Package: localdatasus (via r-universe)

October 16, 2026

Type Package

Title Neighborhood-Level Data from Brazil's Unified Health System

Version 0.2.0

Description Retrieves health data from Brazil's Unified Health System (Sistema Único de Saúde, SUS) at the neighborhood level, with 2022 Census population, rates per 10,000 inhabitants, coordinates and maps. Simple functions return deaths, live births, notifiable diseases, hospital admissions, outpatient consultations, emergency (SAMU) calls and health facilities by neighborhood, choosing the data source automatically: microdata with the patient's postal code (CEP), downloaded with the 'microdatasus' package of Saldanha, Bastos and Barcellos (2019) <doi:10.1590/0102-311x00032419>; files and open data published by health departments; and regional 'TabNet' tabulators that tabulate by neighborhood or district. Postal codes and neighborhood names are linked to the official neighborhoods of the Brazilian Institute of Geography and Statistics (IBGE) through the National Address Register for Statistical Purposes (CNEFE) and the 2022 Census tracts, with quality rules for generic postal codes. Function names, arguments, messages and documentation are in Portuguese, aimed at Brazilian health researchers.

URL https://github.com/Hendesson/localdatasus, https://hendesson.github.io/projetos/localdatasus/

BugReports https://github.com/Hendesson/localdatasus/issues

License MIT + file LICENSE

Encoding UTF-8

Language pt-BR

Depends R (>= 4.1.0)

Imports curl, data.table, jsonlite, microdatasus, stats, tools, utils

Suggests ggplot2, knitr, leaflet, rmarkdown, sf, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 7.3.2

NeedsCompilation no

Author Hendesson Alves [aut, cre] (ORCID: <https://orcid.org/0009-0009-0342-9201>)

Maintainer Hendesson Alves <hendesson124@gmail.com>

Repository https://cran.r-universe.dev

Date/Publication 2026-10-16 12:50:13 UTC

RemoteUrl https://github.com/cran/localdatasus

RemoteRef HEAD

RemoteSha 8b9d2628adc62a5e58fd46907485a79b776104c6

Contents

adicionar_bairro . . . 3
agravos_bairro . . . 4
agregar_bairro . . . 5
ajuda . . . 6
atendimentos_bairro . . . 8
baixar_bairro . . . 9
cep_bairro . . . 10
cnes_coordenadas . . . 11
distritos_censo . . . 12
estabelecimentos_bairro . . . 13
fontes_tabnet . . . 14
internacoes_bairro . . . 14
ligar_bairros . . . 15
limpar_cache . . . 16
mapa_bairro . . . 17
mapa_interativo . . . 18
nascimentos_bairro . . . 19
obitos_bairro . . . 19
onde_tem_bairro . . . 20
populacao_bairro . . . 21
samu_bairro . . . 21
sih_bairro . . . 22
sistemas_bairro . . . 24
tabnet_bairro . . . 25
tabnet_opcoes . . . 26

Index **27**

`adicionar_bairro` *Adiciona bairro e coordenadas a uma tabela com CEP*

Description

Funciona com QUALQUER ‘data.frame‘ que tenha uma coluna de CEP — dados do DATASUS ou seus próprios dados. Cada linha recebe o bairro (pelo CNEFE 2022, ver [cep_bairro()]), as coordenadas do CEP e do bairro, e um diagnóstico da qualidade do CEP (‘situacao_cep‘).

Usage

```
adicionar_bairro(
  dados,
  uf,
  col_cep = "CEP",
  col_mun = NULL,
  col_id = NULL,
  pct_min = 0.6,
  limite_excesso = 20
)
```

Arguments

`dados`	Um ‘data.frame‘.
`uf`	Sigla da UF dos CEPs (a tabela CEP-bairro é montada por UF).
`col_cep`	Nome da coluna de CEP.
`col_mun`	Nome da coluna do município de residência (6 ou 7 dígitos), opcional. Permite detectar CEPs de outro município.
`col_id`	Nome de uma coluna que identifica a pessoa ou o episódio, opcional. Usada para que pessoas com muitos registros (ex.: diálise mensal) não sejam confundidas com CEPs genéricos.
`pct_min`	Fração mínima dos endereços de um CEP no mesmo bairro.
`limite_excesso`	Razão máxima entre a fração de pessoas/registros e a fração de endereços de um CEP (valor esperado: ~1).

Value

‘dados‘ com as colunas acima acrescentadas.

Colunas adicionadas

cep_paciente CEP normalizado (8 dígitos).

codmun_paciente município de residência (6 dígitos), se ‘col_mun‘ for dado.

situacao_cep se o CEP permitiu atribuir bairro e, se não, por quê.

id_bairro, bairro, fonte_bairro, cd_bairro_ibge o bairro (só quando ‘situacao_cep == "bairro atribuído"‘).

lat_bairro, lon_bairro ponto do bairro (mediana dos endereços).

lat_cep, lon_cep ponto do CEP (mediana dos endereços do CEP). Preenchido para todo CEP que existe no CNEFE — inclusive CEPs gerais, cujo ponto representa o centro da área que eles cobrem; use ‘situacao_cep‘ e ‘n_end_cep‘ para julgar.

codmun_cep, n_end_cep, pct_bairro município do CEP, nº de endereços do CEP e fração deles no bairro atribuído.

See Also

[cep_bairro()], com as fontes e referências da tabela CEP → bairro.

Examples

```
minha_tabela <- data.frame(id = 1:3, cep = c("20031170", "22041001", "24020005"))
adicionar_bairro(minha_tabela, "RJ", col_cep = "cep")
```

`agravos_bairro` *Casos de agravos de notificação por bairro*

Description

Conta casos do SINAN (dengue, tuberculose, sífilis, violência...) por bairro (ou distrito) de residência, com população, taxa por 10 mil e coordenadas. Fontes: TabNets da Prefeitura do Rio, da Prefeitura de São Paulo e da DIVE de Santa Catarina (todos os municípios) e, no Recife, os registros de dengue, chikungunya e zika publicados pela Prefeitura (a dengue traz os casos notificados; use ‘por = "classificacao"‘ para separar confirmados e descartados). Veja [onde_tem_bairro()].

