Package: rsdot (via r-universe)

May 20, 2026

Title Programmatic Access to Spatial Data from 'SDOT-PCM' Peru

Version 0.1.0

Author Paul Efren Santos Andrade [aut, cre, cph] (<https://orcid.org/0000-0002-6635-0375>)

Maintainer Paul Efren Santos Andrade <paulefrens@gmail.com>

Description Provides programmatic access to official geographic information from the Secretariat of Demarcation and Territorial Organization ('SDOT') of the Presidency of the Council of Ministers ('PCM') of Peru. Facilitates the download, reading, and manipulation in 'R' of vector layers on geographic modeling, National Institute of Statistics and Informatics ('INEI') census limits, population centers, infrastructure, risks, and hazards. IMPORTANT: 'INEI' census limit information is referential and does not constitute official territorial limits nor has legal demarcatory effect.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0)

Imports sf, utils

Suggests terra, ggplot2, testthat (>= 3.0.0), knitr, rmarkdown, dplyr, tidyr, stringr, scales, mockery

VignetteBuilder knitr

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

URL https://github.com/PaulESantos/rsdot, https://paulesantos.github.io/rsdot/

BugReports https://github.com/PaulESantos/rsdot/issues

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

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

Date/Publication 2026-03-21 02:44:14 UTC

RemoteUrl https://github.com/PaulESantos/rsdot

RemoteRef HEAD

RemoteSha f1aee26c8fa552f3c49075bc8e3000314c06c14e

Contents

`clear_rsdot_cache` *Clear package cache*

Description

Deletes all downloaded files in the rsdot cache directory.

Usage

```
clear_rsdot_cache()
```

Value

Invisible TRUE if deleted successfully.

Examples

```
## Not run:
clear_rsdot_cache()

## End(Not run)
```

`get_aerodromos` *Get Airfields of Peru*

Description

Downloads and loads information on airfields in Peru updated to the year 2022, prepared by the Ministry of Transport and Communications (MTC). Data includes POINT type geometry and information on airports, airfields, and heliports. Allows stepwise filtering by department, province, and district.

Usage

```
get_aerodromos(
  departamento = NULL,
  provincia = NULL,
  distrito = NULL,
  show_progress = TRUE,
  force_update = FALSE
)
```

Arguments

`departamento`	Character vector. Name(s) of the department(s) to download. Valid options: "AMAZONAS", "ANCASH", "APURIMAC", "AREQUIPA", "AYACUCHO", "CAJAMARCA", "CUSCO", "HUANUCO", "ICA", "JUNIN", "LA LIBERTAD", "LAMBAYEQUE", "LIMA", "LORETO", "MADRE DE DIOS", "MOQUEGUA", "PASCO", "PIURA", "PUNO", "SAN MARTIN", "TACNA", "TUMBES", "UCAYALI". Case-insensitive. If `NULL`, shows the list of available departments.
`provincia`	Character vector. Name(s) of the province(s) to filter within the specified department(s). Optional.
`distrito`	Character vector. Name(s) of the district(s) to filter within the specified province(s). Optional.
`show_progress`	Logical. If `TRUE` (default), shows informative messages about download progress. If `FALSE`, executes silently.
`force_update`	Logical. If `TRUE`, forces a new download of the file even if it exists in cache. Default `FALSE`.

Details

The function downloads data from OSF (Open Science Framework) and stores it in cache during the R session. Data is in GeoPackage (.gpkg) format.

Data Source:

- Source: Ministry of Transport and Communications (MTC)
- Year: 2022
- Level: Airfield/Airport/Heliport
- Application: Territorial planning and connectivity analysis

Hierarchical Filtering: Filters are applied in cascade:

1. Specified departments are loaded first
2. Then provinces are filtered (if specified)
3. Finally, districts are filtered (if specified)

Cache is stored in: `tempdir()/DEMARCA_cache/aerodromos/`

NOTE: Geometries are POINT type and represent the location of each airfield.

Value

An `sf` (simple feature) object with POINT type geometry containing airfield information, including:

- POINT geometry (airfield coordinates)
- `nombre`: Airfield name
- `label`: Full descriptive label
- `tipo`: Type of facility (INTERNATIONAL AIRPORT, AIRFIELD, HELIPORT)
- `codidep`: Department code
- `nombdep`: Department name
- `nombprov`: Province name
- `nombdist`: District name
- `escala`: Airfield scale
- `lat`: Latitude
- `lon`: Longitude
- `estado`: Operating status (OPERATIONAL, etc.)
- `administ`: Managing entity
- `jerarquia`: Hierarchy (NATIONAL, etc.)
- `titular`: Ownership type (PUBLIC, PRIVATE)

If multiple departments are requested, it returns a combined sf object.

References

Ministry of Transport and Communications (MTC). Airfields of Peru 2022.

DEMARCA OSF Repository: https://osf.io/qy4j6/

See Also

get_departamentos, get_provincias, get_distritos, read_sf

Examples

```
## Not run:
# See available departments
get_aerodromos()

# Load airfields of a full department
aero_cusco <- get_aerodromos(departamento = "CUSCO")

# Filter by specific province
aero_prov_cusco <- get_aerodromos(
  departamento = "CUSCO",
  provincia = "CUSCO"
)

# Filter by specific district
aero_san_sebastian <- get_aerodromos(
  departamento = "CUSCO",
  provincia = "CUSCO",
  distrito = "SAN SEBASTIAN"
)

# Load multiple departments
aero_sur <- get_aerodromos(
  departamento = c("CUSCO", "PUNO", "AREQUIPA")
)

# Visualization with ggplot2
library(ggplot2)
library(dplyr)

# Map of airfields by type
ggplot(aero_cusco) +
  geom_sf(aes(color = tipo, shape = tipo), size = 3) +
  scale_color_manual(
    values = c(
      "AEROPUERTO INTERNACIONAL" = "darkred",
      "AERODROMO" = "darkblue",
      "HELIPUERTO" = "darkgreen"
    ),
    name = "Type"
  ) +
  labs(
    title = "Airfields of the Department of Cusco",
    subtitle = "Updated 2022 - MTC",
    caption = "Source: MTC | Visor - SDOT"
  ) +
  theme_minimal()
```

```
# Airfields by ownership type
aero_cusco |>
  group_by(tipo, titular) |>
  summarise(n = n(), .groups = "drop") |>
  arrange(desc(n))

# Filter operational public airfields
aero_publicos <- aero_cusco |>
  filter(titular == "PUBLICA", estado == "OPERATIVO")

# View airfields by administrator
aero_cusco |>
  filter(!is.na(administ)) |>
  group_by(administ) |>
  summarise(n = n(), .groups = "drop") |>
  arrange(desc(n))

# Map of airfields with labels
ggplot(aero_cusco) +
  geom_sf(aes(color = tipo), size = 4) +
  geom_sf_text(
    aes(label = nombre),
    size = 2.5,
    nudge_y = 0.1,
    check_overlap = TRUE
  ) +
  scale_color_brewer(palette = "Set1", name = "Type") +
  labs(
    title = "Airfields of Cusco",
    subtitle = "MTC 2022",
    caption = "Source: MTC | Visor - SDOT"
  ) +
  theme_minimal()

## End(Not run)
```

`get_capitales` *Get District Capitals of Peru*

Description

Downloads and loads information on district capitals identified in the 2017 Population and Housing Census of the National Institute of Statistics and Informatics (INEI). Data includes demographic and geographic information of the capitals at the departmental, provincial, and district levels, with POINT type geometry.

Usage

```
get_capitales(
```

```
  departamento = NULL,
  provincia = NULL,
  distrito = NULL,
  show_progress = TRUE,
  force_update = FALSE
)
```

Arguments

departamento	Character vector. Name(s) of the department(s) to download. Valid options: "AMAZONAS", "ANCASH", "APURIMAC", "AREQUIPA", "AYACUCHO", "CAJAMARCA", "CALLAO", "CUSCO", "HUANCAVELICA", "HUANUCO", "ICA", "JUNIN", "LA LIBERTAD", "LAMBAYEQUE", "LIMA", "LORETO", "MADRE DE DIOS", "MOQUEGUA", "PASCO", "PIURA", "PUNO", "SAN MARTIN", "TACNA", "TUMBES", "UCAYALI". Case-insensitive. If NULL, shows the list of available departments.
provincia	Character vector. Name(s) of province(s) to filter the results (optional). It is applied after loading the data.
distrito	Character vector. Name(s) of district(s) to filter the results (optional). It is applied after loading the data.
show_progress	Logical. If TRUE (default), shows informative messages about download progress. If FALSE, executes silently.
force_update	Logical. If TRUE, forces a new download of the file even if it exists in cache. Default FALSE.

Details

The function downloads data from OSF (Open Science Framework) and stores it in cache during the R session. Data is in GeoPackage (.gpkg) format.

Cache is stored in: `tempdir()/DEMARCA_cache/capitales/`

NOTE: Geometries are POINT type and represent the approximate geographic location of the capitals according to the 2017 Census. Population data correspond to the census year.

Value

An sf (simple feature) object with POINT type geometry containing district capital information, including:

- POINT geometry (location of the capital)
- gid: Unique record identifier
- ubigeo: Geographic location code (6 digits)
- nombdep: Department name
- nombprov: Province name
- nombdist: District name
- codccpp: Population center code (10 digits)
- cen_pob: Population center name

- pob: Total population
- capital: Capital type (1=departmental, 2=provincial, 3=distrital)

If multiple departments are requested, it returns a combined sf object.

References

INEI. 2017 Population and Housing Census. National Institute of Statistics and Informatics.

SDOT Viewer: https://geosdot.servicios.gob.pe/visor/

See Also

get_centros_poblados, get_departamentos, read_sf

Examples

```
## Not run:
# See available departments
get_capitales()

# Load capitals of a department
capitales_cusco <- get_capitales(departamento = "CUSCO")

# Load multiple departments
capitales_sur <- get_capitales(
  departamento = c("CUSCO", "PUNO", "AREQUIPA")
)

# Filter by province
capitales_lima_prov <- get_capitales(
  departamento = "LIMA",
  provincia = "LIMA"
)

# Filter by district
capital_cusco_dist <- get_capitales(
  departamento = "CUSCO",
  provincia = "CUSCO",
  distrito = "CUSCO"
)

# Visualization with ggplot2
library(ggplot2)
library(dplyr)

# Map of capitals by hierarchy
ggplot(capitales_cusco) +
  geom_sf(aes(color = factor(capital), size = pob), alpha = 0.7) +
  scale_color_manual(
    values = c("1" = "red", "2" = "blue", "3" = "green"),
    labels = c("Departmental", "Provincial", "Distrital"),
    name = "Capital Type"
```

```
  ) +
  scale_size_continuous(range = c(1, 10), name = "Population") +
  labs(
    title = "Capitals of Cusco by Hierarchy",
    subtitle = "2017 Population and Housing Census",
    caption = "Source: INEI | Visor - SDOT"
  ) +
  theme_minimal()

# Population analysis by capital type
capitales_cusco |>
  st_drop_geometry() |>
  group_by(capital) |>
  summarise(
    n = n(),
    pob_total = sum(pob, na.rm = TRUE),
    pob_promedio = mean(pob, na.rm = TRUE),
    pob_mediana = median(pob, na.rm = TRUE)
  ) |>
  arrange(capital)

# Most populated capitals
capitales_cusco |>
  st_drop_geometry() |>
  arrange(desc(pob)) |>
  select(nombdist, cen_pob, pob, capital) |>
  head(10)

# Filter only provincial capitals
caps_prov <- capitales_cusco |>
  filter(capital == 2)

# Analysis by province
capitales_cusco |>
  st_drop_geometry() |>
  group_by(nombprov) |>
  summarise(
    n_capitales = n(),
    pob_total = sum(pob, na.rm = TRUE)
  ) |>
  arrange(desc(pob_total))

## End(Not run)
```

`get_centros_poblados` *Get Population Centers of Peru*

Description

Downloads and loads information on population centers of Peru from the 2017 Population and Housing Census of the National Institute of Statistics and Informatics (INEI). Data includes demographic and geographic information at the departmental, provincial, and district levels, with POINT type geometry.

