text{FALSE}$, then only the statistics $T_{t,\text{min}} = \min_{i \in V} T_{t,i}$ and $T_{t,\text{max}} = \max_{i \in V} T_{t,i}$ are plotted. An out-of-control alarm is signalled if the statistics are below -1 or above 1.

Value
If $\text{plot} = \text{TRUE}$ (default), represents the control chart for the marginal distribution. Otherwise, the function returns a matrix with the values of the standardized statistics for each time $t$.

Author(s)
Ángel López-Oriona, José A. Vilar

References

Examples
```r
sequence_1 <- SyntheticData1[which(SyntheticData1$Series==1),]
cycle_cc <- plot_ccc(series = sequence_1, mu_t = c(1, 1.5, 1), lcl_t = rep(10, 600), ucl_t = rep(10, 600))
cycle_md <- plot_mcc(series = sequence_1, c = c(0.3, 0.3, 0.4), sigma = matrix(rep(c(1, 1, 1), 600), nrow = 600)) # Representing a control chart for the marginal distribution
cycle_md <- plot_mcc(series = sequence_1, c = c(0.3, 0.3, 0.4), sigma = matrix(rep(c(1, 1, 1), 600), nrow = 600), plot = FALSE) # Computing the # corresponding standardized statistic
```

---

**plot_ph**

Constructs the pattern histogram associated with a given category of a categorical time series.

Description

plot_ph constructs the pattern histogram associated with a given category of a categorical time series.
plot_ph

Usage

plot_ph(
  series,
  category,
  plot = TRUE,
  title = paste0("Pattern histogram (", category, ")"),
  ...
)

Arguments

series    An object of type tsibble (see R package tsibble), whose column named
           Value contains the values of the corresponding CTS. This column must be of
           class factor and its levels must be determined by the range of the CTS.
category   The selected category.
plot       Logical. If plot = TRUE (default), returns the pattern histogram. Otherwise,
           returns the frequencies of cycle lengths associated with the corresponding cate-
           gory.
title      The title of the graph.
...        Additional parameters for the function.

Details

Constructs the pattern histogram for a specific category of a CTS. This graph represents the fre-
quencies of the cycles for the corresponding category according to their length.

Value

The pattern histogram.

Author(s)

Ángel López-Oriona, José A. Vilar

References


Examples

sequence_1 <- GeneticSequences[which(GeneticSequences$Series==1),]
ph <- plot_ph(sequence_1,
  category = 'a') # Constructing the pattern histogram
# for the first CTS in dataset GeneticSequences concerning the category 'a'
cycle_lengths <- plot_ph(sequence_1,
  category = 'a', plot = FALSE) # Obtaining the frequencies of cycle lengths
Description

plot_reg constructs the rate evolution graph proposed by Ribler (1997).

Usage

plot_reg(
  series,
  title = "Rate evolution graph",
  linear_fit = FALSE,
  cat_res = NULL,
  ...
)

Arguments

series An object of type tsibble (see R package tsibble), whose column named Value contains the values of the corresponding CTS. This column must be of class factor and its levels must be determined by the range of the CTS.

title The title of the graph.

linear_fit Logical. If TRUE, the corresponding least squares lines are incorporated to the graph.

cat_res If this parameter is set to any of the categories of the series, then the function returns a graph of residuals for the linear model associated with the corresponding category.

... Additional parameters for the function.

Details

Given a CTS of length $T$ with range $V = \{1, 2, \ldots, r\}$, $X_t = \{X_1, \ldots, X_T\}$, and the corresponding binarized time series, $Y_t = \{Y_1, \ldots, Y_T\}$, the function constructs the rate evolution graph. Specifically, consider the series of cumulated sums given by $C_t = \{C_1, \ldots, C_T\}$, with $C_k = \sum_{s=1}^{k} Y_s$, $k = 1, \ldots, T$. The rate evolution graph displays a standard time series plot for each one of the components of $C_t$ simultaneously in one graph.

Value

The rate evolution graph.

Author(s)

Ángel López-Oriona, José A. Vilar
plot_se

Represent the spectral envelope of a categorical time series

Description

plot_se represents the spectral envelope of a categorical time series

Usage

plot_se(series)

Arguments

series An object of type tsibble (see R package tsibble), whose column named Value contains the values of the corresponding CTS. This column must be of class factor and its levels must be determined by the range of the CTS.

