

Package: trama.series (via r-universe)

October 21, 2026

Title Time Series Collection for trama

Version 0.9.5

Description Nodes for time series in a trama graph: operators (differencing, lags, transforms, windows, moving averages, aggregation), classical and STL decomposition, ARIMA, ETS and Holt-Winters models with forecasts and benchmarks, stationarity and white-noise tests, and series-specific plots such as correlograms, seasonal and lag plots; multivariate series with VAR and VECM models, Granger causality, Engle-Granger and Johansen cointegration tests, impulse responses and residual diagnostics. Builds on trama.data for tables and on trama.view for plots, without duplicating either.

License MIT + file LICENSE

Encoding UTF-8

Depends R (≥ 4.1)

Imports trama ($\geq 0.5.26$), trama.data ($\geq 0.4.5$), trama.view, trama.models ($\geq 0.6.3$), forecast (≥ 8.0), nlme, tsoutliers, urca, vars (≥ 1.5), ggplot2 ($\geq 4.0.0$), rlang, tibble, datasets, stats, utils

Suggests testthat ($\geq 3.0.0$), aTSA, GeneCycle, png, randtests, tseries, trend, modifiedmk, TSA

Config/testthat/edition 3

Config/trama/requires trama.data, trama.view, trama.models

Remotes uaipetro/trama,
trama.data=github::uaipetro/trama/collections/trama.data,
trama.view=github::uaipetro/trama/collections/trama.view,
trama.models=github::uaipetro/trama/collections/trama.models

Config/roxygen2/version 8.0.0

Config/pak/sysreqs cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libjpeg-dev libpng-dev libtiff-dev libuv1-dev libwebp-dev libssl-dev libx11-dev zlib1g-dev

Repository <https://uaipetro.r-universe.dev>

Date/Publication 2026-10-20 21:24:06 UTC

RemoteUrl <https://github.com/uaipedro/trama>

RemoteRef HEAD

RemoteSha a0a356f56db4d8598f58098a83208b81b51ce5ae

RemoteSubdir collections/trama.series

Contents

tr_series_accuracy	3
tr_series_acf	4
tr_series_adf	4
tr_series_aggregate	4
tr_series_arima	5
tr_series_as_decomposition	5
tr_series_baseline	6
tr_series_box_pierce	6
tr_series_combine	7
tr_series_component	7
tr_series_cox_stuart	8
tr_series_decompose	8
tr_series_deseasonalize	9
tr_series_detect_interventions	9
tr_series_detrend	10
tr_series_diff	10
tr_series_errors	11
tr_series_ets	11
tr_series_example	12
tr_series_fisher	12
tr_series_forecast	13
tr_series_from_table	13
tr_series_from_table_mts	14
tr_series_holt_winters	14
tr_series_interpolate	15
tr_series_intervencao	15
tr_series_join	15
tr_series_kpss	16
tr_series_kruskal_wallis	16
tr_series_lag	17
tr_series_lag_plot	17
tr_series_ljung_box	18
tr_series_mann_kendall	18
tr_series_moving_average	19
tr_series_ndiffs	19
tr_series_pacf	19
tr_series_pettitt	20
tr_series_phillips_perron	20
tr_series_pick	21

<code>tr_series_accuracy</code>	3
<code>tr_series_plot</code>	21
<code>tr_series_plot_decomposition</code>	22
<code>tr_series_plot_forecast</code>	22
<code>tr_series_plot_mts</code>	23
<code>tr_series_range_mean</code>	23
<code>tr_series_regression</code>	24
<code>tr_series_residuals</code>	24
<code>tr_series_runs</code>	25
<code>tr_series_seasonal_plot</code>	25
<code>tr_series_stl</code>	26
<code>tr_series_subseries</code>	26
<code>tr_series_transform</code>	27
<code>tr_series_var</code>	27
<code>tr_series_window</code>	28
<code>tr_series_zivot_andrews</code>	28
<code>trama_collection</code>	29
Index	30

<code>tr_series_accuracy</code>	<i>Medidas de erro da previsão, no treino e — se houver — no teste.</i>
---------------------------------	---

Description

Sem a série real, só a linha do treino: erro de um passo à frente dentro da amostra, que é otimista por construção. Com a série real (os períodos que o modelo não viu, recortados por ‘series/window’), sai a linha do teste, que é a que responde se a previsão presta.

Usage

```
tr_series_accuracy(previsao, real = NULL, reais = NULL)
```

Details

Série real que não cobre nenhum período da previsão é ERRO, e não linha vazia: é quase sempre o fio ligado na série de treino por engano, e uma tabela só com o treino pareceria resposta.

tr_series_acf	<i>Correlograma: a autocorrelação em cada defasagem.</i>
---------------	--

Description

Correlograma: a autocorrelação em cada defasagem.

Usage

```
tr_series_acf(
  serie,
  defasagens = 0L,
  aspecto = "16:9",
  tema = "padrão",
  titulo = "",
  rotulo_x = "",
  rotulo_y = "",
  legenda = "direita"
)
```

tr_series_adf	<i>ADF: a série tem raiz unitária?</i>
---------------	--

Description

H0 é a RAIZ UNITÁRIA, e isso inverte a leitura de quem está acostumado a p-valor pequeno ser "achei alguma coisa": aqui rejeitar é concluir ESTACIONÁRIA. O KPSS é o outro lado, com H0 oposta, e é a concordância dos dois que dá chão à conclusão — a ajuda dos dois aponta um para o outro.