Usage

```
agravos_bairro(agravo, municipio = NULL, anos, por = NULL, uf = NULL)
```

Arguments

`agravo`	Nome do agravo, em linguagem comum: ‘"dengue"‘, ‘"tuberculose"‘, ‘"sifilis congenita"‘, ‘"violencia"‘, ‘"zika"‘...
`municipio`	Nome ou código IBGE do município, por exemplo ‘"Rio de Janeiro"‘, ‘"Niterói"‘ ou ‘"355030"‘.
`anos`	Anos (de notificação ou de diagnóstico, conforme o site).
`por`	Variável para abrir as contagens, por exemplo ‘"sexo"‘.
`uf`	Sigla da UF, para pegar o estado inteiro (‘uf = "RJ"‘) ou para desfazer nomes repetidos de município.

Value

Como em [obitos_bairro()], com ‘casos‘ no lugar de ‘obitos‘.

Examples

```
agravos_bairro("dengue", "Rio de Janeiro", 2024)
agravos_bairro("tuberculose", "São Paulo", 2022:2024)
agravos_bairro("dengue", "Joinville", 2024, por = "sexo")
```

---

agregar_bairro *Conta registros por bairro e calcula taxas*

Description

Agrega o resultado de [baixar_bairro()] ou [adicionar_bairro()] por bairro (e por outras colunas, se pedido) e junta a população do Censo 2022 dos bairros oficiais para calcular a taxa por 10 mil habitantes. Registros sem bairro ficam numa linha ‘"(sem bairro)"‘ por município.

Usage

```
agregar_bairro(dados, por = NULL, contar_distintos = NULL)
```

Arguments

dados	Saída de [baixar_bairro()] ou [adicionar_bairro()].
por	Nomes de colunas extras para agrupar, por exemplo ‘c("ano_ref", "AP_CIDPRI")‘.
contar_distintos	Nome de uma coluna para contar valores DISTINTOS (ex.: ‘"id_paciente"‘, para contar pacientes em vez de registros). ‘NULL‘ = conta registros.

Value

Um ‘data.frame‘ com o bairro, as colunas de ‘por‘, ‘n‘, ‘populacao‘ e ‘taxa_10mil‘.

Examples

```
quimio <- baixar_bairro("SIA-AQ", "RJ", 2024)
agregar_bairro(quimio, por = "ano_ref", contar_distintos = "id_paciente")
```

`ajuda` *Ajuda sobre erros e avisos do pacote*

Description

Todo erro ou aviso do localdatasus vem com um código, por exemplo ‘[LDS-03]‘. ‘ajuda("LDS-03")‘ explica o que aconteceu e como resolver. Sem argumento, lista todos os códigos.

Usage

```
ajuda(codigo = NULL)
```

Arguments

`codigo`	Código do erro, como ‘"LDS-03"‘, ‘"03"‘ ou ‘3‘.

Details

O pacote trabalha em silêncio e mostra só um resumo no fim. Para ver cada etapa (consultas, downloads), use ‘options(localdatasus.detalhes = TRUE)‘.

Value

Um objeto da classe ‘localdatasus_ajuda‘, impresso na tela: a explicação do código (ou a lista de todos os códigos).

Códigos de erro e aviso

LDS-01: UF inválida A UF precisa ser UMA sigla válida, como "RJ" ou "SP". **Como resolver:** Use a sigla com duas letras: uf = "RJ".

LDS-02: Município não encontrado O nome (ou código) do município não bate com nenhum município do Censo 2022. **Como resolver:** Confira a grafia (acentos e maiúsculas não importam) ou use o código IBGE: municipio = "330455".

LDS-03: Município com nome repetido Existe mais de um município com esse nome no Brasil (ex.: Bom Jardim, em MA, PE e RJ). **Como resolver:** Diga também a UF: obitos_bairro("Bom Jardim", 2023, uf = "RJ").

LDS-04: Lugar não informado A função precisa saber ONDE: um município ou uma UF. **Como resolver:** Exemplos: municipio = "Rio de Janeiro" ou uf = "RJ".

LDS-05: Não há dado por bairro para esse lugar Só algumas secretarias publicam dados por bairro. Para esse lugar e esse tipo de dado, não há fonte pública conhecida. **Como resolver:** Veja o que existe com onde_tem_bairro(). Para internações, use internacoes_bairro(), que funciona em todas as UFs (pelo CEP).

LDS-06: Agravo não disponível Esse agravo não está no TabNet desse lugar. **Como resolver:** Veja os agravos de cada lugar: subset(onde_tem_bairro(), funcao == "agravos_bairro()").

LDS-07: Agravo ambíguo O nome dado combina com mais de um agravo (ex.: "sifilis" = gestante e congênita). **Como resolver:** Seja mais específico: "sifilis congenita" ou "sifilis gestante".

LDS-08: CID escrita de forma inválida A CID deve ser letras (capítulos: "I", "J") OU códigos de 3 caracteres ("I21", "I20-I25"), sem misturar os dois tipos. Intervalos ficam dentro de uma letra. **Como resolver:** Exemplos válidos: cid = "I"; cid = c("I", "J"); cid = "I21"; cid = "I20-I25".

LDS-09: CID não disponível nesta fonte A fonte não permite filtrar por causa, ou nenhum dos códigos pedidos aparece nela (pode não ter havido caso). **Como resolver:** Tente um capítulo inteiro (cid = "I") ou veja as opções: tabnet_opcoes("RIO-SIM", "filtros", filtro = "Causa (CID10 3C)").

LDS-10: Variável de 'por' não encontrada O texto de 'por' não corresponde a nenhuma variável que a fonte permite cruzar. **Como resolver:** Veja as variáveis possíveis: tabnet_opcoes("RIO-SIM", "colunas") (troque pela fonte certa; veja onde_tem_bairro()).

LDS-11: Ano fora da base A fonte não tem dados para algum dos anos pedidos; esses anos foram deixados de fora. **Como resolver:** Veja os anos disponíveis: tabnet_opcoes("RIO-SIM", "anos").

LDS-12: Nenhum registro encontrado A consulta funcionou, mas não há nenhum registro com esses filtros, lugar e anos. **Como resolver:** Amplie o período, tire filtros ou confira se o agravo teve casos nesses anos.

LDS-13: Falha de conexão O site da secretaria (ou do IBGE/DATASUS) não respondeu. Pode estar fora do ar ou lento. **Como resolver:** Espere alguns minutos e tente de novo. O pacote já tenta 3 vezes. Consultas que já deram certo ficam em cache.