Usage

```
get_centros_poblados(
  departamento = NULL,
  provincia = NULL,
  distrito = NULL,
  show_progress = TRUE,
  force_update = FALSE
)
```

Arguments

departamento	Character vector. Name(s) of the department(s) to download. Valid options: "AMAZONAS", "ANCASH", "APURIMAC", "AREQUIPA", "AYACUCHO", "CAJAMARCA", "CALLAO", "CUSCO", "HUANCAVELICA", "HUANUCO", "ICA", "JUNIN", "LA LIBERTAD", "LAMBAYEQUE", "LIMA", "LORETO", "MADRE DE DIOS", "MOQUEGUA", "PASCO", "PIURA", "PUNO", "SAN MARTIN", "TACNA", "TUMBES", "UCAYALI". Case-insensitive. If NULL, shows the list of available departments.
provincia	Character vector. Name(s) of province(s) to filter the results (optional). It is applied after loading the data.
distrito	Character vector. Name(s) of district(s) to filter the results (optional). It is applied after loading the data.
show_progress	Logical. If TRUE (default), shows informative messages about download progress. If FALSE, executes silently.
force_update	Logical. If TRUE, forces a new download of the file even if it exists in cache. Default FALSE.

Details

La función descarga datos desde OSF (Open Science Framework) y los almacena en caché durante la sesión de R. Los datos están en formato GeoPackage (.gpkg).

IMPORTANTE - Ubicación Referencial: La ubicación de los centros poblados es referencial. Las coordenadas pueden diferir de las registradas en otras bases de datos oficiales o levantamientos de campo específicos. Se recomienda validar las ubicaciones para estudios que requieran precisión espacial.

El caché se almacena en: `tempdir()/DEMARCA_cache/centros_poblados/`

NOTA: Las geometrías son tipo POINT y representan la ubicación aproximada de los centros poblados según el Censo 2017. El sistema de coordenadas es WGS 84 (EPSG:4326).

Value

An `sf` (simple feature) object with POINT type geometry containing population center information, including:

- `viv`: Total de viviendas
- `viv_part`: Viviendas particulares
- `viv_part_o`: Viviendas particulares ocupadas
- `viv_part_1`: Viviendas particulares con 1 o más ocupantes
- `pob_viv_pa`: Población en viviendas particulares
- `y`: Latitud (grados decimales)
- `x`: Longitud (grados decimales)
- `fuente`: Fuente de la información
- `revision`: Información de revisión
- `cap`: Indicador de capital
- `capital`: Tipo de capital si aplica
- `iddpto`: ID del departamento
- `idprov`: ID de la provincia

Si se solicitan múltiples departamentos, retorna un objeto sf combinado.

References

INEI. Censo de Población y Vivienda 2017. Instituto Nacional de Estadística e Informática.

Visor SDOT: https://geosdot.servicios.gob.pe/visor/

See Also

`get_capitales`, `get_departamentos`, `read_sf`

Examples

```
## Not run:
# Ver departamentos disponibles
get_centros_poblados()

# Cargar centros poblados de un departamento
cp_tacna <- get_centros_poblados(departamento = "TACNA")

# Cargar múltiples departamentos
cp_sur <- get_centros_poblados(
  departamento = c("TACNA", "MOQUEGUA")
)

# Filtrar por provincia
cp_cusco_prov <- get_centros_poblados(
  departamento = "CUSCO",
  provincia = "CUSCO"
```

```
)

# Visualización con ggplot2
library(ggplot2)
library(dplyr)

# Clasificar por tamaño poblacional
cp_tacna <- cp_tacna |>
  mutate(
    categoria = case_when(
      pob >= 5000 ~ "Grande (5000+)",
      pob >= 1000 ~ "Mediano (1000-4999)",
      pob >= 500 ~ "Pequeño (500-999)",
      TRUE ~ "Muy pequeño (<500)"
    )
  )

# Mapa de centros poblados por tamaño
ggplot(cp_tacna) +
  geom_sf(aes(color = categoria, size = pob), alpha = 0.6) +
  scale_size_continuous(range = c(0.5, 5)) +
  scale_color_brewer(palette = "Spectral", direction = -1) +
  labs(
    title = "Centros Poblados de Tacna",
    subtitle = "Clasificados por Tamaño - Censo 2017",
    color = "Categoría",
    size = "Población",
    caption = "Fuente: INEI | Visor - SDOT"
  ) +
  theme_minimal()

# Análisis de vivienda
cp_tacna |>
  st_drop_geometry() |>
  filter(viv > 0) |>
  summarise(
    centros = n(),
    pob_total = sum(pob, na.rm = TRUE),
    viv_total = sum(viv, na.rm = TRUE),
    pob_por_vivienda = pob_total / viv_total,
    tasa_ocupacion = sum(viv_part_o) / sum(viv_part) * 100
  )

# Distribución por distrito
cp_tacna |>
  st_drop_geometry() |>
  group_by(nombdist) |>
  summarise(
    n_centros = n(),
    poblacion = sum(pob, na.rm = TRUE),
    viviendas = sum(viv, na.rm = TRUE)
  ) |>
  arrange(desc(poblacion))
```

```
# Análisis de densidad de vivienda
cp_tacna |>
  st_drop_geometry() |>
  filter(viv > 0) |>
  mutate(pob_por_viv = pob / viv) |>
  summarise(
    min = min(pob_por_viv),
    q1 = quantile(pob_por_viv, 0.25),
    mediana = median(pob_por_viv),
    promedio = mean(pob_por_viv),
    q3 = quantile(pob_por_viv, 0.75),
    max = max(pob_por_viv)
  )

## End(Not run)
```

`get_centros_poblados_crecimiento` *Get Population Centers with Population Growth Rate*

Description

Downloads and loads information on population centers with their average annual intercensal population growth rate (2007-2017) from the DEMARCA OSF repository. Data includes POINT type geometry and allows identifying changes in population volume for each population center. Allows stepwise filtering by department, province, and district.

Usage

```
get_centros_poblados_crecimiento(
  departamento = NULL,
  provincia = NULL,
  distrito = NULL,
  show_progress = TRUE,
  force_update = FALSE
)
```

Arguments

`departamento`	Character vector. Name(s) of the department(s) to download. Valid options: "AMAZONAS", "ANCASH", "APURIMAC", "AREQUIPA", "AYACUCHO", "CAJAMARCA", "CALLAO", "CUSCO", "HUANCAVELICA", "HUANUCO", "ICA", "JUNIN", "LA LIBERTAD", "LAMBAYEQUE", "LIMA", "LORETO", "MADRE DE DIOS", "MOQUEGUA", "PASCO", "PIURA", "PUNO", "SAN MARTIN", "TACNA", "TUMBES", "UCAYALI". Case-insensitive. If `NULL`, shows the list of available departments.

provincia	Character vector. Name(s) of the province(s) to filter within the specified department(s). Optional.
distrito	Character vector. Name(s) of the district(s) to filter within the specified province(s). Optional.
show_progress	Logical. If TRUE (default), shows informative messages about download progress. If FALSE, executes silently.
force_update	Logical. If TRUE, forces a new download of the file even if it exists in cache. Default FALSE.

Details

The function downloads data from OSF (Open Science Framework) and stores it in cache during the R session. Data is in GeoPackage (.gpkg) format.

Data Methodology:

- Source: INEI Population Censuses 2007 and 2017
- Level: Population center
- Variable: Average annual intercensal growth rate
- Application: Territorial analysis for district creations

Hierarchical Filtering: Filters are applied in cascade:

1. Specified departments are loaded first
2. Then provinces are filtered (if specified)
3. Finally, districts are filtered (if specified)

Cache is stored in: `tempdir()/DEMARCA_cache/centros_poblados/`

NOTE: Geometries are POINT type and represent the location of each population center.

Value

An `sf` (simple feature) object with POINT type geometry containing information on population centers and their growth rate, including:

- POINT geometry (coordinates of the population center)
- `nro`: Identification number
- `codccpp`: Population center code
- `ubigeo`: UBIGEO code of the district
- `dep`: Department name
- `prov`: Province name
- `distrito`: District name
- `centro_pob`: Population center name
- `pob_2007`: Population in the 2007 census
- `pob_2017`: Population in the 2017 census

- `tasa`: Average annual growth rate (%)
- `capital`: Capital type (Departmental, Provincial, Distrital, Not a capital)
- `region`: Natural region (Coast, Highland, Jungle)
- `urb_rural`: Urban/rural classification
- `tc_catg`: Growth rate category (POSITIVE, NEGATIVE)

If multiple departments are requested, it returns a combined sf object.

References

INEI. National Population and Housing Censuses 2007 and 2017.

DEMARCA OSF Repository: https://osf.io/qy4j6/

See Also

get_densidad_poblacional, get_departamentos, get_provincias, get_distritos, read_sf

Examples

```
## Not run:
# See available departments
get_centros_poblados_crecimiento()

# Load population centers of a full department
ccpp_cusco <- get_centros_poblados_crecimiento(departamento = "CUSCO")

# Filter by specific province
ccpp_prov_cusco <- get_centros_poblados_crecimiento(
  departamento = "CUSCO",
  provincia = "CUSCO"
)

# Filter by specific district
ccpp_wanchaq <- get_centros_poblados_crecimiento(
  departamento = "CUSCO",
  provincia = "CUSCO",
  distrito = "WANCHAQ"
)

# Load multiple departments
ccpp_sur <- get_centros_poblados_crecimiento(
  departamento = c("CUSCO", "PUNO", "AREQUIPA")
)

# Visualization with ggplot2
library(ggplot2)
library(dplyr)

# Map of population centers by growth category
ggplot(ccpp_cusco) +
  geom_sf(aes(color = tc_catg, size = pob_2017), alpha = 0.6) +
```

```
  scale_color_manual(
    values = c("POSITIVO" = "darkgreen", "NEGATIVO" = "darkred"),
    name = "Growth"
  ) +
  scale_size_continuous(
    name = "Population 2017",
    range = c(0.5, 5)
  ) +
  labs(
    title = "Population Centers of the Department of Cusco",
    subtitle = "Population Growth Rate 2007-2017",
    caption = "Source: INEI | Visor - SDOT"
  ) +
  theme_minimal()

# Filter population centers by characteristics
ccpp_urbanos <- ccpp_cusco |>
  filter(urb_rural == "URBANA")

ccpp_capitales <- ccpp_cusco |>
  filter(capital != "No es capital")

# Population centers with highest growth
top_crecimiento <- ccpp_cusco |>
  arrange(desc(tasa)) |>
  head(10)

ggplot(top_crecimiento) +
  geom_sf(aes(size = tasa, color = tasa)) +
  scale_color_gradient(low = "yellow", high = "red", name = "Rate (%)") +
  scale_size_continuous(range = c(3, 10), name = "Rate (%)") +
  geom_sf_text(
    aes(label = centro_pob),
    size = 2.5,
    nudge_y = 0.05
  ) +
  labs(
    title = "Top 10 Population Centers with Highest Growth",
    subtitle = "Department of Cusco (2007-2017)",
    caption = "Source: INEI | Visor - SDOT"
  ) +
  theme_minimal()

# Analysis by natural region
ccpp_cusco |>
  group_by(region, tc_catg) |>
  summarise(
    n_centros = n(),
    pob_total_2017 = sum(pob_2017, na.rm = TRUE),
    tasa_promedio = mean(tasa, na.rm = TRUE),
    .groups = "drop"
  )
```

```
# Urban vs rural comparison
ggplot(ccpp_cusco) +
  geom_boxplot(aes(x = urb_rural, y = tasa, fill = urb_rural)) +
  scale_fill_brewer(palette = "Set2") +
  labs(
    title = "Distribution of Growth Rates",
    subtitle = "By urban and rural area - Cusco",
    x = "Classification",
    y = "Growth Rate (%)",
    caption = "Source: INEI | Visor - SDOT"
  ) +
  theme_minimal() +
  theme(legend.position = "none")

## End(Not run)
```

`get_cobertura_movil_c` *Get Mobile Coverage by Population Center*

Description

Downloads and loads information on mobile coverage by population center from the DEMARCA OSF repository. Data includes binary information (YES/NO) on the existence of 2G, 3G, and 4G coverage for each population center according to the SDOT Viewer.