Usage

```
tr_series_adf(serie, deterministico = "constante", defasagens = 0L)
```

tr_series_aggregate	<i>Muda a frequência para uma menor: mensal -> trimestral -> anual.</i>
---------------------	---

Description

Não usa 'stats::aggregate()', e a razão foi medida: ele agrupa em blocos a partir da PRIMEIRA observação, e não do calendário. Numa série mensal que começa em março, o "ano" 1 soma de março a fevereiro e é rotulado com o ano de março — anos que não existem, verdes. Aqui o primeiro bloco incompleto é descartado, e o último também: o ano de 1960 com seis meses somaria a metade e pareceria uma queda.

Usage

```
tr_series_aggregate(serie, frequencia = 1L, funcao = "soma")
```

<code>tr_series_arima</code>	<i>ARIMA, automático (Hyndman-Khandakar) ou com a ordem digitada.</i>
------------------------------	---

Description

‘automatico = TRUE’ ignora as seis ordens, e a ajuda diz isso com todas as letras: esconder os campos dependeria de um param condicional que o card não tem. O resumo do card mostra a ordem que a busca escolheu, então o caminho natural é rodar automático, ler a ordem, e só então fixar à mão.

Usage

```
tr_series_arima(
  serie,
  automatico = TRUE,
  p = 1L,
  d = 1L,
  q = 1L,
  P = 0L,
  D = 0L,
  Q = 0L,
  sazonal = TRUE,
  constante = TRUE
)
```

<code>tr_series_as_decomposition</code>	<i>Decomposição de um ajuste de série.</i>
---	--

Description

Converte o ‘models/fit’ de ‘series/regression’, ‘series/detrend’ ou ‘series/deseasonalize’ na decomposição aditiva (tendência, sazonal, resto e, se houver, o efeito da covariável) que ‘tr_series_component()’ e ‘tr_series_plot_decomposition()’ leem. É o adaptador que o editor põe no fio entre os dois tipos; exportado para o script gerado chamá-lo pelo nome.

Usage

```
tr_series_as_decomposition(x)
```

Arguments

x Ajuste produzido por um dos nós de regressão de série.

Value

Objeto ‘tr_series_decomp’.

tr_series_baseline	<i>As previsões de referência: o que qualquer modelo tem de bater.</i>
--------------------	--

Description

Um ARIMA que erra mais que "o mesmo mês do ano passado" não merece o card, e sem o ingênuo sazonal ao lado ninguém fica sabendo. Por isso a referência é um nó de primeira ordem, e não uma coluna escondida no ‘accuracy’.

Usage

```
tr_series_baseline(serie, metodo = "ingênuo sazonal", horizonte = 12L)
```

tr_series_box_pierce	<i>Box-Pierce: a série é ruído branco?</i>
----------------------	--

Description

A mesma hipótese nula do ‘series/ljung_box’, na fórmula original de 1970 — sem a correção de amostra pequena, que é justamente o que o Ljung-Box acrescentou. Fica na coleção para comparar com trabalho antigo que o reportou; para decidir, o Ljung-Box.

Usage

```
tr_series_box_pierce(serie, defasagens = 0L, graus = 0L)
```

tr_series_combine	<i>Opera duas séries ponto a ponto: $a - b$, $a + b$, a / b, $a \times b$.</i>
-------------------	--

Description

O alinhamento é pelo TEMPO, e não pela posição: ‘a’ de 1949 a 1960 menos ‘b’ de 1955 a 1965 é a diferença de 1955 a 1960 — a interseção. Operar os vetores crus somaria janeiro de 1949 com janeiro de 1955, verde.

Usage

```
tr_series_combine(a, b, operacao = "a - b")
```

Details

Frequências diferentes são erro, e não conversão implícita: somar uma mensal com uma trimestral pede escolher COMO agregar (soma? média?), e é o ‘series/aggregate’ que faz essa pergunta.

Divisão por zero sai NA, e não ‘Inf’/‘NaN’: um ‘Inf’ no meio da série quebra a escala do gráfico e todo nó seguinte, e o aviso do R não chega ao card. O NA chega — o resumo da série conta os faltantes —, e o ‘series/interpolate’ sabe o que fazer com ele.

tr_series_component	<i>Um componente da decomposição, de volta como série.</i>
---------------------	--

Description

‘dessazonalizada’ é a série sem o sazonal — o que se publica quando se diz “o desemprego subiu em março, descontado o efeito do mês”. Na multiplicativa é a divisão, e não a subtração: subtrair um fator de 1,08 de um valor na casa das centenas não tira sazonalidade nenhuma.

Usage

```
tr_series_component(decomposicao, componente = "dessazonalizada")
```

Details

‘sem_tendencia’ é o espelho: a série sem a tendência, com a sazonalidade dentro (sazonal + resto, ou sazonal $\times$ resto). É o “estimo a tendência e subtraio” feito às claras, e a mesma regra da divisão vale — na multiplicativa a tendência é o NÍVEL, e a série sem ela é a razão.

`tr_series_cox_stuart` *Cox-Stuart: a série tem tendência?*

Description

Um teste de SINAL: pareia observações distantes no tempo, conta quantas vezes a segunda supera a primeira e pergunta se essa contagem é compatível com cara-ou-coroa. Responde à mesma pergunta do ‘series/mann_kendall’ por um caminho mais barato — uma contagem de pares em vez de todos os pares da série —, e a dissertação põe os dois lado a lado.