LDS-14: O site mudou ou respondeu algo inesperado A página do TabNet não veio no formato esperado. Ou o site está instável, ou mudou o formulário. **Como resolver:** Tente de novo mais tarde. Se continuar, avise os autores do pacote: o cadastro da fonte pode precisar de ajuste.

LDS-15: Filtro ou opção do TabNet não encontrado O rótulo pedido não existe no formulário do site, ou combina com mais de uma opção. **Como resolver:** Veja as opções exatas com tabnet_opcoes(fonte, "filtros", filtro = "..."). Pode usar só o começo do texto.

LDS-16: Fonte ou sistema desconhecido O código da fonte (TabNet) ou do sistema (microdados) não está no cadastro do pacote. **Como resolver:** Veja as opções com fontes_tabnet() ou sistemas_bairro().

LDS-17: Coluna não encontrada na tabela A tabela passada não tem a coluna indicada. **Como resolver:** Confira o nome da coluna com names(sua_tabela).

LDS-18: SINASC-RJ só existe para o RJ Os microdados de nascimentos com CEP e bairro são publicados só pela SES-RJ. **Como resolver:** Para outras UFs, use nascimentos_bairro() com a capital de SP, ou o TabNet da sua secretaria.

LDS-19: Arquivo do IBGE não encontrado O arquivo do IBGE para esse município ou UF não foi achado no servidor. **Como resolver:** Confira o código do município. Se o problema continuar, o IBGE pode ter mudado o nome do arquivo.

LDS-20: Faltam pacotes para mapas Os mapas usam os pacotes sf e ggplot2 (estático) ou sf e leaflet (interativo), que não estão instalados. **Como resolver:** Instale uma vez: install.packages(c("sf", "ggplot2", "leaflet")).

LDS-21: Tabela sem dados de lugar para o mapa A tabela não tem código IBGE do bairro nem coordenadas (lat/lon). **Como resolver:** Use a tabela devolvida por obitos_bairro(), nascimentos_bairro(), agravos_bairro() ou agregar_bairro().

LDS-22: Poucos registros ganharam bairro Uma parte grande dos registros ficou sem bairro, geralmente por CEP genérico (um só CEP para a cidade toda) ou CEP do hospital no lugar do da residência. **Como resolver:** Não é um erro. Compare bairros dentro do mesmo município e confira attr(x, "cobertura").

LDS-23: Mais de um ano ou categoria no mapa interativo O mapa interativo mostra um valor por bairro. Com vários anos ou categorias, o pacote soma tudo. **Como resolver:** Para um ano só, filtre antes: mapa_interativo(subset(x, ano == 2023)).

Examples

```
ajuda()
ajuda("LDS-03")
ajuda(3)
```

`atendimentos_bairro` *Atendimentos médicos por bairro (Curitiba)*

Description

Conta os atendimentos médicos da rede municipal de saúde de Curitiba (unidades de saúde, UPAs e outros serviços do Sistema e-Saúde) por bairro de residência do paciente, com população, taxa por 10 mil habitantes e coordenadas. Fonte: portal de dados abertos da Prefeitura de Curitiba (2019 em diante).

Usage

```
atendimentos_bairro(
  municipio = "Curitiba",
  anos,
  meses = 1:12,
  cid = NULL,
  por = NULL
)
```

Arguments

`municipio`	Só ‘"Curitiba"‘ (residentes de Curitiba).
`anos`	Anos do atendimento.
`meses`	Meses do atendimento (1 a 12).
`cid`	CID do atendimento: letras para capítulos (‘"J"‘) ou códigos (‘"J45"‘, ‘"J00-J06"‘).
`por`	Variável para abrir as contagens: ‘"sexo"‘, ‘"faixa etaria"‘, ‘"tipo de unidade"‘ (UPA, unidade de saúde...), ‘"profissional"‘ (especialidade do médico), ‘"internamento"‘ ou ‘"unidade de saude"‘ (o estabelecimento do atendimento).

Details

Cada arquivo publicado tem cerca de 480 MB e cobre três meses; a primeira consulta de um ano baixa quatro arquivos. Os meses ficam no cache, e as consultas seguintes são rápidas.

Value

Como em [obitos_bairro()], com ‘atendimentos‘ no lugar de ‘obitos‘.

Examples

```
atendimentos_bairro("Curitiba", 2025, meses = 1:3, cid = "J")
atendimentos_bairro("Curitiba", 2025, meses = 6, por = "tipo de unidade")
```

---

baixar_bairro *Baixa uma base com CEP já com bairro e coordenadas*

Description

Baixa os microdados de uma base que tem o CEP do paciente (ver [sistemas_bairro()]) e devolve TODOS os registros e colunas originais, acrescidos de bairro, coordenadas e qualidade do CEP (ver [adicionar_bairro()]).

Usage

```
baixar_bairro(
  sistema,
  uf,
  ano_inicio,
  ano_fim = ano_inicio,
  mes_inicio = 1,
  mes_fim = 12,
  vars = NULL,
  pct_min = 0.6,
  limite_excesso = 20
)
```

Arguments

`sistema`	Código da base, por exemplo ‘"SIH-RD"‘ ou ‘"SIA-AQ"‘ (veja [sistemas_bairro()]).
`uf`	Sigla da UF.
`ano_inicio, ano_fim`	Anos de competência (de nascimento, no SINASC-RJ).
`mes_inicio, mes_fim`	Meses (do primeiro e do último ano).
`vars`	Colunas a baixar (‘NULL‘ = todas). As colunas necessárias para localizar o bairro são incluídas automaticamente.

`pct_min`	Fração mínima dos endereços de um CEP no mesmo bairro.
`limite_excesso`	Razão máxima entre a fração de pessoas/registros e a fração de endereços de um CEP (valor esperado: ~1).

Details

Os arquivos do DATASUS são mensais, por mês de COMPETÊNCIA (processamento), e por UF do ESTABELECIMENTO: pacientes de outras UFs atendidos na UF aparecem com ‘situacao_cep = "residente de outra UF"‘. O ‘"SINASC-RJ"‘ (SES-RJ) vem em arquivos anuais por data de nascimento.

Quando a base traz também o NOME do bairro (SINASC-RJ), os registros que não recebem bairro pelo CEP são ligados pelo nome, com [ligar_bairros()]. A coluna ‘origem_bairro‘ diz de onde veio o bairro de cada registro.