Usage

```
get_cobertura_movil_c(
  departamento = NULL,
  provincia = NULL,
  distrito = NULL,
  show_progress = TRUE,
  force_update = FALSE
)
```

Arguments

`departamento`	Character vector. Name(s) of the department(s) to download. Valid options: "AMAZONAS", "ANCASH", "APURIMAC", "AREQUIPA", "AYACUCHO", "CAJAMARCA", "CALLAO", "CUSCO", "HUANCAVELICA", "HUANUCO", "ICA", "JUNIN", "LA LIBERTAD", "LAMBAYEQUE", "LIMA", "LORETO", "MADRE DE DIOS", "MOQUEGUA", "PASCO", "PIURA", "PUNO", "SAN MARTIN", "TACNA", "TUMBES", "UCAYALI".
`provincia`	Character vector. Name(s) of the province(s) to filter within the specified department(s). Optional.
`distrito`	Character vector. Name(s) of the district(s) to filter within the specified province(s). Optional.

show_progress	Logical. If TRUE (default), shows informative messages about download progress. If FALSE, executes silently.
force_update	Logical. If TRUE, forces a new download of the file even if it exists in cache. Default FALSE.

Details

La función descarga datos desde OSF (Open Science Framework) y los almacena en caché durante la sesión de R. Los datos están en formato GeoPackage (.gpkg).

Fuente de los datos:

- Fuente: Ministerio de Transportes y Comunicaciones (MTC) / OSIPTEL
- Nivel: Centro poblado
- Aplicación: Análisis de brecha digital y planificación de telecomunicaciones

Interpretación de variables binarias:

Las variables de tecnología (2G, 3G, 4G, 5G), servicios (VOZ, SMS, MMS) y velocidad (hasta1mbps, masde1mbps) son declaradas por las empresas operadoras y toman únicamente valores de 0 o 1:

- **Valor 1**: El centro poblado cuenta con la tecnología/servicio
- **Valor 0**: El centro poblado NO cuenta con la tecnología/servicio

Tecnologías de red móvil:

- **2G (GSM)**: Segunda generación - servicios básicos de voz y SMS
- **3G (UMTS/HSPA)**: Tercera generación - datos móviles básicos, navegación web limitada
- **4G (LTE)**: Cuarta generación - banda ancha móvil, streaming de video, aplicaciones en tiempo real
- **5G**: Quinta generación - ultra banda ancha, baja latencia, IoT masivo

Estaciones base:

Las variables canteb2g, canteb3g, canteb4g y canteb5g indican la cantidad de estaciones base (antenas) que pueden brindar servicio de cobertura móvil con cada tecnología al centro poblado. Un mayor número de estaciones base generalmente indica mejor calidad de señal y capacidad de red.

Filtrado jerárquico: Los filtros se aplican en cascada:

1. Primero se cargan los departamentos especificados
2. Luego se filtran las provincias (si se especifican)
3. Finalmente se filtran los distritos (si se especifican)

El caché se almacena en: `tempdir()/DEMARCA_cache/cobertura_movil/`

NOTA: Las geometrías son tipo POINT (puntos) y representan la ubicación de cada centro poblado con información de cobertura.

Value

An `sf` (simple feature) object with POINT geometry containing mobile coverage information, including:

Population Center Identification:

- `ubigeo`: Ubigeo code of the population center
- `nom_dep`: Department name
- `nom_prov`: Province name
- `nom_dist`: District name
- `nomccpp`: Population center name

Mobile Network Technologies (values: 1 = available, 0 = not available):

- `2g`: 2G technology declared by the operator at population center level
- `3g`: 3G technology declared by the operator at population center level
- `4g`: 4G technology declared by the operator at population center level
- `5g`: 5G technology declared by the operator at population center level

Communication Services (values: 1 = available, 0 = not available):

- `voz`: Voice service provided by the operator at population center level
- `sms`: Short message service (SMS) provided by the operator at population center level
- `mms`: Multimedia message service (MMS) provided by the operator at population center level
- `internet`: Internet service provided by the operator at population center level

Other Indicators:

- `y_latitud`: Latitude of the population center
- `x_longitud`: Longitude of the population center
- `emoperador`: Operator presence indicator
- `layer`: Source layer

Velocidad de internet móvil (valores: 1 = disponible, 0 = no disponible):

- `hasta1mbps`: Servicio de acceso a internet móvil con velocidad de hasta 1 Mbps brindado por la empresa operadora
- `masde1mbps`: Servicio de acceso a internet móvil con velocidad de más de 1 Mbps brindado por la empresa operadora

Infraestructura - Estaciones base:

- `canteb2g`: Cantidad de estaciones base que pueden brindar servicio de cobertura móvil con tecnología 2G al centro poblado
- `canteb3g`: Cantidad de estaciones base que pueden brindar servicio de cobertura móvil con tecnología 3G al centro poblado
- `canteb4g`: Cantidad de estaciones base que pueden brindar servicio de cobertura móvil con tecnología 4G al centro poblado
- `canteb5g`: Cantidad de estaciones base que pueden brindar servicio de cobertura móvil con tecnología 5G al centro poblado

Si se solicitan múltiples departamentos, retorna un objeto sf combinado.

References

Ministerio de Transportes y Comunicaciones (MTC). Cobertura de Servicio Móvil por Centros Poblados.

OSIPTEL - Organismo Supervisor de Inversión Privada en Telecomunicaciones.

See Also

get_departamentos, get_provincias, get_distritos, get_centros_poblados, read_sf

Examples

```
## Not run:
# Ver departamentos disponibles
get_cobertura_movil_c()

# Cargar cobertura de un departamento completo
cob_cusco <- get_cobertura_movil_c(departamento = "CUSCO")

# Filtrar por provincia específica
cob_prov_cusco <- get_cobertura_movil_c(
  departamento = "CUSCO",
  provincia = "CUSCO"
)

# Filtrar por distrito específico
cob_san_sebastian <- get_cobertura_movil_c(
  departamento = "CUSCO",
  provincia = "CUSCO",
  distrito = "SAN SEBASTIAN"
)

# Cargar múltiples departamentos
cob_sur <- get_cobertura_movil_c(
  departamento = c("CUSCO", "PUNO", "AREQUIPA")
)

# Visualización con ggplot2
library(ggplot2)
library(dplyr)

# Mapa de cobertura 4G
ggplot(cob_cusco) +
  geom_sf(aes(color = factor(`4g`)), size = 1, alpha = 0.6) +
  scale_color_manual(
    values = c("0" = "red", "1" = "darkgreen"),
    labels = c("0" = "Sin cobertura", "1" = "Con cobertura"),
    name = "Cobertura 4G"
  ) +
  labs(
    title = "Cobertura 4G en Centros Poblados de Cusco",
    subtitle = "MTC / OSIPTEL",
```

```
    caption = "Fuente: MTC | Visor - SDOT"
  ) +
  theme_minimal()

# Resumen de cobertura por tecnología
cob_cusco |>
  sf::st_drop_geometry() |>
  summarise(
    total_ccpp = n(),
    con_2g = sum(`2g`),
    con_3g = sum(`3g`),
    con_4g = sum(`4g`),
    con_5g = sum(`5g`),
    pct_4g = round(sum(`4g`) / n() * 100, 1)
  )

## End(Not run)
```

`get_densidad_poblacional` *Get Population Density and Growth by Department*

Description

Downloads and loads intercensal population density and growth information by department from the DEMARCA OSF repository. Data comes from INEI 2007 and 2017 population censuses, processed through spatial interpolation. Allows stepwise filtering by department, province, and district.

Usage

```
get_densidad_poblacional(
  departamento = NULL,
  provincia = NULL,
  distrito = NULL,
  show_progress = TRUE,
  force_update = FALSE
)
```

Arguments

`departamento`	Character vector. Name(s) of the department(s) to download. Valid options: "AMAZONAS", "ANCASH", "APURIMAC", "AREQUIPA", "AYACUCHO", "CAJAMARCA", "CALLAO", "CUSCO", "HUANCAVELICA", "HUANUCO", "ICA", "JUNIN", "LA LIBERTAD", "LAMBAYEQUE", "LIMA", "LORETO", "MADRE DE DIOS", "MOQUEGUA", "PASCO", "PIURA", "PUNO", "SAN MARTIN", "TACNA", "TUMBES", "UCAYALI". Case-insensitive. If `NULL`, shows the list of available departments.

provincia	Character vector. Name(s) of the province(s) to filter within the specified department(s). Optional.
distrito	Character vector. Name(s) of the district(s) to filter within the specified province(s). Optional.
show_progress	Logical. If TRUE (default), shows informative messages about download progress. If FALSE, executes silently.
force_update	Logical. If TRUE, forces a new download of the file even if it exists in cache. Default FALSE.

Details

The function downloads data from OSF (Open Science Framework) and stores it in cache during the R session. Data is in GeoPackage (.gpkg) format.

Data Methodology:

- Source: INEI 2007 and 2017 Population Censuses
- Coverage: Population centers with > 150 inhabitants
- Technique: Spatial interpolation for continuous representation
- Variable: Annual intercensal population growth rate

Hierarchical Filtering: Filters are applied in cascade:

1. Specified departments are loaded first
2. Then provinces are filtered (if specified)
3. Finally, districts are filtered (if specified)

Cache is stored in: `tempdir()/DEMARCA_cache/densidad_poblacional/`

NOTE: Geometries are automatically validated with `st_make_valid()` to prevent errors in subsequent spatial operations.

Value

An `sf` (simple feature) object with density and population growth information, including:

- Geometry of interpolated polygons (with validated geometries)
- `ubigeo`: District UBIGEO code
- `nombdep`: Department name
- `nombprov`: Province name
- `nombdist`: District name
- `nivel`: Population growth level
- `rango`: Growth rate range
- `descrip`: Category description

If multiple departments are requested, it returns a combined sf object.

References

INEI. National Population and Housing Censuses 2007 and 2017.