Usage

```
tr_series_cox_stuart(
  serie,
  pareamento = "terços",
  correcao = "nenhuma",
  .seed = NULL
)
```

Details

O ‘pareamento’ existe porque o artigo original e a dissertação DISCORDAM, e as duas leituras se defendem. Em TERÇOS (o original, e o que o ‘trend’ faz) o miolo da série é descartado: são os extremos que acumularam a tendência, e parear vizinho dilui o contraste. Em METADES (a dissertação, seção 3.3.4) cada observação é pareada com a que está meia série adiante — entram mais pares, cada um com contraste menor. O param não é cosmético: os dois caminhos dão estatísticas diferentes na mesma série, e é por isso que a escolha vai na nota.

`tr_series_decompose` *Decomposição clássica por médias móveis.*

Description

A multiplicativa pede série positiva: o sazonal é um fator, e um fator calculado sobre zero ou negativo sai infinito ou com o sinal trocado, sem erro. Recusar é o que manda a pessoa para a aditiva, ou para o log.

Usage

```
tr_series_decompose(serie, tipo = "aditiva")
```

`tr_series_deseasonalize`*Tira a sazonalidade da série, com os efeitos estimados.*

Description

Os efeitos de cada período saem de uma regressão em ‘período’ (dummies, contraste à escolha). Com ****controlar_tendencia****, a tendência linear entra no MESMO ajuste (‘valor $\sim t + \text{período}$ ’) e só o sazonal sai da série: numa série com tendência, a média de dezembro fica acima da de janeiro só porque dezembro vem depois, e as dummies sozinhas chamariam isso de sazonalidade.

Usage

```
tr_series_deseasonalize(  
    serie,  
    controlar_tendencia = TRUE,  
    contraste = "soma_zero",  
    excluir = "",  
    remover_ns = FALSE,  
    confianca = 0.95  
)
```

`tr_series_detect_interventions`*Procura intervenções que ninguém declarou: o método de Chen & Liu (1993).*

Description

Ajusta um ARIMA (o do ‘modelo’ ligado ou um automático), calcula para cada instante e cada tipo a estatística t do efeito, marca os que passam do valor crítico, reajusta e descarta os que deixaram de ser significativos. A conta é a do ‘tsoutliers::tso()’; a coleção traduz os tipos e o tempo.

Usage

```
tr_series_detect_interventions(  
    serie,  
    modelo = NULL,  
    pulso = TRUE,  
    degrau = TRUE,  
    temporaria = TRUE,  
    inovacional = FALSE,  
    valor_critico = 0  
)
```

<code>tr_series_dettrend</code>	<i>Tira a tendência da série, e diz qual tendência tirou.</i>
---------------------------------	---

Description

É o "estimo num lm e depois subtraio" num nó só. A sazonalidade FICA — quem quer tirá-la junto usa a decomposição —, e a tendência estimada vai pendurada na série como atributo 'tendencia' (um 'ts' no mesmo tempo): é o que permite conferir, no console, que 'saída + tendência' devolve a série.

Usage

```
tr_series_dettrend(série, metodo = "linear", grau = 2L, suavidade = 0.75)
```

Details

O polinômio é ortogonal ('poly()'), e não cru como o da 'series/regression': aqui só o AJUSTE importa, não a leitura dos coeficientes, e 't' cru numa série de 144 meses tem número de condição astronômico — o 'lm' perderia dígitos sem avisar.

Faltante não impede o ajuste ('na.exclude'): a tendência é prevista em todo t, e a saída tem NA só onde a série tinha.

'diferenca' é o outro jeito de tirar tendência, e o único sem modelo: a "tendência" é o valor do período anterior, e a série perde a 1ª observação — começa um período depois, com o calendário certo.

<code>tr_series_diff</code>	<i>Diferença simples ou sazonal.</i>
-----------------------------	--------------------------------------

Description

'tipo = "sazonal"' usa a frequência como defasagem, e é aqui que a série anual precisa de erro: 'diff(x, lag = 1)' com outro nome não é diferença sazonal, e o card afirmaria uma coisa que a conta não fez.

Usage

```
tr_series_diff(série, tipo = "simples", ordem = 1L)
```

<code>tr_series_errors</code>	<i>Classes de erro da coleção ‘series’.</i>
-------------------------------	---

Description

Mesmo princípio de ‘trama::tr_errors()’ e das coleções irmãs, e mesmo teste de duas direções: uma tabela que um teste confere contra o código, pra não envelhecer em silêncio.

Usage

```
tr_series_errors()
```

Details

Série temporal tem um modo de falha próprio, e é por ele que metade destas classes existe: o resultado ERRADO que sai com cara de certo. Uma série com um mês faltando vira uma série com o calendário deslocado dali em diante; uma agregação anual de série que começa em março soma anos que não existem; uma diferença sazonal numa série anual é uma diferença de lag 1 com outro nome. Nenhum desses erra no R — todos saem verdes. Aqui viram card vermelho.

Value

data.frame com ‘class’ e ‘when’.

<code>tr_series_ets</code>	<i>Suavização exponencial em espaço de estados (ETS).</i>
----------------------------	---

Description

O modelo é o código de três letras da literatura — erro, tendência, sazonalidade — , e não três enums, porque é assim que ele aparece em todo livro e em todo resumo (“ETS(M,Ad,M)”: quem leu ‘MAM’ no resumo de um ajuste automático copia o código para fixá-lo.

Usage

```
tr_series_ets(serie, modelo = "ZZZ", amortecida = "auto")
```

tr_series_example	<i>Uma das séries do pacote ‘datasets’.</i>
-------------------	---

Description

É a metade que o ‘data/example’ recusa de propósito: dos conjuntos do R, 31 são ‘ts’, e lá eles pintariam o card de vermelho no ‘store’ de ‘data/table’. Aqui eles são o assunto.