Colunas extras: ‘data_ref‘ e ‘ano_ref‘ (data de internação no SIH; competência de realização nas APACs; data de nascimento no SINASC-RJ), ‘origem_bairro‘ e, nas APACs, ‘id_paciente‘ — o cartão SUS pseudonimizado (‘AP_CNSPCN‘) convertido em texto legível, que permite seguir o mesmo paciente entre meses.

Value

Um ‘data.frame‘, um registro por linha; o atributo ‘"cobertura"‘ resume ‘situacao_cep‘.

Examples

```
quimio <- baixar_bairro("SIA-AQ", "RJ", 2024, mes_inicio = 1, mes_fim = 3)
internacoes <- baixar_bairro("SIH-RD", "RJ", 2023)
nascimentos <- baixar_bairro("SINASC-RJ", "RJ", 2023)
```

`cep_bairro` *Tabela de CEPs e bairros de uma UF*

Description

Monta, a partir do CNEFE 2022 (todos os endereços do Censo), a ligação entre cada CEP e o bairro onde estão seus endereços.

Usage

```
cep_bairro(uf, manter_cnefe = FALSE)
```

Arguments

`uf`	Sigla da UF, por exemplo ‘"RJ"‘.
`manter_cnefe`	Se ‘TRUE‘, mantém o arquivo do CNEFE no cache depois de processado (ocupa espaço). Padrão: ‘FALSE‘.

Details

O bairro de cada endereço vem do bairro OFICIAL do IBGE (pelo setor censitário), quando o município tem bairros oficiais; senão, da localidade informada no próprio endereço do CNEFE. Cada CEP recebe o bairro onde está a MAIORIA dos seus endereços.

Na primeira vez baixa o CNEFE da UF (de ~10 MB no AC a ~1 GB em SP) e guarda só a tabela resultante no cache; as próximas chamadas são instantâneas.

Value

Uma lista com dois ‘data.frame‘s:

ceps ‘cep‘, ‘codmun_cep‘, ‘id_bairro‘, ‘pct_bairro‘ (fração dos endereços do CEP que está nesse bairro), ‘n_end_cep‘ (nº de endereços do CEP) e ‘lat_cep‘, ‘lon_cep‘ (mediana das coordenadas dos endereços do CEP).

bairros ‘id_bairro‘, ‘codmun‘, ‘bairro‘, ‘fonte_bairro‘, ‘cd_bairro_ibge‘, ‘lat_bairro‘, ‘lon_bairro‘ (mediana das coordenadas dos endereços do bairro) e ‘n_enderecos‘.

References

Instituto Brasileiro de Geografia e Estatística. Coordenadas geográficas dos endereços no Censo Demográfico 2022: nota metodológica n. 01. Rio de Janeiro: IBGE; 2024.

Empresa Brasileira de Correios e Telégrafos. Endereçamento de correspondências: guia técnico. Versão 1.4. Brasília: Correios; 2021. (Estrutura do CEP e sufixos especiais.)

Abordagem relacionada, com serviços de geocodificação: Rocha TAH, Silva NC, Amaral PVM et al. Geolocalização de internações cadastradas no Sistema de Informações Hospitalares do Sistema Único de Saúde: uma solução baseada no programa estatístico R. Epidemiologia e Serviços de Saúde 2018; 27(4):e2017444. doi:10.5123/s1679497420180004000016.

Examples

```
tab <- cep_bairro("AC")
head(tab$ceps)
```

---

cnes_coordenadas *Coordenadas de estabelecimentos de saúde (CNES)*

Description

Consulta a API de Dados Abertos do Ministério da Saúde e devolve nome, tipo, endereço e coordenadas de cada estabelecimento. Útil para ligar os registros (coluna ‘CNES‘ no SIH, ‘AP_CODUNI‘ nas APACs) ao local de atendimento — por exemplo, para medir a distância entre a residência (‘lat_cep‘, ‘lon_cep‘) e o hospital.

Usage

```
cnes_coordenadas(cnes)
```

Arguments

`cnes`	Vetor de códigos CNES (com ou sem zeros à esquerda).

Details

Os resultados ficam em cache; só CNES novos são consultados.

Value

Um ‘data.frame‘ com ‘cnes‘ (7 dígitos), ‘nome‘, ‘tipo_unidade‘, ‘codmun‘, ‘cep‘, ‘endereco‘, ‘bairro‘, ‘lat‘ e ‘lon‘. Coordenadas que caem fora do Brasil viram ‘NA‘. Atenção: o cadastro tem erros (ex.: hospitais com coordenadas em outro município); confira ‘codmun‘.

Examples

```
cnes_coordenadas(c("2269880", "2270234"))
```

`distritos_censo` *População e coordenadas dos distritos de um município (Censo 2022)*

Description

Distritos são a divisão oficial dos municípios usada, por exemplo, pela Prefeitura de São Paulo nos seus TabNets ("Distrito Administrativo"). A população vem dos setores censitários; as coordenadas são a mediana dos endereços do CNEFE 2022 de cada distrito (baixa só o CNEFE do município, uma vez).

Usage

```
distritos_censo(codmun)
```

Arguments

`codmun`	Código do município (6 ou 7 dígitos).

Value

Um ‘data.frame‘ com ‘codmun‘, ‘cd_distrito_ibge‘, ‘distrito‘, ‘populacao‘, ‘lat_distrito‘ e ‘lon_distrito‘.

Examples

```
distritos_censo("355030")   # cidade de São Paulo
```

estabelecimentos_bairro

Estabelecimentos de saúde por bairro

Description

Conta os estabelecimentos de saúde ativos por bairro, em qualquer município do país, a partir do cadastro do CNES publicado no Portal de Dados Abertos do SUS. O bairro vem do CEP do estabelecimento (pelo CNEFE 2022) e, quando o CEP não resolve, do nome do bairro informado no cadastro. Mede a oferta de serviços no bairro, não onde moram os pacientes.

Usage

```
estabelecimentos_bairro(
  municipio = NULL,
  uf = NULL,
  tipo = NULL,
  so_sus = FALSE,
  por = NULL
)
```

Arguments

`municipio`	Nome ou código IBGE do município.
`uf`	Sigla da UF, para pegar o estado inteiro ou desfazer nomes repetidos.
`tipo`	Parte do nome do tipo de unidade, por exemplo ‘"hospital"‘, ‘"unidade basica"‘, ‘"pronto atendimento"‘, ‘"psicossocial"‘ (CAPS). ‘NULL‘ = todos os tipos.
`so_sus`	‘TRUE‘ para contar só os estabelecimentos com atendimento ambulatorial pelo SUS.
`por`	‘"tipo"‘ para abrir a contagem por tipo de unidade.