DEMARCA OSF Repository: https://osf.io/qy4j6/

See Also

get_departamentos, get_provincias, get_distritos, read_sf, st_make_valid

Examples

```
## Not run:
# See available departments
get_densidad_poblacional()

# Load a full department
densidad_cusco <- get_densidad_poblacional(departamento = "CUSCO")

# Load a specific province
densidad_prov_cusco <- get_densidad_poblacional(
  departamento = "CUSCO",
  provincia = "CUSCO"
)

# Load a specific district
densidad_wanchaq <- get_densidad_poblacional(
  departamento = "CUSCO",
  provincia = "CUSCO",
  distrito = "WANCHAQ"
)

# Load multiple departments
densidad_sur <- get_densidad_poblacional(
  departamento = c("CUSCO", "PUNO", "AREQUIPA")
)

# Filter multiple provinces
densidad_valle <- get_densidad_poblacional(
  departamento = "CUSCO",
  provincia = c("CUSCO", "URUBAMBA", "CALCA")
)

# Visualization with ggplot2
library(ggplot2)
library(dplyr)

# Growth level map
ggplot(densidad_cusco) +
  geom_sf(aes(fill = nivel), color = NA) +
  scale_fill_brewer(
    palette = "RdYlGn",
    name = "Growth\nLevel"
  ) +
```

```
  labs(
    title = "Population Growth by Level",
    subtitle = "Department of Cusco (2007-2017)",
    caption = "Source: INEI | Visor - SDOT"
  ) +
  theme_minimal()

# Aggregated layer by district (one polygon per district)
distritos_anta <- prov_anta |>
  group_by(ubigeo, nombdist) |>
  summarise(.groups = "drop")
distritos_anta

# Points to place labels inside each district
distritos_centroides <- st_point_on_surface(distritos_anta)
distritos_centroides

ggplot() +
  # 1. Cells / polygons with growth level
  geom_sf(
    data = prov_anta,
    aes(fill = nivel),
    color = NA
  ) +
  # 2. District outline (to make them stand out)
  geom_sf(
    data = distritos_anta,
    fill = NA,
    color = "grey20",
    linewidth = 0.4
  ) +
  # 3. District names
  geom_sf_text(
    data = distritos_centroides,
    aes(label = nombdist),
    size = 3
  ) +
  scale_fill_brewer(
    palette = "RdYlGn",
    name    = "Growth\nlevel"
  ) +
  labs(
    title    = "Population Growth",
    subtitle = "Province of Anta (2007-2017)",
    caption  = "Source: INEI | Visor SDOT"
  ) +
  theme_minimal() +
  theme(
    legend.position = "right",
    panel.grid.major = element_line(linewidth = 0.2, colour = "grey90")
  )

## End(Not run)
```

`get_departamentos` *Get Departmental Census Limits of Peru*

Description

Downloads and loads the official departmental limits from INEI (2023) from the DEMARCA OSF repository. Implements persistent caching to optimize repeated downloads and keeps the console clean during execution.

Usage

```
get_departamentos(
  departamento = NULL,
  show_progress = TRUE,
  force_update = FALSE
)
```

Arguments

`departamento`	Character vector. Name(s) of the department(s) to filter. Can be a single department (e.g., "CUSCO") or multiple departments (e.g., c("CUSCO", "PUNO")). Case-insensitive. If `NULL` (default), returns all departments of Peru.
`show_progress`	Logical. If `TRUE` (default), shows informative messages about download progress. If `FALSE`, executes silently.
`force_update`	Logical. If `TRUE`, forces a new download of the file even if it exists in cache. Default `FALSE` to leverage cached data.

Details

The function downloads data directly from OSF (Open Science Framework) and stores it in the user's cache directory (`tools::R_user_dir("rsdot", "cache")`). The data corresponds to the official INEI 2023 census limits.

Data source: DEMARCA Repository on OSF (https://osf.io/grhe3/)

Value

An `sf` (simple feature) object with the geometry of the requested departments, including their geographic and administrative attributes.

References

INEI (2023). Departmental Census Limits of Peru. SDOT DEMARCA PERU VISOR Repository: https://osf.io/qy4j6/overview

See Also

read_sf, st_geometry

Examples

```
## Not run:
# Load all departments of Peru
peru <- get_departamentos()

# Load a specific department (case-insensitive)
cusco <- get_departamentos(departamento = "Cusco")

# Load multiple departments (southern region)
sur <- get_departamentos(
  departamento = c("PUNO", "TACNA", "MOQUEGUA", "AREQUIPA")
)

# Force data update
peru_updated <- get_departamentos(force_update = TRUE)

# Silent execution
lima <- get_departamentos(departamento = "LIMA", show_progress = FALSE)

# Visualization with ggplot2
library(ggplot2)

ggplot(peru) +
  geom_sf(fill = "#ff9999", color = "#e60000", linewidth = 1) +
  labs(
    title = "Departmental Census Limits of Peru",
    subtitle = "INEI 2023 Census Limits",
    caption = "Source: INEI | Visor - SDOT"
  ) +
  theme_minimal()

## End(Not run)
```

---

get_distritos *Get District Census Limits of Peru*

Description

Downloads and loads the official district limits from INEI (2023) from the DEMARCA OSF repository. Implements persistent caching to optimize repeated downloads and keeps the console clean during execution.

Usage

```
get_distritos(
  distrito = NULL,
  provincia = NULL,
  departamento = NULL,
  show_progress = TRUE,
  force_update = FALSE
)
```

Arguments

distrito	Character vector. Name(s) of the district(s) to filter. Can be a single district (e.g., "WANCHAQ") or multiple districts (e.g., c("CUSCO", "WANCHAQ", "SANTIAGO")). Case-insensitive. If NULL, no filtering is performed by district.
provincia	Character vector. Name of the province to filter all its districts (e.g., "CUSCO"). If specified, it returns all the districts of the indicated province. Case-insensitive.
departamento	Character vector. Name of the department to filter all its districts (e.g., "CUSCO"). If specified, it returns all the districts of the indicated department. Case-insensitive.
show_progress	Logical. If TRUE (default), shows informative messages about download progress. If FALSE, executes silently.
force_update	Logical. If TRUE, forces a new download of the file even if it exists in cache. Default FALSE to leverage cached data.

Details

The function downloads data directly from OSF (Open Science Framework) and stores it in the user's cache directory (`tools::R_user_dir("rsdot", "cache")`). The data corresponds to the official INEI 2023 census limits.

The function automatically handles special character encoding (accents, ñ) to ensure compatibility across different operating systems.

Filters can be combined hierarchically:

- Department only: returns all districts in the department
- Department + province: returns all districts in that province
- Department + province + district: returns specific districts
- District only: returns that district from anywhere in Peru

Data source: DEMARCA Repository on OSF (https://osf.io/j9ax8/)

Value

An sf (simple feature) object with the geometry of the requested districts, including their geographic and administrative attributes.

References

INEI (2023). District Census Limits of Peru. DEMARCA Repository: https://osf.io/j9ax8/

See Also

get_departamentos, get_provincias, read_sf, st_geometry

Examples

```
## Not run:
# Load all districts of Peru
distritos_peru <- get_distritos()

# Load a specific district
wanchaq <- get_distritos(distrito = "Wanchaq")

# Load all districts of a province
distritos_cusco <- get_distritos(provincia = "Cusco")

# Load all districts of a department
distritos_puno <- get_distritos(departamento = "Puno")

# Load multiple specific districts
distritos_centrales <- get_distritos(
  distrito = c("CUSCO", "WANCHAQ", "SANTIAGO")
)

# Combine filters: districts of a specific province
costa_lima <- get_distritos(
  provincia = "CAÑETE",
  departamento = "LIMA"
)

# Full hierarchical filter
distrito_especifico <- get_distritos(
  distrito = "SAN SEBASTIÁN",
  provincia = "CUSCO",
  departamento = "CUSCO"
)

# Force data update
distritos_updated <- get_distritos(force_update = TRUE)

# Silent execution
callao <- get_distritos(provincia = "CALLAO", show_progress = FALSE)

# Visualization with ggplot2
library(ggplot2)

# Map of all districts in the province of Cusco
ggplot(distritos_cusco) +
  geom_sf(aes(fill = nombdist), color = "white", linewidth = 0.2) +
  labs(
    title = "Districts of the Province of Cusco",
    subtitle = "INEI 2023 Census Limits",
    fill = "District",
```

```
    caption = "Source: INEI | Visor - SDOT"
  ) +
  scale_fill_brewer(palette = "Paired") +
  theme_minimal() +
  theme(
    legend.position = "right",
    legend.key.size = unit(0.4, "cm")
  )

# Map of a specific district
ggplot(wanchaq) +
  geom_sf(fill = "#FFA07A", color = "darkred", linewidth = 1.5) +
  labs(
    title = "District of Wanchaq",
    subtitle = "Province and Department of Cusco",
    caption = "Source: INEI | Visor - SDOT"
  ) +
  theme_minimal()

# Map of districts in Cañete
ggplot(costa_lima) +
  geom_sf(aes(fill = nombdist), color = "white", linewidth = 0.3) +
  labs(
    title = "Districts of the Province of Cañete",
    fill = "District",
    caption = "Source: INEI | Visor - SDOT"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")

# Map of central districts with labels
ggplot(distritos_centrales) +
  geom_sf(aes(fill = nombdist), color = "white", linewidth = 0.5) +
  geom_sf_text(aes(label = nombdist), size = 3, fontface = "bold") +
  labs(
    title = "Central Districts of Cusco",
    fill = "District",
    caption = "Source: INEI | Visor - SDOT"
  ) +
  scale_fill_manual(
    values = c(
      "CUSCO" = "#FF6B6B", "WANCHAQ" = "#4ECDC4",
      "SANTIAGO" = "#45B7D1"
    )
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")

## End(Not run)
```

get_ipress *Get Health Service Provider Institutions (IPRESS) of Peru*

Description

Downloads and loads information on Health Service Provider Institutions (IPRESS) of Peru, prepared by the National Health Superintendence (SUSALUD). Data includes POINT type geometry and information on health establishments, supporting medical services, categorization, operating hours, and health networks. Updated to 2024. Allows stepwise filtering by department, province, and district.

Usage

```
get_ipress(
  departamento = NULL,
  provincia = NULL,
  distrito = NULL,
  show_progress = TRUE,
  force_update = FALSE
)
```

Arguments

departamento	Character vector. Name(s) of the department(s) to download. Valid options: "AMAZONAS", "ANCASH", "APURIMAC", "AREQUIPA", "AYACUCHO", "CAJAMARCA", "CALLAO", "CUSCO", "HUANCAVELICA", "HUANUCO", "ICA", "JUNIN", "LA LIBERTAD", "LAMBAYEQUE", "LIMA", "LORETO", "MADRE DE DIOS", "MOQUEGUA", "PASCO", "PIURA", "PUNO", "SAN MARTIN", "TACNA", "TUMBES", "UCAYALI". Case-insensitive. If NULL, shows the list of available departments.
provincia	Character vector. Name(s) of the province(s) to filter within the specified department(s). Optional.
distrito	Character vector. Name(s) of the district(s) to filter within the specified province(s). Optional.
show_progress	Logical. If TRUE (default), shows informative messages about download progress. If FALSE, executes silently.
force_update	Logical. If TRUE, forces a new download of the file even if it exists in cache. Default FALSE.

Details

The function downloads data from OSF (Open Science Framework) and stores it in cache during the R session. Data is in GeoPackage (.gpkg) format.

Data Source:

- Source: National Health Superintendence (SUSALUD)

- Registration: RENIPRESS (National IPRESS Registry)
- Update year: 2024
- Level: Health establishment
- Application: Analysis of access to health services, health planning, and epidemiological studies

Health Establishment Categorization System in Peru:

Categorization determines the resolutive capacity of the establishment:

- **First level (I-1 to I-4)**: Basic and preventive care
  - I-1: Health post (nursing technician)
  - I-2: Health post (health professional)
  - I-3: Health center without inpatient care
  - I-4: Health center with inpatient care
- **Second level (II-1, II-2, II-E)**: Specialized care
  - II-1: Hospital I
  - II-2: Hospital II
  - II-E: Specialized hospital
- **Third level (III-1, III-2, III-E)**: High specialization
  - III-1: Hospital III
  - III-2: Specialized institute
  - III-E: High complexity specialized institute

IPRESS Types:

1. **Establishments without inpatient care**: Offices, health posts, medical centers, polyclinics
2. **Establishments with inpatient care**: Hospitals, clinics, health centers with beds
3. **Supporting medical services**: Laboratories, image diagnostic centers, blood banks, clinical pathology

Hierarchical Filtering: Filters are applied in cascade:

1. Specified departments are loaded first
2. Then provinces are filtered (if specified)
3. Finally districts are filtered (if specified)

Cache is stored in: `tempdir()/DEMARCA_cache/ipress/`

NOTE: Geometries are POINT type and represent the location of each health establishment.