Usage

```
tr_series_example(dataset = "AirPassengers")
```

Details

Série múltipla vira uma opção por coluna (‘EuStockMarkets\$DAX’): o tipo é univariado, e um param a mais só para escolher a coluna seria um campo que metade das opções ignora.

tr_series_fisher	<i>Fisher: existe uma periodicidade escondida?</i>
------------------	--

Description

O irmão do ‘series/seasonality_kw’ que faz a pergunta ao contrário. O Kruskal-Wallis compara estações que VOCÊ já declarou — ele precisa da frequência da série para saber o que é janeiro. Este aqui não declara nada: ele varre o periodograma inteiro e pergunta se o MAIOR pico é maior do que o acaso produziria. Quem sabe qual é o ciclo usa o Kruskal-Wallis; quem está caçando um ciclo desconhecido usa este.

Usage

```
tr_series_fisher(serie, remover = "reta")
```

Details

E é dessa varredura que vem a única armadilha da página: o teste acha o maior pico ONDE QUER QUE ELE ESTEJA, e nem todo pico é estação. No ‘Nile’ — uma série anual sem sazonalidade nenhuma — ele rejeita, e o pico está no período 100, que é a série inteira. Isso não é um ciclo: é a tendência que sobrou. Por isso o bloco publica o PERÍODO junto com o veredito, e avisa na ‘nota’ quando o pico não chega a se repetir dentro da série.

tr_series_forecast	<i>Previsão de um modelo ajustado, com intervalos de 80 e 95%.</i>
--------------------	--

Description

Os níveis são fixos: são os que todo gráfico de previsão mostra, e os que o adaptador para tabela nomeia ('li_80', 'ls_95'). Um param de nível mudaria o nome das colunas a jusante, e um 'data/filter' escrito sobre 'ls_95' quebraria ao trocar o nível no card.

Usage

```
tr_series_forecast(
  modelo = NULL,
  horizonte = 12L,
  intervalo = "normal",
  ajuste = NULL,
  futuro = NULL,
  var = NULL,
  .seed = NULL
)
```

tr_series_from_table	<i>Monta a série a partir de uma tabela.</i>
----------------------	--

Description

A frequência é declarada AQUI, uma vez, e viaja com a série — é o motivo de o tipo 'series/ts' existir.

Usage

```
tr_series_from_table(
  dados,
  valor = "",
  tempo = "",
  frequencia = 12L,
  inicio = ""
)
```

Details

Com uma coluna de tempo, o nó ORDENA por ela e confere a grade. As duas recusas (tempo repetido, período faltando) são o coração do nó: um 'ts' é só um vetor com início e frequência, e não sabe que a linha de maio sumiu. Sem a checagem, junho passaria a ser chamado de maio e todo o calendário dali em diante sairia deslocado de um mês — a sazonalidade estimada, a previsão e o gráfico sazonal errados, e o card verde.

A grade só é conferida onde há calendário a conferir: data com frequência 12, 4 ou 1, e ano inteiro com frequência 1. Para o resto (diária com ciclo semanal, por exemplo) a coluna só ordena, e a ajuda diz isso.

‘início’ e uma coluna de datas ao mesmo tempo é recusado, e não resolvido por precedência: seriam duas fontes para o mesmo fato, e a que perdesse discordaria da tela sem aviso.

`tr_series_from_table_mts`

Tabela larga -> série múltipla: uma coluna por série, um tempo comum.

Description

Reaproveita o ‘series/from_table’ coluna a coluna, para que as regras de calendário (tempo repetido, buraco, início) sejam as mesmas, ditas com as mesmas mensagens.

Usage

```
tr_series_from_table_mts(
  dados,
  valores = character(),
  tempo = "",
  frequencia = 12L,
  inicio = ""
)
```

`tr_series_holt_winters`

Holt-Winters clássico, pelos mínimos quadrados do ‘stats’.

Description

Fica ao lado do ETS, que o generaliza, porque é o método que se ensina e o que se cita: tirar a tendência ou a sazonalidade é desligar uma chave, e o efeito na previsão aparece no card seguinte.

Usage

```
tr_series_holt_winters(
  serie,
  tendencia = TRUE,
  sazonalidade = TRUE,
  tipo = "aditiva"
)
```

`tr_series_interpolate`*Preenche faltantes pela estrutura da série.*

Description

Existe ao lado do ‘data/replace_na’ porque faz outra coisa: não põe uma constante, estima o valor pelo vizinho no tempo e, em série sazonal, pela mesma estação dos outros anos (‘forecast::na.interp’). Preencher o julho faltante com zero derruba a sazonalidade inteira; com a interpolação, o julho sai com cara de julho.

Usage

```
tr_series_interpolate(serie)
```

`tr_series_intervencao`*Declara uma intervenção na série, para o modelo seguinte estimar.*

Description

Devolve a mesma série, com a intervenção somada às que já vinham declaradas. ‘pulso’ é o outlier aditivo (AO), ‘degrau’ a mudança de nível (LS), ‘rampa’ a mudança de inclinação e ‘inovacional’ o choque que passa pela dinâmica do modelo (IO). Com ‘dinamica = ”gradual”’ (pulso e degrau), o efeito entra por $/(1 - B)$, de Box & Tiao (1975).

Usage

```
tr_series_intervencao(serie, data, tipo = "degrau", dinamica = "imediata")
```

`tr_series_join`*Junta séries univariadas numa múltipla.*

Description

Os fios variádicos não trazem nome: o nome vem de ‘nomes’ (separados por vírgula, na ordem dos fios) ou, sem ele, do atributo ‘nome’ que a série carrega desde a origem (‘series/example’ com ‘EuStockMarkets\$DAX’, ‘series/from_table’) — e, faltando os dois, ‘serie_1’, ‘serie_2’...