Value

Como em [obitos_bairro()], com ‘estabelecimentos‘ no lugar de ‘obitos‘ e ‘ano‘ igual ao ano do cadastro baixado. A taxa é por 10 mil habitantes. ‘attr(x, "registros")‘ traz cada estabelecimento, com nome, tipo, CEP, coordenadas e bairro.

Examples

```
estabelecimentos_bairro("Recife", tipo = "unidade basica")
estabelecimentos_bairro("Belo Horizonte", por = "tipo", so_sus = TRUE)
```

fontes_tabnet *TabNets regionais com dados por bairro*

Description

Lista as tabulações de secretarias estaduais e municipais que descem ao bairro (ou ao distrito) de residência e que podem ser consultadas com [tabnet_bairro()]. São dados AGREGADOS (contagens), não registros.

Usage

```
fontes_tabnet()
```

Details

Levantamento feito em out/2026 nos TabNets das 27 UFs e das capitais: a maioria vai só até o município. Para acrescentar uma fonte nova, copie uma linha desta tabela, ajuste-a e passe-a para [tabnet_bairro()].

Value

Um ‘data.frame‘, uma linha por fonte (ver o código-fonte para o significado de cada coluna).

Examples

```
fontes_tabnet()[, c("fonte", "descricao", "unidade")]
```

internacoes_bairro *Internações por bairro*

Description

Conta as internações do SUS (AIHs aprovadas) de residentes por bairro, pela data de internação, a partir do CEP do paciente. Atalho para [sih_bairro()] com a mesma sintaxe das outras funções simples.

Usage

```
internacoes_bairro(municipio = NULL, anos, cid = NULL, uf = NULL)
```

Arguments

municipio	Nome ou código IBGE do município, por exemplo ‘"Rio de Janeiro"‘, ‘"Niterói"‘ ou ‘"355030"‘.
anos	Anos de internação.
cid	Diagnóstico principal: letras (‘"I"‘) ou códigos (‘"I21"‘).
uf	Sigla da UF. Opcional quando ‘municipio‘ é dado (a UF sai do município); com ‘municipio = NULL‘, pega a UF inteira.

Value

Um ‘data.frame‘ por bairro e ano com ‘internacoes‘, ‘obitos_hosp‘, ‘populacao‘, ‘taxa_10mil‘ e coordenadas.

Examples

```
internacoes_bairro("Rio de Janeiro", 2023, cid = c("I", "J"), uf = "RJ")
```

---

ligar_bairros — *Liga nomes de bairro aos bairros do IBGE, com coordenadas e população*

Description

Recebe nomes de bairro escritos à mão ou vindos de outros sistemas (por exemplo, de um TabNet ou do campo "bairro" de uma declaração) e procura, DENTRO de cada município, o bairro correspondente: primeiro o nome idêntico (ignorando acentos, abreviações e preposições), depois o nome sem a numeração final ("Santa Rosa II" -> "Santa Rosa"), depois um nome que contém o outro em palavras inteiras ("Complexo da Maré" -> "Maré") e, por fim, o nome mais parecido (distância de edição de até 15 tamanho).

Usage

```
ligar_bairros(
  nome,
  codmun,
  uf,
  unidade = c("bairro", "distrito"),
  distancia_max = 0.15
)
```

Arguments

nome	Vetor de nomes de bairro.
codmun	Vetor (do mesmo tamanho, ou de tamanho 1) com o código do município de cada nome (6 ou 7 dígitos).
uf	Sigla da UF.
unidade	‘"bairro"‘ ou ‘"distrito"‘.
distancia_max	Distância de edição máxima, relativa ao tamanho do nome, para a ligação aproximada.

Details

Os bairros candidatos são os bairros oficiais do IBGE e, nos municípios sem bairros oficiais, as localidades do CNEFE (ver [cep_bairro()]). Com ‘unidade = "distrito"‘, são os distritos do IBGE dos municípios em ‘codmun‘ (ver [distritos_censo()]).

Value

Um ‘data.frame‘ com uma linha por nome: ‘nome‘, ‘codmun‘, ‘ligacao‘ (‘"exata"‘, ‘"sem numeração"‘, ‘"parcial"‘, ‘"aproximada"‘, ‘"não encontrado"‘ ou ‘"ignorado"‘), ‘id_unidade‘, ‘nome_ibge‘, ‘fonte_unidade‘, ‘cd_ibge‘, ‘lat‘, ‘lon‘ e ‘populacao‘ (só em unidades oficiais do IBGE).

References

Levenshtein VI. Binary codes capable of correcting deletions, insertions, and reversals. Soviet Physics Doklady 1966; 10:707-10. (Distância de edição, calculada com [utils::adist()].)

Christen P. Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection. Berlin: Springer; 2012. doi:10.1007/9783642311642.

Camargo Jr. KR, Coeli CM. Reclink: aplicativo para o relacionamento de bases de dados, implementando o método probabilistic record linkage. Cadernos de Saúde Pública 2000; 16(2):439-47. doi:10.1590/s0102311x2000000200014.

Examples

```
ligar_bairros(c("Jd. América", "TIJUCA", "Ignorado"), "330455", "RJ")
```

---

limpar_cache — *Apaga os arquivos baixados e as tabelas em cache*

Description

O pacote guarda em cache os arquivos do IBGE e as tabelas CEP-bairro já processadas, para não refazer o trabalho. Use esta função para liberar espaço ou forçar um novo download.

Usage

```
limpar_cache()
```

Value

Invisivelmente, o caminho da pasta de cache.

Examples

```
limpar_cache()
```

`mapa_bairro` *Mapa por bairro*

Description

Desenha um mapa a partir da tabela devolvida por [obitos_bairro()], [nascimentos_bairro()], [agravos_bairro()], [internacoes_bairro()] ou [agregar_bairro()]. Cada bairro é pintado pelo valor escolhido (por padrão, a taxa por 10 mil habitantes), com os contornos oficiais do IBGE (Censo 2022). Bairros sem contorno oficial aparecem como pontos.