Value

An `sf` (simple feature) object with POINT geometry containing IPRESS information, including:

Establishment Identification:

- POINT geometry (establishment coordinates)
- `cod_unic_e`: Unique establishment code (RENIPRESS)

- nom_esta: Health establishment name
- dir_esta: Full address of the establishment
- num_tele: Contact phone number

Type and Classification:

- cod_tipo_e: Establishment type code
- des_tipo_e: Establishment type description:
  - 1 = Health establishment without inpatient care
  - 2 = Health establishment with inpatient care
  - 3 = Supporting medical service
  - 4 = Flexible offer
  - 5 = Therapeutic communities
- cod_clas_e: Establishment classification code
- des_clas_e: Classification description (e.g., Health posts, Health centers, Hospitals, Clinics, Doctor's offices, etc.)

Administering Institution:

- cod_inst_d: Administering institution code
- des_inst_d: Administering institution description

Geographic Location:

- cod_ubig_e: Establishment ubigeo code
- nombdep: Department name
- nombprov: Province name
- nombdist: District name
- val_lati: Establishment latitude
- val_long: Establishment longitude

Categorization and Operation:

- cod_cate: Establishment category code (e.g., I-1, I-2, I-3, I-4, II-1, II-2, II-E, III-1, III-2, III-E)
- doc_cate: Categorization resolution document
- fec_inic_o: Operations start date
- des_hora_e: Establishment operating hours
- cod_cond_e: Establishment condition (e.g., ACTIVE, INACTIVE)

Health Network:

- cod_auto_s: Health authority code
- des_auto_s: Health authority description (region)
- cod_reds: Health network code
- des_reds: Health network name

- cod_micr_r: Micro-network code
- des_micr_r: Micro-network name

Representative Data:

- cod_nruc: Institution RUC number
- nom_repr_l: Legal representative name

If multiple departments are requested, it returns a combined sf object.

References

National Health Superintendence (SUSALUD). National Registry of Health Service Provider Institutions (RENIPRESS) 2024.

Ministry of Health of Peru. Technical Standard for Categorization of Health Establishments.

DEMARCA OSF Repository: https://osf.io/qy4j6/

See Also

get_departamentos, get_provincias, get_distritos, get_centros_poblados, read_sf

Examples

```
## Not run:
# See available departments
get_ipress()

# Load IPRESS of a full department
ipress_cusco <- get_ipress(departamento = "CUSCO")

# Filter by specific province
ipress_prov_cusco <- get_ipress(
  departamento = "CUSCO",
  provincia = "CUSCO"
)

# Filter by specific district
ipress_wanchaq <- get_ipress(
  departamento = "CUSCO",
  provincia = "CUSCO",
  distrito = "WANCHAQ"
)

# Load multiple departments
ipress_sur <- get_ipress(
  departamento = c("CUSCO", "PUNO", "AREQUIPA")
)

# Visualization with ggplot2
library(ggplot2)
library(dplyr)
```

```
# Map of IPRESS by establishment type
ggplot(ipress_cusco) +
  geom_sf(aes(color = des_tipo_e), size = 1.5, alpha = 0.7) +
  scale_color_brewer(palette = "Set1", name = "Type") +
  labs(
    title = "Health Service Provider Institutions - Cusco",
    subtitle = "SUSALUD 2024",
    caption = "Source: SUSALUD - RENIPRESS"
  ) +
  theme_minimal()

# Summary by establishment type
ipress_cusco |>
  sf::st_drop_geometry() |>
  count(des_tipo_e, sort = TRUE)

# Summary by administering institution
ipress_cusco |>
  sf::st_drop_geometry() |>
  count(des_inst_d, sort = TRUE)

# Filter active public establishments
ipress_publicos <- ipress_cusco |>
  filter(
    des_inst_d == "GOBIERNO REGIONAL",
    cod_cond_e == "ACTIVO"
  )

# Analysis by category
ipress_cusco |>
  sf::st_drop_geometry() |>
  count(cod_cate, sort = TRUE) |>
  head(10)

# Map of hospitals and clinics (establishments with inpatient care)
ipress_cusco |>
  filter(cod_tipo_e == "2") |>
  ggplot() +
  geom_sf(aes(color = des_inst_d), size = 3) +
  scale_color_brewer(palette = "Dark2", name = "Administrator") +
  labs(
    title = "Health Establishments with Inpatient Care - Cusco",
    subtitle = "Hospitals and Clinics",
    caption = "Source: SUSALUD - RENIPRESS 2024"
  ) +
  theme_minimal()

# Coverage analysis by district
cobertura_distrito <- ipress_cusco |>
  sf::st_drop_geometry() |>
  group_by(nombdist) |>
  summarise(
```

```
    total_ipress = n(),
    con_internamiento = sum(cod_tipo_e == "2"),
    sin_internamiento = sum(cod_tipo_e == "1"),
    supporting_services = sum(cod_tipo_e == "3"),
    .groups = "drop"
  ) |>
  arrange(desc(total_ipress))

# Distribution by health network
ipress_cusco |>
  sf::st_drop_geometry() |>
  filter(!is.na(des_reds)) |>
  count(des_reds, sort = TRUE)

# Map of establishments by category (first level)
ipress_cusco |>
  filter(grepl("^I-", cod_cate)) |>
  ggplot() +
  geom_sf(aes(color = cod_cate), size = 2, alpha = 0.7) +
  scale_color_viridis_d(name = "Category") +
  labs(
    title = "First Level Establishments - Cusco",
    subtitle = "Categories I-1 to I-4",
    caption = "Source: SUSALUD - RENIPRESS 2024"
  ) +
  theme_minimal()

# Search establishments by name
regional_hospitals <- ipress_cusco |>
  filter(grepl("REGIONAL|HOSPITAL", nom_esta, ignore.case = TRUE))

## End(Not run)
```

`get_locales_educativos`

Get Educational Facilities of Peru

Description

Downloads and loads information on educational facilities in Peru, prepared by the Ministry of Education (MINEDU). An educational facility is a property where one or more educational establishments (educational services) operate. Data includes POINT type geometry and information on educational services operating at each facility, location, and census area. Allows stepwise filtering by department, province, and district.

Usage

```
get_locales_educativos(
  departamento = NULL,
```

```
  provincia = NULL,
  distrito = NULL,
  show_progress = TRUE,
  force_update = FALSE
)
```

Arguments

departamento	Character vector. Name(s) of the department(s) to download. Valid options: "AMAZONAS", "ANCASH", "APURIMAC", "AREQUIPA", "AYACUCHO", "CAJAMARCA", "CALLAO", "CUSCO", "HUANCAVELICA", "HUANUCO", "ICA", "JUNIN", "LA LIBERTAD", "LAMBAYEQUE", "LIMA", "LORETO", "MADRE DE DIOS", "MOQUEGUA", "PASCO", "PIURA", "PUNO", "SAN MARTIN", "TACNA", "TUMBES", "UCAYALI". Case-insensitive. If NULL, shows the list of available departments.
provincia	Character vector. Name(s) of the province(s) to filter within the specified department(s). Optional.
distrito	Character vector. Name(s) of the district(s) to filter within the specified province(s). Optional.
show_progress	Logical. If TRUE (default), shows informative messages about download progress. If FALSE, executes silently.
force_update	Logical. If TRUE, forces a new download of the file even if it exists in cache. Default FALSE.

Details

The function downloads data from OSF (Open Science Framework) and stores it in cache during the R session. Data is in GeoPackage (.gpkg) format.

Data Source:

- Source: Ministry of Education (MINEDU)
- Registration: Educational Institutions Registry
- Level: Educational facility
- Application: Education coverage analysis, school infrastructure planning, and accessibility studies

Key Concepts:

Educational Facility vs Educational Institution:

- **Educational Facility**: Physical infrastructure (property) where educational services operate. A facility may house multiple educational institutions.
- **Educational Service/Institution**: Operating unit of the educational system that provides a specific type of education (initial, primary, secondary, CETPRO, etc.)

Educational Levels and Modalities in Peru:

- **Regular Basic Education (EBR)**:

  - Initial (0-5 years): Nursery, Kindergarten
  - Primary (6-11 years)
  - Secondary (12-16 years)
- **Alternative Basic Education (EBA)**: For persons who did not timely access EBR
- **Special Basic Education (EBE)**: For persons with disabilities or gifted talent
- **Technical-Productive Education (CETPRO)**: Technical training and vocational training
- **Higher Education**: Institutes and schools of pedagogical, technological, and artistic higher education

Interpretation of the servicios field:

The `servicios` field contains a text string with all educational services operating in the facility, separated by commas. Each service includes the name of the institution and its level/modality in parentheses.

Example: "501330 (PRIMARIA), 501330 (SECUNDARIA)" indicates that primary and secondary levels of IE 501330 operate in that facility.

Hierarchical Filtering: Filters are applied in cascade:

1. Specified departments are loaded first
2. Then provinces are filtered (if specified)
3. Finally districts are filtered (if specified)

Cache is stored in: `tempdir()/DEMARCA_cache/locales_educativos/`

NOTE: Geometries are POINT type and represent the location of each educational facility.

Value

An `sf` (simple feature) object with POINT geometry containing educational facility information, including:

Facility Identification:

- POINT geometry (facility coordinates)
- `codlocal`: Unique educational facility code assigned by MINEDU
- `servicios`: List of educational services (educational institutions) operating in the facility, including name, level, and modality. A single facility may house multiple educational services (e.g., initial, primary, and secondary)
- `dir_cen`: Educational facility address

Geographic Location:

- `localidad`: Name of the locality where the facility is located
- `codcp_inei`: Population center code according to INEI
- `codccpp`: Population center code (alternative)
- `centro_pob`: Population center name
- `departamen`: Department name
- `provincia`: Province name

- `distrito`: District name

Facility Characteristics:

- `area_censo`: Census area where the facility is located:
  - URBANA: Urban zone
  - RURAL: Rural zone

Coordinates:

- `nlat_ie`: Educational facility latitude (Y coordinate)
- `nlong_ie`: Educational facility longitude (X coordinate)

If multiple departments are requested, it returns a combined sf object.

References

Ministry of Education of Peru (MINEDU). Educational Institutions and Educational Programs Registry.

ESCALE - Statistics of Educational Quality: https://escale.minedu.gob.pe/

DEMARCA OSF Repository: https://osf.io/qy4j6/

See Also

get_departamentos, get_provincias, get_distritos, get_centros_poblados, read_sf

Examples

```
## Not run:
# See available departments
get_locales_educativos()

# Load educational facilities of a full department
locales_cusco <- get_locales_educativos(departamento = "CUSCO")

# Filter by specific province
locales_prov_cusco <- get_locales_educativos(
  departamento = "CUSCO",
  provincia = "CUSCO"
)

# Filter by specific district
locales_wanchaq <- get_locales_educativos(
  departamento = "CUSCO",
  provincia = "CUSCO",
  distrito = "WANCHAQ"
)

# Load multiple departments
locales_sur <- get_locales_educativos(
  departamento = c("CUSCO", "PUNO", "AREQUIPA")
```

```
)

# Visualization with ggplot2
library(ggplot2)
library(dplyr)

# Map of educational facilities by census area
ggplot(locales_cusco) +
  geom_sf(aes(color = area_censo), size = 1, alpha = 0.6) +
  scale_color_manual(
    values = c("URBANA" = "darkblue", "RURAL" = "darkgreen"),
    name = "Area"
  ) +
  labs(
    title = "Educational Facilities of the Department of Cusco",
    subtitle = "Ministry of Education",
    caption = "Source: MINEDU - Educational Institutions Registry"
  ) +
  theme_minimal()

# Summary by census area
locales_cusco |>
  sf::st_drop_geometry() |>
  count(area_censo)

# Distribution by district
locales_cusco |>
  sf::st_drop_geometry() |>
  count(distrito, sort = TRUE) |>
  head(10)

# Filter facilities with initial level
initial_locales <- locales_cusco |>
  filter(grepl("INICIAL", servicios, ignore.case = TRUE))

# Filter facilities with secondary
secondary_locales <- locales_cusco |>
  filter(grepl("SECUNDARIA", servicios, ignore.case = TRUE))

# Facilities with multiple levels (initial + primary + secondary)
complete_locales <- locales_cusco |>
  filter(
    grepl("INICIAL", servicios) &
      grepl("PRIMARIA", servicios) &
      grepl("SECUNDARIA", servicios)
  )

# Coverage analysis by province
province_coverage <- locales_cusco |>
  sf::st_drop_geometry() |>
  group_by(provincia) |>
  summarise(
    total_locales = n(),
```

```
    urban_locales = sum(area_censo == "URBANA"),
    rural_locales = sum(area_censo == "RURAL"),
    pct_rural = round(sum(area_censo == "RURAL") / n() * 100, 1),
    .groups = "drop"
  ) |>
  arrange(desc(total_locales))

# Map of rural facilities
locales_cusco |>
  filter(area_censo == "RURAL") |>
  ggplot() +
  geom_sf(color = "darkgreen", size = 0.8, alpha = 0.5) +
  labs(
    title = "Rural Educational Facilities - Cusco",
    subtitle = "MINEDU",
    caption = "Source: MINEDU - Educational Institutions Registry"
  ) +
  theme_minimal()

# Count educational services per facility
locales_cusco |>
  sf::st_drop_geometry() |>
  mutate(
    n_servicios = stringr::str_count(servicios, "\\(")
  ) |>
  count(n_servicios, name = "n_locales")

# Search facilities by institution name
emblematic_locales <- locales_cusco |>
  filter(grepl("INCA GARCILASO|CLORINDA MATTO", servicios, ignore.case = TRUE))

## End(Not run)
```

`get_provincias` *Get Provincial Census Limits of Peru*

Description

Downloads and loads the official provincial limits from INEI (2023) from the DEMARCA OSF repository. Implements persistent caching to optimize repeated downloads and keeps the console clean during execution.