Usage

```
tr_series_join(series, nomes = "")
```

Details

Frequências diferentes são recusadas: alinhar mensal com trimestral pede agregação, e é o ‘series/aggregate’ que a faz, explicitamente. Janelas diferentes, não: o recorte é a interseção, e a nota diz quanto se perdeu.

tr_series_kpss	<i>KPSS: a série é estacionária?</i>
----------------	--------------------------------------

Description

O outro lado do ADF: aqui H_0 é a ESTACIONARIEDADE, e a rejeição vem da cauda de CIMA (‘sentido = "maior"’). É o único teste da coleção assim, e é por causa dele que o campo existe — sem ele a regra de decisão viveria num ‘if’ por nome de teste.

Usage

```
tr_series_kpss(serie, deterministico = "constante")
```

tr_series_kruskal_wallis	<i>Kruskal-Wallis: a série tem sazonalidade?</i>
--------------------------	--

Description

O não paramétrico da sazonalidade, e o primeiro bloco da categoria: põe TODAS as observações em postos e pergunta se a soma dos postos muda de uma estação para outra. Janeiro sempre alto e julho sempre baixo afastam as somas, e o H cresce. O F da sazonalidade (‘series/regression’ $\rightarrow$ ‘models/anova_table’) responde à mesma pergunta pedindo erro normal em troca.

Usage

```
tr_series_kruskal_wallis(serie)
```

Details

A sazonalidade que ele enxerga é DETERMINÍSTICA: o padrão que se repete igual todo ciclo. A ESTOCÁSTICA, a que vai mudando de ano para ano, não tem um “nível de janeiro” fixo para o posto encontrar e passa por aqui sem ser vista — a dissertação separa as duas na seção 3.4.

E o que mais engana: TENDÊNCIA ATRAPALHA. Numa série que cresce, os postos altos são todos dos anos finais e se espalham por todas as estações, que é por que o ‘AirPassengers’ — a série sazonal de livro — não rejeita aqui. Tirar a tendência antes (‘series/diff’) resolve; passar um log NÃO, porque o log é monotônico e não troca a ordem de valor nenhum, então não mexe em posto nenhum. O detalhe está na página do nó, com os números medidos.

tr_series_lag	<i>Defasa a série k períodos: o valor de t passa a morar em $t + k$.</i>
---------------	---

Description

Os valores não mudam, só o tempo — que é exatamente o que o ‘ts’ guarda. Isso importa para quem vai juntar a série defasada com a original numa tabela (‘data/join’ por ‘tempo’): é o join que alinha ‘x_t’ com ‘x_t-k’. ‘k’ negativo adianta.

Usage

```
tr_series_lag(serie, k = 1L)
```

tr_series_lag_plot	<i>Gráfico de defasagens: ‘x_t’ contra ‘x_t-k’, um painel por k.</i>
--------------------	---

Description

É o correlograma sem o resumo: cada painel mostra a nuvem cuja correlação é uma barra do ACF, e aí se vê o que o número esconde — relação curva, um ponto sozinho puxando tudo, dois regimes. Numa série sazonal os pontos saem coloridos pela estação, e o painel do ciclo (12 na mensal) é o que fica colado na diagonal.

Usage

```
tr_series_lag_plot(  
  serie,  
  defasagens = 0L,  
  aspecto = "16:9",  
  tema = "padrão",  
  titulo = "",  
  rotulo_x = "",  
  rotulo_y = "",  
  legenda = "direita"  
)
```

`tr_series_ljung_box` *Ljung-Box: a série é ruído branco?*

Description

O corrigido para amostra pequena, e por isso o que se reporta — o ‘series/box_pierce’ testa a mesma hipótese nula pela fórmula original.

Usage

```
tr_series_ljung_box(serie, defasagens = 0L, graus = 0L)
```

Details

O uso típico é nos RESÍDUOS de um modelo, e aí ‘graus’ importa: os parâmetros ajustados ($p + q + P + Q$ de um ARIMA) consomem graus de liberdade, e sem descontá-los o teste fica generoso demais — aceita ruído branco onde há estrutura.

`tr_series_mann_kendall`
Mann-Kendall: a série tem tendência?

Description

O teste de tendência mais usado em climatologia, e não paramétrico: conta, par a par, quantas vezes o futuro supera o passado. Não supõe distribuição nenhuma, que é por que ele é o padrão justamente onde a série não é normal — o F da tendência (‘series/regression’ → ‘models/anova_table’) mede a mesma coisa pedindo erro normal em troca.

Usage

```
tr_series_mann_kendall(serie, correcao = "nenhuma", .seed = NULL)
```

Details

A tendência que ele enxerga é MONOTÔNICA: numa série que sobe e depois desce, os pares de um lado cancelam os do outro e o teste pode sair sem tendência nenhuma. A direção, quando rejeita, está no SINAL de Z, e é ela que vai para a conclusão — um bilateral que só dissesse “há tendência” jogaria fora o que o usuário foi perguntar.

tr_series_moving_average

Média móvel de ordem k — a tendência a olho, sem modelo.

Description

Centrada com ordem par é a média $2 \times k$ (a de ordem 12 numa série mensal é a que remove a sazonalidade inteira), que é o que ‘forecast::ma()’ faz e o que a decomposição clássica usa por dentro. As bordas saem NA — metade da janela para cada lado —, e não encurtadas: é o honesto, e o gráfico mostra.

Usage

```
tr_series_moving_average(serie, ordem = 12L, centrada = TRUE)
```

tr_series_ndiffs

Quantas diferenças a série pede para ficar estacionária.