Details

The function represents the spectral envelope of a categorical time series

Value

Returns returns a plot of the spectral envelope.

Author(s)

Ángel López-Oriona, José A. Vilar

References


Examples

sequence_1 <- GeneticSequences[which(GeneticSequences$Series==1),]
plot_se(sequence_1)  # Representing the spectral envelope for the first series in dataset GeneticSequences

References

ProteinSequences

Description
Categorical time series (CTS) of protein sequences from different species

Usage
data(ProteinSequences)

Format
A tsibble with four columns, which are:

Value  The categorical values of the time series in the dataset.
Series  Integer values indicating the considered time series (there are 40 time series in the dataset).
Time    Integer values indicating the temporal indexes of the observations.
Class   Integer values indicating the class of each time series.

Details
The column Value is the concatenation of 40 time series taking four categorical values (amino-acids). The column Class is formed by integers from 1 to 4, indicating that there are 4 different classes in the database. Each class is associated with a different family of viruses. For more information, see López-Oriona et al. (2023).

References

SleepStages

Description
Categorical time series (CTS) of sleep stages from different subjects

Usage
data(SleepStages)
**Format**

A `tsibble` with four columns, which are:

- **Value**: The categorical values of the time series in the dataset.
- **Series**: Integer values indicating the considered time series (there are 62 time series in the dataset).
- **Time**: Integer values indicating the temporal indexes of the observations.
- **Class**: Integer values indicating the class of each time series.

**Details**

The column `Value` is the concatenation of 62 time series taking six categorical values (sleep stages). The column `Class` is formed by the integers 1 and 2 indicating that there are 2 different classes in the database. Each class is associated with a sleep disorder (class 1 refers to nocturnal frontal lobe epilepsy, while class refers 2 to REM behavior disorder). For more information, see López-Oriona et al. (2023).

**References**


---

**Description**

Synthetic dataset containing 80 CTS generated from four different generating processes.

**Usage**

```r
data(SyntheticData1)
```
SyntheticData2

References


---

**SyntheticData2**

**Description**

Synthetic dataset containing 80 CTS generated from four different generating processes.

**Usage**

data(SyntheticData2)

**Format**

A `tsibble` with four columns, which are:

- **Value** The categorical values of the time series in the dataset.
- **Series** Integer values indicating the considered time series (there are 80 time series in the dataset).
- **Time** Integer values indicating the temporal indexes of the observations.
- **Class** Integer values indicating the class of each time series.

@details The column `Value` is the concatenation of 80 time series of length 600 taking three categorical values. Series 1-20, 21-40, 41-60 and 61-80 were generated from Hidden Markov Models with different matrices of transition and emission probabilities (see Scenario 2 in López-Oriona et al. (2023)). Therefore, there are 4 different classes in the dataset.

**References**

Description

Synthetic dataset containing 80 CTS generated from four different generating processes.

Usage

data(SyntheticData3)

Format

A tsibble with four columns, which are:

Value  The categorical values of the time series in the dataset.
Series  Integer values indicating the considered time series (there are 80 time series in the dataset).
Time   Integer values indicating the temporal indexes of the observations.
Class  Integer values indicating the class of each time series.

@details The column Value is the concatenation of 80 time series of length 600 taking three categorical values. Series 1-20, 21-40, 41-60 and 61-80 were generated from NDARMA processes with different orders and vectors of coefficients (see Scenario 3 in López-Oriona et al. (2023)). Therefore, there are 4 different classes in the dataset.

References

Index

* datasets
  GeneticSequences, 9
  ProteinSequences, 23
  SleepStages, 23
  SyntheticData1, 24
  SyntheticData2, 25
  SyntheticData3, 26

binarization, 2

calculate_features, 3
calculate_motifs, 5
calculate_subfeatures, 6
conditional_probabilities, 8

GeneticSequences, 9

joint_probabilities, 10

marginal_probabilities, 11

plot_ccc, 12
plot_cohen, 13
plot_cramer, 14
plot_cts, 16
plot_ifsct, 17
plot_mcc, 18
plot_ph, 19
plot_reg, 21
plot_se, 22

ProteinSequences, 23

SleepStages, 23
SyntheticData1, 24
SyntheticData2, 25
SyntheticData3, 26