Usage

```
get_provincias(
  provincia = NULL,
  departamento = NULL,
  show_progress = TRUE,
  force_update = FALSE
)
```

Arguments

provincia	Character vector. Name(s) of the province(s) to filter. Can be a single province (e.g., "URUBAMBA") or multiple provinces (e.g., c("CAÑETE", "CHINCHA")). Case-insensitive. If NULL, no filtering is performed by province.
departamento	Character vector. Name of the department to filter all its provinces (e.g., "CUSCO"). If specified, it returns all the provinces of the indicated department. Case-insensitive.
show_progress	Logical. If TRUE (default), shows informative messages about download progress. If FALSE, executes silently.
force_update	Logical. If TRUE, forces a new download of the file even if it exists in cache. Default FALSE to leverage cached data.

Details

The function downloads data directly from OSF (Open Science Framework) and stores it in the user's cache directory (`tools::R_user_dir("rsdot", "cache")`). The data corresponds to the official INEI 2023 census limits.

The function automatically corrects character encoding issues (especially Ñ and accents) in the original data.

Data source: DEMARCA Repository on OSF (https://osf.io/t36aj/)

Value

An sf (simple feature) object with the geometry of the requested provinces, including their geographic and administrative attributes.

References

INEI (2023). Provincial Census Limits of Peru. DEMARCA Repository: https://osf.io/t36aj/

See Also

get_departamentos, read_sf, st_geometry

Examples

```
## Not run:
# Load all provinces of Peru
provincias_peru <- get_provincias()

# Load a specific province
urubamba <- get_provincias(provincia = "Urubamba")

# Load all provinces of a department
provincias_cusco <- get_provincias(departamento = "Cusco")

# Load multiple specific provinces
```

```
sur_chico <- get_provincias(
  provincia = c("CAÑETE", "CHINCHA", "PISCO")
)

# Combine filters: specific provinces of a department
costa_lima <- get_provincias(
  provincia = c("CAÑETE", "YAUYOS"),
  departamento = "LIMA"
)

# Force data update
provincias_updated <- get_provincias(force_update = TRUE)

# Silent execution
pasco <- get_provincias(departamento = "PASCO", show_progress = FALSE)

# Visualization with ggplot2
library(ggplot2)

# Map of all provinces in Cusco
ggplot(provincias_cusco) +
  geom_sf(aes(fill = nombprov), color = "white", linewidth = 0.3) +
  labs(
    title = "Provinces of the Department of Cusco",
    subtitle = "INEI 2023 Census Limits",
    fill = "Province",
    caption = "Source: INEI | Visor - SDOT"
  ) +
  scale_fill_viridis_d(option = "plasma") +
  theme_minimal() +
  theme(legend.position = "right")

## End(Not run)
```

`get_red_vial_departamental` *Get Departmental Road Network of Peru*

Description

Downloads and loads information on the Departmental Road Network updated by PROVIAS NACIONAL as of July 2022, according to the Route Classifier approved by Supreme Decree 011-2016-MTC and its amendments. Data includes MULTILINESTRING type geometry and information on trajectories, status, and surface of departmental roads. Allows stepwise filtering by department and province.

Usage

```
get_red_vial_departamental(
```

```
  departamento = NULL,
  provincia = NULL,
  show_progress = TRUE,
  force_update = FALSE
)
```

Arguments

departamento	Character vector. Name(s) of the department(s) to download. Valid options: "AMAZONAS", "ANCASH", "APURIMAC", "AREQUIPA", "AYACUCHO", "CAJAMARCA", "CALLAO", "CUSCO", "HUANCAVELICA", "HUANUCO", "ICA", "JUNIN", "LA LIBERTAD", "LAMBAYEQUE", "LIMA", "LORETO", "MADRE DE DIOS", "MOQUEGUA", "PASCO", "PIURA", "PUNO", "SAN MARTIN", "TACNA", "TUMBES", "UCAYALI". Case-insensitive. If NULL, shows the list of available departments.
provincia	Character vector. Name(s) of the province(s) to filter within the specified department(s). Optional.
show_progress	Logical. If TRUE (default), shows informative messages about download progress. If FALSE, executes silently.
force_update	Logical. If TRUE, forces a new download of the file even if it exists in cache. Default FALSE.

Details

The function downloads data from OSF (Open Science Framework) and stores it in cache during the R session. Data is in GeoPackage (.gpkg) format.

Data Source:

- Source: PROVIAS NACIONAL - Ministry of Transport and Communications
- Update: July 2022
- Legal basis: Supreme Decree 011-2016-MTC and amendments
- Level: Departmental Road Network
- Application: Road planning and territorial connectivity analysis

Surface Classification:

- 1 = PAVED
- 2 = UNPAVED
- 3 = DIRT
- 4 = TRACK

Status Classification:

- 1 = GOOD
- 2 = REGULAR
- 3 = BAD

- 4 = VERY BAD

Hierarchical Filtering: Filters are applied in cascade:

1. Specified departments are loaded first
2. Then provinces are filtered (if specified)

Cache is stored in: `tempdir()/DEMARCA_cache/red_vial_departamental/`

NOTE: Geometries are MULTILINESTRING type and represent the trajectories of departmental roads.

Value

An `sf` (simple feature) object with MULTILINESTRING geometry containing departmental road network information, including:

- MULTILINESTRING geometry (road trajectory)
- `nombdep`: Department name
- `nombprov`: Province name
- `trayectori`: Description of the route trajectory
- `iddpto`: Department code
- `jerarq`: Route hierarchy (RD = Red Departamental)
- `estado`: Numeric road status (1, 2, 3, 4)
- `codruta`: Route code (e.g., PI-102, CU-105)
- `superfic`: Surface type code (1-4)
- `longitud`: Segment length in kilometers
- `estado_l`: Descriptive status (GOOD, REGULAR, BAD)
- `superfic_l`: Surface type (PAVED, UNPAVED, DIRT, TRACK)

If multiple departments are requested, it returns a combined sf object.

References

PROVIAS NACIONAL - MTC. Base Cartográfica de la Red Vial Nacional 2022.

Supreme Decree 011-2016-MTC - Route Classifier of the National Road System (SINAC).

DEMARCA OSF Repository: https://osf.io/qy4j6/

See Also

`get_departamentos`, `get_provincias`, `read_sf`

Examples

```
## Not run:
# See available departments
get_red_vial_departamental()

# Load road network of a full department
vias_cusco <- get_red_vial_departamental(departamento = "CUSCO")

# Filter by specific province
vias_cusco_prov <- get_red_vial_departamental(
  departamento = "CUSCO",
  provincia = "CUSCO"
)

# Load multiple departments
vias_sur <- get_red_vial_departamental(
  departamento = c("CUSCO", "PUNO", "AREQUIPA")
)

# Visualization with ggplot2
library(ggplot2)
library(dplyr)

# Map of road network by surface type
ggplot(vias_cusco) +
  geom_sf(aes(color = superfic_l), linewidth = 0.8) +
  scale_color_manual(
    values = c(
      "PAVIMENTADO" = "darkblue",
      "AFIRMADO" = "darkgreen",
      "SIN AFIRMAR" = "orange",
      "TROCHA" = "brown"
    ),
    name = "Surface Type"
  ) +
  labs(
    title = "Departmental Road Network of Cusco",
    subtitle = "By Surface Type - PROVIAS 2022",
    caption = "Source: PROVIAS NACIONAL | Visor - SDOT"
  ) +
  theme_minimal()

# Map by conservation status
ggplot(vias_cusco) +
  geom_sf(aes(color = estado_l), linewidth = 1) +
  scale_color_manual(
    values = c(
      "BUENO" = "darkgreen",
      "REGULAR" = "orange",
      "MALO" = "red"
    ),
    name = "Status"
```

```
  ) +
  labs(
    title = "Conservation Status of the Road Network",
    subtitle = "Department of Cusco",
    caption = "Source: PROVIAS NACIONAL | Visor - SDOT"
  ) +
  theme_minimal()

# Length analysis by surface type
vias_cusco |>
  st_drop_geometry() |>
  group_by(superfic_l) |>
  summarise(
    total_km = sum(longitud, na.rm = TRUE),
    n_tramos = n(),
    km_promedio = mean(longitud, na.rm = TRUE)
  ) |>
  arrange(desc(total_km))

# Analysis by status and surface
vias_cusco |>
  st_drop_geometry() |>
  group_by(estado_l, superfic_l) |>
  summarise(
    longitud_total = sum(longitud, na.rm = TRUE),
    .groups = "drop"
  ) |>
  arrange(desc(longitud_total))

# Filter paved roads in good condition
vias_buenas <- vias_cusco |>
  filter(superfic_l == "PAVIMENTADO", estado_l == "BUENO")

# Analysis by province
vias_cusco |>
  st_drop_geometry() |>
  group_by(nombprov) |>
  summarise(
    longitud_total = sum(longitud, na.rm = TRUE),
    n_rutas = n_distinct(codruta),
    .groups = "drop"
  ) |>
  arrange(desc(longitud_total))

# Visualize a specific province
vias_prov <- vias_cusco |>
  filter(nombprov == "CUSCO")

ggplot(vias_prov) +
  geom_sf(aes(color = codruta), linewidth = 1.2) +
  labs(
    title = "Road Network of the Province of Cusco",
    subtitle = "Route Codes",
```

```
    caption = "Source: PROVIAS NACIONAL"
  ) +
  theme_minimal() +
  theme(legend.position = "right")

# General statistics
vias_cusco |>
  st_drop_geometry() |>
  summarise(
    longitud_total = sum(longitud, na.rm = TRUE),
    n_rutas = n_distinct(codruta),
    n_provincias = n_distinct(nombprov),
    km_pavimentado = sum(longitud[superfic_l == "PAVIMENTADO"], na.rm = TRUE),
    pct_pavimentado = (km_pavimentado / longitud_total) * 100
  )

## End(Not run)
```

`get_red_vial_nacional` *Get National Road Network of Peru*

Description

Downloads and loads information on the National Road Network updated by PROVIAS NACIONAL as of July 2022, according to the Route Classifier approved by Supreme Decree 011-2016-MTC and its amendments. Data includes MULTILINESTRING type geometry and detailed information on trajectories, classification, status, surface, and concessions of national roads.