Description

É a pergunta que se faz ANTES de montar o ‘series/diff’ ou de fixar o ‘d’ e o ‘D’ de um ARIMA à mão. As diferenças sazonais vêm do teste de força sazonal do ‘forecast’, e só existem para série com ciclo — na anual a linha sai NA com o motivo, e não zero, que afirmaria “testei e não precisa”.

Usage

```
tr_series_ndiffs(serie, teste = "kpss")
```

tr_series_pacf

Autocorrelação parcial: a correlação com a defasagem k , descontadas as intermediárias.

Description

Autocorrelação parcial: a correlação com a defasagem k , descontadas as intermediárias.

Usage

```
tr_series_pacf(
  serie,
  defasagens = 0L,
  aspecto = "16:9",
  tema = "padrão",
  titulo = "",
  rotulo_x = "",
  rotulo_y = "",
  legenda = "direita"
)
```

tr_series_pettitt	<i>Pettitt: a série tem um ponto de mudança?</i>
-------------------	--

Description

O único dos quatro não paramétricos desta categoria que NÃO pergunta por tendência. A hipótese nula é HOMOGENEIDADE: que ‘Z₁..Z_t’ e ‘Z_{t+1}..Z_N’ venham da mesma população. Rejeitar localiza uma RUPTURA — a dissertação é enfática nisso, citando Niel et al. (1998) —, e não autoriza dizer que a série sobe ou desce. Os três irmãos de categoria testam tendência, então a conclusão daqui é o lugar mais fácil da coleção para um copiar-colar mentir: ela nomeia QUANDO a série mudou e cala sobre direção, de propósito.

Usage

```
tr_series_pettitt(serie, correcao = "nenhuma", .seed = NULL)
```

tr_series_phillips_perron	<i>Phillips-Perron: a série tem raiz unitária?</i>
---------------------------	--

Description

Mesma H₀ do ADF — raiz unitária —, e mesma leitura invertida: rejeitar é concluir estacionária. Difere no meio do caminho: em vez de acrescentar defasagens à regressão, corrige a estatística para a autocorrelação dos resíduos.

Usage

```
tr_series_phillips_perron(serie, deterministico = "tendência")
```

tr_series_pick	<i>Uma coluna da série múltipla, de volta ao mundo univariado.</i>
----------------	--

Description

Uma coluna da série múltipla, de volta ao mundo univariado.

Usage

```
tr_series_pick(series, variavel = "")
```

tr_series_plot	<i>A série no tempo, e opcionalmente uma segunda série no mesmo eixo.</i>
----------------	---

Description

‘sobreposta’ é para VER uma série estimada sobre a original antes de usá-la — a tendência que se vai subtrair, a média móvel, a previsão de outro modelo. Mesmo eixo Y, e não dois painéis: a pergunta é se a tendência acompanha a série, e isso só se lê com as duas uma sobre a outra.

Usage

```
tr_series_plot(
  serie,
  pontos = FALSE,
  sobreposta = NULL,
  nome_serie = "",
  nome_sobreposta = "",
  aspecto = "16:9",
  tema = "padrão",
  titulo = "",
  rotulo_x = "",
  rotulo_y = "",
  legenda = "direita"
)
```

Details

Frequências diferentes são recusadas, como no ‘series/combine’: uma mensal e uma anual no mesmo eixo parecem comparáveis ponto a ponto, e não são. Janelas diferentes, não: o eixo é a união, e cada linha ocupa o seu trecho.

‘sobreposta’ vem DEPOIS de ‘pontos’ para não mudar a posição dos argumentos de quem já chamava ‘tr_series_plot(x, TRUE)’ no console.

`tr_series_plot_decomposition`*Os quatro componentes, empilhados, cada um na sua escala.*

Description

Escala livre por painel, e é o que se tem de ler com cuidado: o sazonal com amplitude de ± 40 e o resto de ± 10 saem com a MESMA altura. A escala de cada painel está no eixo, e é ela que diz o tamanho do componente.

Usage

```
tr_series_plot_decomposition(  
  decomposicao,  
  aspecto = "4:3",  
  tema = "padrão",  
  titulo = "",  
  rotulo_x = "",  
  rotulo_y = "",  
  legenda = "direita"  
)
```

`tr_series_plot_forecast`*O histórico e a previsão, com os leques de 80 e 95%.*

Description

‘historico’ corta o passado mostrado, e só ele: com 40 anos de série mensal e 12 meses previstos, o leque vira um risco na ponta direita. Não muda a previsão — ela já foi feita no nó anterior.

Usage

```
tr_series_plot_forecast(  
  previsao,  
  historico = 0L,  
  aspecto = "16:9",  
  tema = "padrão",  
  titulo = "",  
  rotulo_x = "",  
  rotulo_y = "",  
  legenda = "direita"  
)
```

tr_series_plot_mts	<i>Uma faixa por série, eixo y livre: níveis diferentes não se espremem.</i>
--------------------	--

Description

Uma faixa por série, eixo y livre: níveis diferentes não se espremem.

Usage

```
tr_series_plot_mts(  
  serie,  
  aspecto = "16:9",  
  tema = "padrão",  
  titulo = "",  
  rotulo_x = "",  
  rotulo_y = "",  
  legenda = "direita"  
)
```

tr_series_range_mean	<i>Diagnóstico amplitude-média para variância não constante.</i>
----------------------	--

Description

Agrupa observações consecutivas, calcula média e amplitude de cada bloco e testa a inclinação da regressão amplitude ~ média.