Usage

```
mapa_bairro(dados, valor = NULL, titulo = NULL, cores = "Reds")
```

Arguments

`dados`	Tabela de uma das funções do pacote.
`valor`	Coluna que pinta o mapa, por exemplo ‘"obitos"‘. Padrão: a taxa por 10 mil habitantes.
`titulo`	Título do mapa (opcional).
`cores`	Paleta do ColorBrewer, por exemplo ‘"Reds"‘ (padrão), ‘"Blues"‘, ‘"YlOrRd"‘.

Details

Se a tabela tiver mais de um ano, ou uma variável de ‘por‘ (sexo, faixa etária...), o mapa é dividido em painéis.

O resultado é um gráfico ‘ggplot2‘ comum: dá para acrescentar títulos e temas com ‘+‘, e salvar com ‘ggplot2::ggsave()‘.

Precisa dos pacotes ‘sf‘ e ‘ggplot2‘.

Value

Um gráfico ‘ggplot2‘.

Examples

```
x <- obitos_bairro("Rio de Janeiro", 2023, cid = "I")
mapa_bairro(x)
mapa_bairro(x, valor = "obitos", titulo = "Óbitos por doenças circulatórias, 2023")
mapa_bairro(obitos_bairro("Rio de Janeiro", 2023, por = "sexo"))   # um painel por sexo
ggplot2::ggsave("mapa.png", width = 8, height = 6, dpi = 300)
```

mapa_interativo *Mapa interativo por bairro*

Description

Abre um mapa interativo (leaflet), com um mapa de ruas ao fundo: passe o mouse sobre um bairro para ver o nome e clique para ver contagem, taxa e população. Útil para explorar os dados.

Usage

```
mapa_interativo(dados, valor = NULL, cores = "Reds")
```

Arguments

dados	Tabela de uma das funções do pacote.
valor	Coluna que pinta o mapa, por exemplo ‘"obitos"‘. Padrão: a taxa por 10 mil habitantes.
cores	Paleta do ColorBrewer, por exemplo ‘"Reds"‘ (padrão), ‘"Blues"‘, ‘"YlOrRd"‘.

Details

O mapa mostra um valor por bairro. Se a tabela tiver vários anos ou categorias, as contagens são somadas (filtre antes, se preferir).

Precisa dos pacotes ‘sf‘ e ‘leaflet‘.

Value

Um mapa ‘leaflet‘ (aparece no Viewer do RStudio). Para salvar como página web: ‘htmlwidgets::saveWidget(mapa, "mapa.html")‘.

Examples

```
x <- agravos_bairro("dengue", "Rio de Janeiro", 2024)
mapa_interativo(x)
```

nascimentos_bairro *Nascimentos por bairro*

Description

Conta os nascidos vivos de mães residentes por bairro (ou distrito), ano a ano, com população, taxa por 10 mil e coordenadas. Fontes: TabNet das Prefeituras do Rio, de São Paulo e de Fortaleza; nos demais municípios do RJ, os microdados da SES-RJ, localizados pelo CEP e pelo nome do bairro da mãe.

Usage

```
nascimentos_bairro(municipio = NULL, anos, por = NULL, uf = NULL)
```

Arguments

municipio	Nome ou código IBGE do município, por exemplo ‘"Rio de Janeiro"‘, ‘"Niterói"‘ ou ‘"355030"‘.
anos	Anos de nascimento.
por	Variável para abrir as contagens: ‘"sexo"‘, ‘"tipo de parto"‘, ‘"idade da mae"‘, ‘"consultas"‘...
uf	Sigla da UF, para pegar o estado inteiro (‘uf = "RJ"‘) ou para desfazer nomes repetidos de município.

Value

Como em [obitos_bairro()], com ‘nascimentos‘ no lugar de ‘obitos‘.

Examples

```
nascimentos_bairro("Rio de Janeiro", 2023)
nascimentos_bairro("Niterói", 2022:2023, por = "tipo de parto")
```

obitos_bairro *Óbitos por bairro*

Description

Conta os óbitos de residentes por bairro (ou distrito), ano a ano, com população, taxa por 10 mil habitantes e coordenadas de cada bairro. A fonte é escolhida sozinha pelo lugar: TabNet da Prefeitura do Rio (2006+), da Prefeitura de São Paulo (distritos, 2006+), da Prefeitura de Fortaleza (1999+, só capítulos da CID) ou da SES-RJ (demais municípios do RJ, bairro a partir de 2011). Veja [onde_tem_bairro()].

Usage

```
obitos_bairro(municipio = NULL, anos, cid = NULL, por = NULL, uf = NULL)
```

Arguments

`municipio`	Nome ou código IBGE do município, por exemplo ‘"Rio de Janeiro"‘, ‘"Niterói"‘ ou ‘"355030"‘.
`anos`	Anos do óbito, por exemplo ‘2023‘ ou ‘2019:2023‘.
`cid`	Causa básica (CID-10): letras para capítulos (‘"I"‘ = circulatório, ‘c("I", "J")‘) ou códigos de 3 caracteres (‘"I21"‘, ‘"I20-I25"‘). ‘NULL‘ = todas as causas.
`por`	Variável para abrir as contagens, em linguagem comum: ‘"sexo"‘, ‘"faixa etaria"‘, ‘"raca"‘, ‘"escolaridade"‘...
`uf`	Sigla da UF, para pegar o estado inteiro (‘uf = "RJ"‘) ou para desfazer nomes repetidos de município.

Value

Um ‘data.frame‘ com ‘ano‘, ‘codigo_municipio‘, ‘municipio‘, ‘bairro‘ (ou ‘distrito‘), ‘bairro_no_site‘ (como o site escreve), a variável de ‘por‘, ‘obitos‘, ‘populacao‘ (Censo 2022), ‘taxa_por_10mil‘, ‘lat‘, ‘lon‘, ‘codigo_ibge‘, ‘ligacao‘ (como o nome foi ligado ao IBGE) e ‘fonte‘.

Examples

```
obitos_bairro("Rio de Janeiro", 2023, cid = "I")
obitos_bairro("Rio de Janeiro", 2019:2023, cid = c("I", "J"), por = "sexo")
obitos_bairro("Niterói", 2022, cid = "I20-I25")
obitos_bairro(uf = "RJ", anos = 2023)
obitos_bairro("São Paulo", 2023, por = "faixa etaria")
```

`onde_tem_bairro` *Onde há dados por bairro*

Description

Lista, em linguagem simples, que dados por bairro existem, para que lugar e para que anos.