Usage

```
get_red_vial_nacional(
  departamento = NULL,
  show_progress = TRUE,
  force_update = FALSE
)
```

Arguments

`departamento`	Character vector. Name(s) of the department(s) to download. Valid options: "AMAZONAS", "ANCASH", "APURIMAC", "AREQUIPA", "AYACUCHO", "CAJAMARCA", "CALLAO", "CUSCO", "HUANCAVELICA", "HUANUCO", "ICA", "JUNIN", "LA LIBERTAD", "LAMBAYEQUE", "LIMA", "LORETO", "MADRE DE DIOS", "MOQUEGUA", "PASCO", "PIURA", "PUNO", "SAN MARTIN", "TACNA", "TUMBES", "UCAYALI". Case-insensitive. If `NULL`, shows the list of available departments.
`show_progress`	Logical. If `TRUE` (default), shows informative messages about download progress. If `FALSE`, executes silently.
`force_update`	Logical. If `TRUE`, forces a new download of the file even if it exists in cache. Default `FALSE`.

Details

The function downloads data from OSF (Open Science Framework) and stores it in cache during the R session. Data is in GeoPackage (.gpkg) format.

Data Source:

- Source: PROVIAS NACIONAL - Ministry of Transport and Communications
- Update: July 2022
- Legal basis: Supreme Decree 011-2016-MTC and amendments
- Level: National Road Network (National Road System - SINAC)
- Application: National road planning and connectivity analysis

Surface Classification:

- 1 = Paved
- 2 = Unpaved
- 11 = Economic asphalt
- Other codes according to PROVIAS technical specifications

Status Classification:

- 1 = Good
- 2 = Regular
- 3 = Bad

Classification Axes (ejeclas):

- Coast Longitudinal (PE-1N, PE-1S)
- Sierra Longitudinal
- Selva Longitudinal
- Transversal

Cache is stored in: `tempdir()/DEMARCA_cache/red_vial_nacional/`

NOTE: Geometries are MULTILINESTRING type and represent the trajectories of national roads. Some segments may be under concession (field `codconces`).

Value

An `sf` (simple feature) object with MULTILINESTRING geometry containing national road network information, including:

- MULTILINESTRING geometry (road trajectory)
- `inicio`: Start kilometer of the segment
- `fin`: End kilometer of the segment
- `trayectori`: Full description of the route trajectory
- `nrocarril`: Number of lanes

- ejeclas: Classification axis (Coast Longitudinal, Transversal, etc.)
- iddpto: Department code
- nombdep: Department name
- codruta: Route code (e.g., PE-1S, PE-3N, PE-22A)
- jerarq: Route hierarchy (RN = Red Nacional)
- superfic: Surface type code (1-11)
- longitud: Segment length in kilometers
- codconces: Concession code (if applicable)
- codclog: Logistic corridor code
- superfic_l: Descriptive surface type (Paved, Unpaved, etc.)
- codclog_l: Logistic corridor description
- estado: Numeric road status (1-3)
- estado_l: Descriptive status (Good, Regular, Bad)

If multiple departments are requested, it returns a combined sf object.

References

PROVIAS NACIONAL - MTC. Base Cartográfica de la Red Vial Nacional 2022.

Supreme Decree 011-2016-MTC - Route Classifier of the National Road System (SINAC).

DEMARCA OSF Repository: https://osf.io/qy4j6/

See Also

get_red_vial_departamental, get_departamentos, read_sf

Examples

```
## Not run:
# See available departments
get_red_vial_nacional()

# Load national road network of a department
vias_cusco <- get_red_vial_nacional(departamento = "CUSCO")

# Load multiple departments
vias_sur <- get_red_vial_nacional(
  departamento = c("CUSCO", "PUNO", "AREQUIPA")
)

# Visualization with ggplot2
library(ggplot2)
library(dplyr)

# Map of national road network by surface type
ggplot(vias_cusco) +
  geom_sf(aes(color = superfic_l), linewidth = 1) +
```

```
  scale_color_manual(
    values = c(
      "Pavimentado" = "darkblue",
      "Afirmado" = "darkgreen",
      "Asfaltado económico" = "purple"
    ),
    name = "Surface"
  ) +
  labs(
    title = "National Road Network of Cusco",
    subtitle = "By Surface Type - PROVIAS 2022",
    caption = "Source: PROVIAS NACIONAL | Visor - SDOT"
  ) +
  theme_minimal()

# Map by conservation status
ggplot(vias_cusco) +
  geom_sf(aes(color = estado_l, linewidth = estado_l)) +
  scale_color_manual(
    values = c(
      "Bueno" = "darkgreen",
      "Regular" = "orange",
      "Malo" = "red"
    ),
    name = "Status"
  ) +
  scale_linewidth_manual(
    values = c("Bueno" = 1.2, "Regular" = 0.8, "Malo" = 0.6),
    name = "Status"
  ) +
  labs(
    title = "Conservation Status of the National Road Network",
    subtitle = "Department of Cusco",
    caption = "Source: PROVIAS NACIONAL | Visor - SDOT"
  ) +
  theme_minimal()

# Map by route code
ggplot(vias_cusco) +
  geom_sf(aes(color = codruta), linewidth = 1.2) +
  labs(
    title = "National Road Network - Route Codes",
    subtitle = "Cusco",
    color = "Route Code",
    caption = "Source: PROVIAS NACIONAL"
  ) +
  theme_minimal()

# Length analysis by surface type
vias_cusco |>
  st_drop_geometry() |>
  group_by(superfic_l) |>
  summarise(
```

```
    total_km = sum(longitud, na.rm = TRUE),
    n_tramos = n(),
    km_promedio = mean(longitud, na.rm = TRUE)
  ) |>
  arrange(desc(total_km))

# Analysis by classification axis
vias_cusco |>
  st_drop_geometry() |>
  group_by(ejeclas, estado_l) |>
  summarise(
    longitud_total = sum(longitud, na.rm = TRUE),
    .groups = "drop"
  ) |>
  arrange(ejeclas, desc(longitud_total))

# Filter roads under concession
vias_concesion <- vias_cusco |>
  filter(!is.na(codconces))

# Analysis of logistic corridors
vias_cusco |>
  st_drop_geometry() |>
  filter(!is.na(codclog_l)) |>
  group_by(codclog_l) |>
  summarise(
    longitud_total = sum(longitud, na.rm = TRUE),
    n_tramos = n(),
    .groups = "drop"
  ) |>
  arrange(desc(longitud_total))

# Analysis by number of lanes
vias_cusco |>
  st_drop_geometry() |>
  group_by(nrocarril) |>
  summarise(
    longitud_total = sum(longitud, na.rm = TRUE),
    pct = (longitud_total / sum(vias_cusco$longitud, na.rm = TRUE)) * 100,
    .groups = "drop"
  )

# Main national routes
vias_cusco |>
  st_drop_geometry() |>
  group_by(codruta) |>
  summarise(
    longitud_total = sum(longitud, na.rm = TRUE),
    km_inicio = min(inicio, na.rm = TRUE),
    km_fin = max(fin, na.rm = TRUE),
    .groups = "drop"
  ) |>
  arrange(desc(longitud_total))
```

```
# Visualize only longitudinal roads
vias_long <- vias_cusco |>
  filter(grepl("Longitudinal", ejeclas))

ggplot(vias_long) +
  geom_sf(aes(color = ejeclas), linewidth = 1.5) +
  labs(
    title = "Longitudinal Roads",
    subtitle = "National Road Network - Cusco",
    color = "Classification Axis"
  ) +
  theme_minimal()

# General statistics
vias_cusco |>
  st_drop_geometry() |>
  summarise(
    longitud_total = sum(longitud, na.rm = TRUE),
    n_rutas = n_distinct(codruta),
    km_pavimentado = sum(longitud[superfic_l == "Pavimentado"], na.rm = TRUE),
    km_afirmado = sum(longitud[superfic_l == "Afirmado"], na.rm = TRUE),
    pct_pavimentado = (km_pavimentado / longitud_total) * 100,
    km_buen_estado = sum(longitud[estado_l == "Bueno"], na.rm = TRUE),
    pct_buen_estado = (km_buen_estado / longitud_total) * 100
  )

## End(Not run)
```

`get_red_vial_vecinal` *Get Local Road Network of Peru*

Description

Downloads and loads information on the Local Road Network updated by PROVIAS DESCENTRALIZADO as of July 2022, according to the Route Classifier approved by Supreme Decree 011-2016-MTC and its amendments. Data includes MULTILINESTRING type geometry and information on trajectories, status, and surface of local or rural roads. Allows stepwise filtering by department and province.

Usage

```
get_red_vial_vecinal(
  departamento = NULL,
  provincia = NULL,
  show_progress = TRUE,
  force_update = FALSE
)
```

Arguments

departamento	Character vector. Name(s) of the department(s) to download. Valid options: "AMAZONAS", "ANCASH", "APURIMAC", "AREQUIPA", "AYACUCHO", "CAJAMARCA", "CUSCO", "HUANCAVELICA", "HUANUCO", "ICA", "JUNIN", "LA LIBERTAD", "LAMBAYEQUE", "LIMA", "LORETO", "MADRE DE DIOS", "MOQUEGUA", "PASCO", "PIURA", "PUNO", "SAN MARTIN", "TACNA", "TUMBES", "UCAYALI". Case-insensitive. If NULL, shows the list of available departments.
provincia	Character vector. Name(s) of the province(s) to filter within the specified department(s). Optional.
show_progress	Logical. If TRUE (default), shows informative messages about download progress. If FALSE, executes silently.
force_update	Logical. If TRUE, forces a new download of the file even if it exists in cache. Default FALSE.

Details

The function downloads data from OSF (Open Science Framework) and stores it in cache during the R session. Data is in GeoPackage (.gpkg) format.

Data Source:

- Source: PROVIAS DESCENTRALIZADO - Ministry of Transport and Communications
- Update: July 2022
- Legal basis: Supreme Decree 011-2016-MTC and amendments
- Level: Local or Rural Road Network (National Road System - SINAC)
- Application: Local road planning and rural connectivity analysis

Surface Classification:

- 1 = Paved
- 2 = Unpaved
- 3 = Dirt
- 4 = Track

Status Classification:

- 1 = Good
- 2 = Regular
- 3 = Bad

Local Road Network Characteristics: Local or rural roads connect minor population centers, hamlets, and production areas with the departmental and national road networks. They are the responsibility of local governments (municipalities).

Hierarchical Filtering: Filters are applied in cascade:

1. Specified departments are loaded first

2. Then provinces are filtered (if specified)

Cache is stored in: `tempdir()/DEMARCA_cache/red_vial_vecinal/`

NOTE: Geometries are MULTILINESTRING type and represent the trajectories of local roads.

Value

An `sf` (simple feature) object with MULTILINESTRING geometry containing local road network information, including:

- MULTILINESTRING geometry (road trajectory)
- `nombdep`: Department name
- `nombprov`: Province name
- `iddpto`: Department code
- `jerarq`: Route hierarchy (RV = Red Vecinal)
- `trayectori`: Description of the route trajectory
- `estado`: Numeric road status (1, 2, 3)
- `codruta`: Local route code (e.g., LO-517, CU-123)
- `superfic`: Surface type code (1-4)
- `longitud`: Segment length in kilometers
- `estado_l`: Descriptive status (Good, Regular, Bad)
- `superfic_l`: Surface type (Paved, Unpaved, Dirt, Track)

If multiple departments are requested, it returns a combined sf object.

References

PROVIAS DESCENTRALIZADO - MTC. Base Cartográfica de la Red Vial Vecinal 2022.

Supreme Decree 011-2016-MTC - Route Classifier of the National Road System (SINAC).