Usage

```
tr_series_range_mean(  
  serie,  
  tamanho = stats::frequency(serie),  
  aspecto = "4:3",  
  tema = "claro",  
  titulo = "",  
  rotulo_x = "Média do bloco",  
  rotulo_y = "Amplitude do bloco",  
  legenda = "direita"  
)
```

tr_series_regression *Regressão da série por FÓRMULA, e a previsão dela.*

Description

O que a clássica e a STL não dão: coeficiente, erro-padrão e p-valor. A fórmula é escrita sobre nomes que o bloco constrói a partir da série — ‘valor’ (a série), ‘t’ (1, 2, ..., n), ‘período’ (fator do ciclo: jan, ..., dez), ‘ano’ (o ano civil de cada observação) e ‘regressor’ (a série ligada na entrada de mesmo nome) —, e é por serem construídos AQUI que a previsão sabe estendê-los: o futuro de ‘t’, ‘período’ e ‘ano’ é conhecido; o do ‘regressor’, não, e vem pela entrada ‘futuro’ do ‘series/forecast’.

Usage

```
tr_series_regression(
  serie,
  formula = "valor ~ t + periodo",
  contraste = "soma_zero",
  regressor = NULL,
  erro = "independente",
  ar = 1L,
  ma = 0L,
  excluir = "",
  remover_ns = FALSE,
  confianca = 0.95
)
```

Details

O ajuste sai como ‘models/fit’ (contrato da ‘trama.models’): coeficientes, quadro da ANOVA, estatísticas e diagnóstico são os blocos de lá. Os componentes — tendência, sazonal, efeito do regressor e resto — viajam no mesmo objeto, e o ‘series/component’ os lê pelo adaptador ‘models/fit -> series/decomposition’.

O ‘contraste’ muda como os coeficientes de ‘período’ são LIDOS, e não a decomposição. Com ‘soma_zero’, cada coeficiente sazonal é o desvio do período em relação à média; com ‘categoria_base’, é a diferença para o primeiro período (o ‘summary()’ do R). Em ambos o componente sazonal devolvido é centrado em zero e a tendência absorve a média.

tr_series_residuals *Os resíduos do ajuste, como série — para testar e para ver.*

Description

Série, e não tabela, porque o destino natural é ‘series/ljung_box’ e ‘series/acf’: o diagnóstico de um modelo é perguntar se o que sobrou ainda tem estrutura temporal.

Usage

```
tr_series_residuals(modelo)
```

tr_series_runs	<i>Run (Wald-Wolfowitz): a série é aleatória?</i>
----------------	---

Description

Conta SEQUÊNCIAS (runs): trechos seguidos de valores do mesmo lado da mediana. Poucas sequências é sinal de que os valores se agrupam; muitas, de que alternam. H0 aqui é ALEATORIEDADE, e não ausência de tendência — o que separa este bloco dos dois irmãos de categoria, que perguntam por tendência direto. Rejeitar diz que a série não foi gerada ao acaso; tendência é UMA das causas possíveis, ao lado de mudança de nível, ciclo e agrupamento, e é por isso que a conclusão não fala em tendência.

Usage

```
tr_series_runs(serie)
```

tr_series_seasonal_plot	<i>Gráfico sazonal: um traço por ano, percorrendo o ciclo.</i>
-------------------------	--

Description

Responde duas perguntas de uma vez: qual é o FORMATO da sazonalidade (o pico é em julho?) e se ele MUDA com o tempo (os anos recentes, de cor mais clara, repetem a forma dos antigos?). A cor é contínua pelo ano para que a mudança se leia como degradê — na escala contínua do TEMA, que o gráfico não fixa: trocar ‘contínua’ no tema do projeto troca este degradê também.

Usage

```
tr_series_seasonal_plot(  
  serie,  
  aspecto = "16:9",  
  tema = "padrão",  
  titulo = "",  
  rotulo_x = "",  
  rotulo_y = "",  
  legenda = "direita"  
)
```

tr_series_stl	<i>Decomposição STL (Cleveland et al., 1990).</i>
---------------	---

Description

‘janela_sazonal = 0’ é “periódica”: o mesmo padrão sazonal em todos os anos, o que aproxima a clássica. Um número (ímpar, ≥ 7) é quantos anos entram na suavização de cada estação — menor deixa a sazonalidade mudar mais depressa. O zero como “periódica” evita um param de texto que aceitasse “periodic” e mais nada.

Usage

```
tr_series_stl(serie, janela_sazonal = 0L, robusta = FALSE)
```

Details

‘robusta’ troca a média pela suavização com pesos que ignoram outlier: um mês de greve deixa de puxar a sazonalidade de todos os outros anos — e vai parar no resto, que é onde ele deve aparecer.

tr_series_subseries	<i>Subséries: um painel por estação, com os anos em sequência e a média.</i>
---------------------	--

Description

O complemento do gráfico sazonal: aquele mostra o formato do ciclo, este mostra como CADA estação evoluiu. Um julho que sobe mais que os outros meses — sazonalidade que muda — é invisível no sazonal e óbvio aqui.

Usage

```
tr_series_subseries(
  serie,
  aspecto = "16:9",
  tema = "padrão",
  titulo = "",
  rotulo_x = "",
  rotulo_y = "",
  legenda = "direita"
)
```

Details

Até frequência 24: com 52 painéis semanais o gráfico vira uma parede.

<code>tr_series_transform</code>	<i>Log, raiz ou Box-Cox: estabiliza a variância que cresce com o nível.</i>
----------------------------------	---

Description

Log de zero é ‘-Inf’, e de negativo é ‘NaN’ com um aviso que não chega ao card: a série sairia com buracos que parecem faltantes. Recusar com a contagem é o que manda a pessoa somar uma constante antes (na ‘data’), com a escolha de quanto somar feita às claras.