Usage

```
onde_tem_bairro()
```

Value

Um ‘data.frame‘ com ‘dado‘ (o nome a usar, por exemplo, em ‘agravos_bairro("dengue", ...)‘), ‘descricao‘, ‘local‘, ‘unidade‘, ‘funcao‘ (a função que busca) e ‘fonte‘.

Examples

```
onde_tem_bairro()
```

---

populacao_bairro *População por bairro oficial (Censo 2022)*

Description

Soma a população dos setores censitários do Censo 2022 por bairro oficial do IBGE. Só municípios com bairros oficialmente delimitados (895 no Brasil) aparecem.

Usage

```
populacao_bairro(uf)
```

Arguments

uf	Sigla da UF, por exemplo ‘"RJ"‘.

Value

Um ‘data.frame‘ com ‘codmun‘ (6 dígitos), ‘municipio‘, ‘cd_bairro_ibge‘, ‘bairro‘, ‘populacao‘ e ‘pop_em_favela‘ (pessoas em setores de favelas e comunidades urbanas).

Examples

```
pop <- populacao_bairro("RJ")
```

---

samu_bairro *Chamados do SAMU por bairro (Recife e região metropolitana)*

Description

Conta as solicitações ao SAMU 192 do Recife por bairro do LOCAL DA OCORRÊNCIA (não da residência), com população, taxa por 10 mil habitantes e coordenadas. O SAMU do Recife é metropolitano: também há chamados de Jaboatão dos Guararapes, Olinda, Paulista e outros municípios. Fonte: portal de dados abertos da Prefeitura do Recife, 2016 em diante. As solicitações duplicadas são descartadas.

Usage

```
samu_bairro(municipio = "Recife", anos, tipo = NULL, por = NULL)
```

Arguments

`municipio`	Nome do município da ocorrência (padrão: Recife).
`anos`	Anos do chamado (2016 em diante).
`tipo`	Parte do tipo de ocorrência, por exemplo ‘"causas externas"‘, ‘"psiquiatrica"‘, ‘"cardiologica"‘, ‘"respiratoria"‘. ‘NULL‘ = todos.
`por`	Variável para abrir as contagens: ‘"tipo"‘, ‘"subtipo"‘, ‘"sexo"‘, ‘"origem do chamado"‘, ‘"motivo de finalizacao"‘ ou ‘"desfecho"‘.

Value

Como em [obitos_bairro()], com ‘chamados‘ no lugar de ‘obitos‘.

Examples

```
samu_bairro("Recife", 2024)
samu_bairro("Recife", 2024, tipo = "causas externas", por = "subtipo")
samu_bairro("Olinda", 2023:2024)
```

---

sih_bairro *Internações do SIH/SUS por bairro de residência*

Description

Baixa as AIHs aprovadas (arquivo RD do SIH/SUS) de uma UF, conta uma internação por AIH principal, localiza o bairro de residência de cada paciente pelo CEP e devolve a contagem por bairro, ano e grupo de causa.

Usage

```
sih_bairro(
  uf,
  ano_inicio,
  ano_fim = ano_inicio,
  cid = NULL,
  nivel = c("bairro", "internacao"),
  minimo = 0,
  meses_extras = 6,
  pct_min = 0.6,
  limite_excesso = 20
)
```

Arguments

`uf`	Sigla da UF, por exemplo ‘"RJ"‘.
`ano_inicio`, `ano_fim`	Anos de INTERNAÇÃO (inclusive).
`cid`	Vetor com prefixos da CID-10 do diagnóstico principal, por exemplo ‘c("I", "J")‘ (capítulos IX e X) ou ‘c("I21", "I22")‘. Cada prefixo vira um grupo no resultado. ‘NULL‘ (padrão) = todas as causas.
`nivel`	‘"bairro"‘ (padrão) devolve contagens por bairro; ‘"internacao"‘ devolve uma linha por internação, com CEP, bairro e coordenadas (ver [adicionar_bairro()]).
`minimo`	Se maior que 0, contagens entre 1 e ‘minimo - 1‘ viram ‘NA‘ (supressão de células pequenas). Padrão: 0 (não suprime).
`meses_extras`	Meses de competência baixados após ‘ano_fim‘.
`pct_min`	Fração mínima dos endereços de um CEP no mesmo bairro.
`limite_excesso`	Razão máxima entre a fração de pessoas/registros e a fração de endereços de um CEP (valor esperado: ~1).

Value

Com ‘nivel = "bairro"‘, um ‘data.frame‘ com ‘codmun‘, ‘id_bairro‘, ‘bairro‘, ‘fonte_bairro‘, ‘cd_bairro_ibge‘, ‘lat_bairro‘, ‘lon_bairro‘, ‘ano‘, ‘grupo‘, ‘internacoes‘, ‘obitos_hosp‘, ‘populacao‘ (só bairros oficiais), ‘taxa_10mil‘ e ‘suprimido‘.

Como o bairro é atribuído

Cada CEP é ligado ao bairro onde está a maioria dos seus endereços no CNEFE 2022 (ver [cep_bairro()]). Um CEP NÃO recebe bairro quando:

- é inválido ou ausente;
- é especial (final 900 a 999: grandes usuários, caixas postais);
- não existe no CNEFE 2022;
- o paciente mora em outra UF;
- tem internações demais para o número de endereços — típico do CEP geral do município usado como "padrão", ou do CEP do hospital (ver ‘limite_excesso‘);
- pertence a outro município que não o de residência;
- cobre vários bairros sem maioria clara (ver ‘pct_min‘).

Essas internações aparecem com ‘bairro = "(sem bairro)"‘, para que os totais por município continuem corretos. O motivo de cada caso fica no atributo ‘"cobertura"‘ do resultado.

Datas

Os arquivos do SIH são organizados por mês de COMPETÊNCIA (faturamento), mas a contagem é feita pela DATA DE INTERNAÇÃO. Por isso a função baixa também ‘meses_extras‘ meses depois de ‘ano_fim‘, para pegar as AIHs faturadas com atraso.

References

Instituto Brasileiro de Geografia e Estatística. Coordenadas geográficas dos endereços no Censo Demográfico 2022: nota metodológica n. 01. Rio de Janeiro: IBGE; 2024.

Empresa Brasileira de Correios e Telégrafos. Endereçamento de correspondências: guia técnico. Versão 1.4. Brasília: Correios; 2021. (Estrutura do CEP e sufixos especiais.)

Abordagem relacionada, com serviços de geocodificação: Rocha TAH, Silva NC, Amaral PVM et al. Geolocalização de internações cadastradas no Sistema de Informações Hospitalares do Sistema Único de Saúde: uma solução baseada no programa estatístico R. Epidemiologia e Serviços de Saúde 2018; 27(4):e2017444. doi:10.5123/s167949742018000400016.

Examples

```
# Internações por doenças circulatórias e respiratórias, Acre, 2023
x <- sih_bairro("AC", 2023, cid = c("I", "J"))
attr(x, "cobertura")
```

`sistemas_bairro` *Bases de dados disponíveis por bairro*

Description

Lista as bases cujo microdado público traz o CEP do paciente e que, por isso, podem ser localizadas por bairro, registro a registro, com [baixar_bairro()]: as do DATASUS (via microdatasus) e as publicadas por secretarias estaduais (SINASC do RJ). Para dados AGREGADOS por bairro de outras bases (SIM, SINAN...), veja [fontes_tabnet()].

Usage

```
sistemas_bairro()
```

Value

Um ‘data.frame‘ com o código do sistema (como no microdatasus), a descrição, as colunas usadas para CEP, município de residência, identificador, data e CID, a origem dos dados e a coluna com o nome do bairro (quando existe).

Examples

```
sistemas_bairro()
```

tabnet_bairro — *Consulta um TabNet regional por bairro*

Description

Baixa de um TabNet de secretaria estadual ou municipal (ver [fontes_tabnet()]) as contagens por bairro (ou distrito) de residência, ano a ano, e liga cada bairro ao bairro do IBGE (código, coordenadas e população) com [ligar_bairros()].

Usage

```
tabnet_bairro(
  fonte,
  anos,
  coluna = NULL,
  filtros = list(),
  incremento = NULL,
  municipios = NULL,
  ligar = TRUE
)
```

Arguments

`fonte`	Código da fonte (ver [fontes_tabnet()]) ou uma linha no mesmo formato, para usar um TabNet que ainda não está no cadastro.
`anos`	Anos (de notificação, óbito ou nascimento, conforme a base).
`coluna`	Rótulo de uma variável para abrir as contagens (ex.: ‘"Sexo"‘, ‘"Faixa Etária"‘); veja ‘tabnet_opcoes(fonte, "colunas")‘. ‘NULL‘ = só o total por bairro.
`filtros`	Lista nomeada: nome do filtro = rótulo(s) das opções aceitas, por exemplo ‘list("Causa (Cap CID10)" = c("IX", "X"))‘. O rótulo pode ser só o começo do texto. Veja ‘tabnet_opcoes(fonte, "filtros")‘.
`incremento`	O que contar, quando a fonte oferece mais de uma medida; veja ‘tabnet_opcoes(fonte, "incrementos")‘.
`municipios`	Códigos de município (6 dígitos) a consultar, nas fontes estaduais. ‘NULL‘ = todos os que têm registros.
`ligar`	Se ‘TRUE‘, liga os nomes aos bairros/distritos do IBGE.

Details

São dados AGREGADOS: não há registro individual nem CEP. Cada consulta cruza só duas variáveis (bairro x ‘coluna‘); para mais dimensões, use ‘filtros‘ e repita a chamada. As consultas ficam em cache.

Nas fontes estaduais em que o rótulo do bairro não diz o município (como a DIVE-SC), é feita uma consulta por município: primeiro descobre quais municípios têm registros no ano, depois consulta cada um. Use ‘municipios‘ para limitar.

Value

Um ‘data.frame‘ com ‘fonte‘, ‘ano‘, ‘codmun‘, ‘municipio‘, ‘bairro_tabnet‘ (como veio do site), ‘categoria‘ (valor de ‘coluna‘), ‘n‘ e, com ‘ligar = TRUE‘, as colunas de [ligar_bairros()] (‘ligacao‘, ‘id_unidade‘, ‘nome_ibge‘, ‘cd_ibge‘, ‘lat‘, ‘lon‘, ‘populacao‘) e ‘taxa_10mil‘ (por ano, sobre a população de 2022).

Examples

```
# Óbitos por doenças do aparelho circulatório, por bairro do Rio
tabnet_bairro("RIO-SIM", 2022:2023, coluna = "Faixa Etária",
              filtros = list("Causa (Cap CID10)" = "IX"))
# Dengue por bairro em Joinville (SC)
tabnet_bairro("SC-SINAN-DENGUE", 2024, municipios = "420910")
# Óbitos por distrito da cidade de São Paulo
tabnet_bairro("SP-SIM", 2023, coluna = "Sexo")
```

---

tabnet_opcoes *Opções de um TabNet regional*

Description

Mostra o que dá para pedir a [tabnet_bairro()]: as variáveis que podem ir na coluna, os filtros, as opções de um filtro, as medidas ("incrementos") e os anos disponíveis.

Usage

```
tabnet_opcoes(
  fonte,
  tipo = c("colunas", "filtros", "incrementos", "anos"),
  filtro = NULL
)
```

Arguments

fonte	Código da fonte (ver [fontes_tabnet()]) ou uma linha no mesmo formato.
tipo	‘"colunas"‘, ‘"filtros"‘, ‘"incrementos"‘ ou ‘"anos"‘.
filtro	Com ‘tipo = "filtros"‘, o nome de um filtro: lista as opções dele.

Value

Um vetor de rótulos.

Examples

```
tabnet_opcoes("RIO-SIM", "colunas")
tabnet_opcoes("RIO-SIM", "filtros", filtro = "Sexo")
```

Index

adicionar_bairro, 3
agravos_bairro, 4
agregar_bairro, 5
ajuda, 6
atendimentos_bairro, 8

baixar_bairro, 9

cep_bairro, 10
cnes_coordenadas, 11

distritos_censo, 12

estabelecimentos_bairro, 13

fontes_tabnet, 14

internacoes_bairro, 14

ligar_bairros, 15
limpar_cache, 16

mapa_bairro, 17
mapa_interativo, 18

nascimentos_bairro, 19

obitos_bairro, 19
onde_tem_bairro, 20

populacao_bairro, 21

samu_bairro, 21
sih_bairro, 22
sistemas_bairro, 24

tabnet_bairro, 25
tabnet_opcoes, 26