Usage

```
tr_series_transform(serie, metodo = "log", lambda = "")
```

Details

‘lambda’ em branco é Box-Cox AUTOMÁTICO (Guerrero), e o escolhido fica pendurado na série como atributo — é o que o ‘forecast’ usa, e o que o resumo mostraria se alguém perguntar “qual ele escolheu?”.

<code>tr_series_var</code>	<i>VAR(p) por mínimos quadrados equação a equação (Lütkepohl 2005, cap. 3).</i>
----------------------------	---

Description

‘defasagens = 0’ escolhe p pelo critério em ‘criterio’ (até ‘max_defasagens’), a mesma conta do ‘series/var_select’; com p fixo, ‘criterio’ é ignorado. Sazonais (dummies centradas do ‘vars’) só em série com ciclo.

Usage

```
tr_series_var(
  series,
  defasagens = 0L,
  max_defasagens = 8L,
  criterio = "AIC",
  deterministico = "constante",
  sazonal = FALSE
)
```

tr_series_window	<i>Recorta a série entre dois períodos.</i>
------------------	---

Description

‘fim’ só com o ano quer dizer o ÚLTIMO período daquele ano. Sem isto, ‘fim = "1956"’ numa série mensal pararia em janeiro de 1956 — que é o que o ‘window()’ do R entende, e não o que ninguém quis dizer.

Usage

```
tr_series_window(serie, inicio = "", fim = "")
```

Details

Período fora da série é erro, não recorte silencioso até a borda: um ‘inicio = "1995"’ digitado no lugar de ‘"1959"’ devolveria a série inteira e o card verde.

tr_series_zivot_andrews	<i>Zivot-Andrews: raiz unitária, com a quebra estimada pelo próprio teste?</i>
-------------------------	--

Description

Mesma H0 do ‘series/adf’ — raiz unitária — e mesma leitura invertida: rejeitar é concluir estacionária. O que o justifica é o VIÉS dos outros dois: ADF e Phillips-Perron tendem a NÃO rejeitar quando a série tem quebra estrutural (MARGARIDO, 2001, citado na dissertação), porque uma série estacionária em torno de uma média deslocada se parece com passeio aleatório para os dois. Aqui a data da quebra é ESTIMADA, não informada: o teste varre os cortes entre 15 estacionariedade. A janela é a do teste publicado, e não a do ‘urca’, que varre todos os cortes.

Usage

```
tr_series_zivot_andrews(
  serie,
  mudanca = "ambas",
  defasagens = -1L,
  selecao = "fixa"
)
```

Details

É o segundo bloco da coleção a publicar um ponto localizado na série, depois do ‘series/pettitt’, e usa o mesmo ‘tr_series_rotulo_em()’ para o rótulo.

trama_collection	<i>A coleção 'series'.</i>
------------------	----------------------------

Description

Carrega DEPOIS de 'trama.data' e 'trama.view': as portas usam 'data/table' e 'view/plot', e o registro recusa porta com tipo desconhecido. É a mesma dependência de ordem que a 'view' tem com a 'data', travada por teste em 'test-catalogo.R'.

Usage

```
trama_collection()
```

Details

Os ids de categoria têm prefixo ('serie_*') porque o registro de categorias é GLOBAL ('registry\$categories[[id]]'): uma categoria 'fonte' aqui sobrescreveria em silêncio a de outra coleção que usasse o mesmo id.

Index

tr_series_accuracy, [3](#)
tr_series_acf, [4](#)
tr_series_adf, [4](#)
tr_series_aggregate, [4](#)
tr_series_arima, [5](#)
tr_series_as_decomposition, [5](#)
tr_series_baseline, [6](#)
tr_series_box_pierce, [6](#)
tr_series_combine, [7](#)
tr_series_component, [7](#)
tr_series_cox_stuart, [8](#)
tr_series_decompose, [8](#)
tr_series_deseasonalize, [9](#)
tr_series_detect_interventions, [9](#)
tr_series_detrend, [10](#)
tr_series_diff, [10](#)
tr_series_errors, [11](#)
tr_series_ets, [11](#)
tr_series_example, [12](#)
tr_series_fisher, [12](#)
tr_series_forecast, [13](#)
tr_series_from_table, [13](#)
tr_series_from_table_mts, [14](#)
tr_series_holt_winters, [14](#)
tr_series_interpolate, [15](#)
tr_series_intervencao, [15](#)
tr_series_join, [15](#)
tr_series_kpss, [16](#)
tr_series_kruskal_wallis, [16](#)
tr_series_lag, [17](#)
tr_series_lag_plot, [17](#)
tr_series_ljung_box, [18](#)
tr_series_mann_kendall, [18](#)
tr_series_moving_average, [19](#)
tr_series_ndiffs, [19](#)
tr_series_pacf, [19](#)
tr_series_pettitt, [20](#)
tr_series_phillips_perron, [20](#)
tr_series_pick, [21](#)
tr_series_plot, [21](#)
tr_series_plot_decomposition, [22](#)
tr_series_plot_forecast, [22](#)
tr_series_plot_mts, [23](#)
tr_series_range_mean, [23](#)
tr_series_regression, [24](#)
tr_series_residuals, [24](#)
tr_series_runs, [25](#)
tr_series_seasonal_plot, [25](#)
tr_series_stl, [26](#)
tr_series_subseries, [26](#)
tr_series_transform, [27](#)
tr_series_var, [27](#)
tr_series_window, [28](#)
tr_series_zivot_andrews, [28](#)
trama_collection, [29](#)