DEMARCA OSF Repository: https://osf.io/qy4j6/

See Also

`get_red_vial_departamental`, `get_red_vial_nacional`, `get_departamentos`, `get_provincias`, `read_sf`

Examples

```
## Not run:
# See available departments
get_red_vial_vecinal()

# Load local road network of a full department
vias_cusco <- get_red_vial_vecinal(departamento = "CUSCO")

# Filter by specific province
vias_cusco_prov <- get_red_vial_vecinal(
```

```
  departamento = "CUSCO",
  provincia = "CUSCO"
)

# Load multiple departments
vias_sur <- get_red_vial_vecinal(
  departamento = c("CUSCO", "PUNO", "AREQUIPA")
)

# Visualization with ggplot2
library(ggplot2)
library(dplyr)

# Map of local road network by surface type
ggplot(vias_cusco) +
  geom_sf(aes(color = superfic_l), linewidth = 0.6) +
  scale_color_manual(
    values = c(
      "Pavimentado" = "darkblue",
      "Afirmado" = "darkgreen",
      "Sin afirmar" = "orange",
      "Trocha" = "brown"
    ),
    name = "Surface Type"
  ) +
  labs(
    title = "Local Road Network of Cusco",
    subtitle = "By Surface Type - PROVIAS 2022",
    caption = "Source: PROVIAS DESCENTRALIZADO | Visor - SDOT"
  ) +
  theme_minimal()

# Map by conservation status
ggplot(vias_cusco) +
  geom_sf(aes(color = estado_l), linewidth = 0.8) +
  scale_color_manual(
    values = c(
      "Bueno" = "darkgreen",
      "Regular" = "orange",
      "Malo" = "red"
    ),
    name = "Status"
  ) +
  labs(
    title = "Conservation Status of the Local Road Network",
    subtitle = "Department of Cusco",
    caption = "Source: PROVIAS DESCENTRALIZADO | Visor - SDOT"
  ) +
  theme_minimal()

# Length analysis by surface type
vias_cusco |>
  st_drop_geometry() |>
```

```
  group_by(superfic_l) |>
  summarise(
    total_km = sum(longitud, na.rm = TRUE),
    n_tramos = n(),
    km_promedio = mean(longitud, na.rm = TRUE)
  ) |>
  arrange(desc(total_km))

# Analysis by status and surface
vias_cusco |>
  st_drop_geometry() |>
  group_by(estado_l, superfic_l) |>
  summarise(
    longitud_total = sum(longitud, na.rm = TRUE),
    .groups = "drop"
  ) |>
  arrange(desc(longitud_total))

# Filter roads in bad condition needing intervention
vias_mal_estado <- vias_cusco |>
  filter(estado_l == "Malo")

# Analysis by province
vias_cusco |>
  st_drop_geometry() |>
  group_by(nombprov) |>
  summarise(
    longitud_total = sum(longitud, na.rm = TRUE),
    n_rutas = n_distinct(codruta),
    km_trocha = sum(longitud[superfic_l == "Trocha"], na.rm = TRUE),
    pct_trocha = (km_trocha / longitud_total) * 100,
    .groups = "drop"
  ) |>
  arrange(desc(longitud_total))

# Bar chart: km by province
vias_prov_summary <- vias_cusco |>
  st_drop_geometry() |>
  group_by(nombprov) |>
  summarise(total_km = sum(longitud, na.rm = TRUE), .groups = "drop") |>
  arrange(desc(total_km)) |>
  head(10)

ggplot(vias_prov_summary, aes(x = reorder(nombprov, total_km), y = total_km)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(
    title = "Top 10 Provinces with Largest Local Road Network",
    subtitle = "Department of Cusco",
    x = "Province",
    y = "Total Length (km)",
    caption = "Source: PROVIAS DESCENTRALIZADO"
  ) +
```

```
  theme_minimal()

# Surface analysis by province
vias_cusco |>
  st_drop_geometry() |>
  group_by(nombprov, superfic_l) |>
  summarise(km = sum(longitud, na.rm = TRUE), .groups = "drop") |>
  pivot_wider(names_from = superfic_l, values_from = km, values_fill = 0) |>
  arrange(desc(Trocha))

# Filter only paved or unpaved roads
vias_mejoradas <- vias_cusco |>
  filter(superfic_l %in% c("Pavimentado", "Afirmado"))

# General statistics
vias_cusco |>
  st_drop_geometry() |>
  summarise(
    longitud_total = sum(longitud, na.rm = TRUE),
    n_rutas = n_distinct(codruta),
    n_provincias = n_distinct(nombprov),
    km_trocha = sum(longitud[superfic_l == "Trocha"], na.rm = TRUE),
    pct_trocha = (km_trocha / longitud_total) * 100,
    km_afirmado = sum(longitud[superfic_l == "Afirmado"], na.rm = TRUE),
    pct_afirmado = (km_afirmado / longitud_total) * 100,
    km_mal_estado = sum(longitud[estado_l == "Malo"], na.rm = TRUE),
    pct_mal_estado = (km_mal_estado / longitud_total) * 100
  )

# Comparison between provinces
vias_cusco |>
  st_drop_geometry() |>
  group_by(nombprov) |>
  summarise(
    total_km = sum(longitud, na.rm = TRUE),
    pct_bueno = sum(longitud[estado_l == "Bueno"], na.rm = TRUE) / total_km * 100,
    pct_regular = sum(longitud[estado_l == "Regular"], na.rm = TRUE) / total_km * 100,
    pct_malo = sum(longitud[estado_l == "Malo"], na.rm = TRUE) / total_km * 100,
    .groups = "drop"
  ) |>
  arrange(desc(total_km))

# Visualize a specific province
vias_prov <- vias_cusco |>
  filter(nombprov == "CUSCO")

ggplot(vias_prov) +
  geom_sf(aes(color = estado_l), linewidth = 1) +
  scale_color_manual(
    values = c("Bueno" = "green", "Regular" = "orange", "Malo" = "red")
  ) +
  labs(
    title = "Local Road Network - Province of Cusco",
```

```
    subtitle = "Conservation Status",
    color = "Status"
  ) +
  theme_minimal()

## End(Not run)
```

`get_servicios_educativos` *Get Educational Services of Peru*

Description

Downloads and loads information on educational services in Peru, prepared by the Ministry of Education (MINEDU). An educational service is a set of educational activities designed and organized to achieve a learning objective. Data includes POINT type geometry and information on educational level, modality, management, shift, and service characteristics. Allows stepwise filtering by department, province, and district.

Usage

```
get_servicios_educativos(
  departamento = NULL,
  provincia = NULL,
  show_progress = TRUE,
  force_update = FALSE
)
```

Arguments

`departamento`	Character vector. Name(s) of the department(s) to download. Valid options: "AMAZONAS", "ANCASH", "APURIMAC", "AREQUIPA", "AYACUCHO", "CAJAMARCA", "CALLAO", "CUSCO", "HUANCAVELICA", "HUANUCO", "ICA", "JUNIN", "LA LIBERTAD", "LAMBAYEQUE", "LIMA", "LORETO", "MADRE DE DIOS", "MOQUEGUA", "PASCO", "PIURA", "PUNO", "SAN MARTIN", "TACNA", "TUMBES", "UCAYALI". Case-insensitive. If `NULL`, shows the list of available departments.
`provincia`	Character vector. Name(s) of the province(s) to filter within the specified department(s). Optional.
`show_progress`	Logical. If `TRUE` (default), shows informative messages about download progress. If `FALSE`, executes silently.
`force_update`	Logical. If `TRUE`, forces a new download of the file even if it exists in cache. Default `FALSE`.

Details

The function downloads data from OSF (Open Science Framework) and stores it in cache during the R session. Data is in GeoPackage (.gpkg) format.

Data Source:

- Source: Ministry of Education (MINEDU)
- Registration: Educational Institutions Registry
- Level: Educational service
- Application: Educational offer analysis, educational policy planning, and coverage studies by level and modality

Difference between educational facility and educational service:

- **Educational Facility**: Physical infrastructure (property) where services operate. See `get_locales_educativos`.
- **Educational Service**: Pedagogical management unit that provides a specific level or modality of education. A facility may house multiple educational services.

Peruvian Educational System - Levels and Modalities:

Regular Basic Education (EBR):

- **Initial**: Nursery (0-2 years), Kindergarten (3-5 years)
- **Primary**: 6 grades (6-11 years)
- **Secondary**: 5 grades (12-16 years)

Classification by teacher characteristic (d_cod_car):

- **Full Polidocente**: Each section has its own teacher. Typical in urban areas.
- **Multigrade Polidocente**: Several teachers who attend more than one grade each. **Unidocente**: A single teacher attends all grades. Frequent in rural areas with low enrollment.

Hierarchical Filtering: Filters are applied in cascade:

1. Specified departments are loaded first
2. Then provinces are filtered (if specified)

Cache is stored in: `tempdir()/DEMARCA_cache/servicios_educativos/`

NOTE: Geometries are POINT type and represent the location of each educational service.

Value

An `sf` (simple feature) object with POINT geometry containing educational service information, including:

Service Identification:

- POINT geometry (educational service coordinates)
- `cod_mod`: Modular code of the educational service (unique identifier assigned by MINEDU)
- `anexo`: Annex number of the educational service (0 if it is the main headquarters)

- cen_edu: Name of the educational center or institution
- codinst: Institution code (RUC or internal code)

Educational Level and Modality:

- d_niv_mod: Educational service level and modality
- d_forma: Form of care:
  - ESCOLARIZADO: Classroom-based education
  - NO ESCOLARIZADO: Non-classroom programs (PRONOEI, etc.)
  - NO APLICA: Not applicable

Management and Dependency:

- d_gestion: Management type:
  - PÚBLICA DE GESTIÓN DIRECTA: Directly administered by the State
  - PÚBLICA DE GESTIÓN PRIVADA: Public but privately administered (agreements)
  - PRIVADA: Private administration
- d_ges_dep: Specific management dependency
- d_dreugel: DRE or UGEL of administrative dependency

Service Characteristics:

- d_cod_car: Teacher characteristic based on number of grades in charge:
  - POLIDOCENTE COMPLETO: One teacher per grade/section
  - POLIDOCENTE MULTIGRADO: Teachers attend several grades
  - UNIDOCENTE: A single teacher for all grades
  - NO APLICA: Does not correspond to the educational level
  - NO DISPONIBLE: Not available
- d_tipssexo: Service type based on students' sex:
  - MIXTO: Attends both sexes
  - VARONES: Males only
  - MUJERES: Females only
  - NO APLICA: Not applicable
- d_tipoprog: Program type (for non-classroom services)
- d_cod_tur: Service shift

Geographic Location:

- departamen: Department name
- provincia: Province name
- nlat_ie: Educational service latitude (Y coordinate)
- nlong_ie: Educational service longitude (X coordinate)

If multiple departments are requested, it returns a combined sf object.

References

Ministry of Education of Peru (MINEDU). Educational Institutions and Educational Programs Registry.

ESCALE - Statistics of Educational Quality: https://escale.minedu.gob.pe/

DEMARCA OSF Repository: https://osf.io/qy4j6/

See Also

get_locales_educativos, get_departamentos, get_provincias, get_distritos, read_sf

Examples

```
## Not run:
# See available departments
get_servicios_educativos()

# Load educational services of a full department
serv_cusco <- get_servicios_educativos(departamento = "CUSCO")

# Filter by specific province
serv_prov_cusco <- get_servicios_educativos(
  departamento = "CUSCO",
  provincia = "CUSCO"
)

# Filter by specific province
serv_prov_urubamba <- get_servicios_educativos(
  departamento = "CUSCO",
  provincia = "URUBAMBA"
)

# Load multiple departments
serv_sur <- get_servicios_educativos(
  departamento = c("CUSCO", "PUNO", "AREQUIPA")
)

# Visualization with ggplot2
library(ggplot2)
library(dplyr)

# Map of educational services by level
serv_cusco |>
  filter(d_niv_mod %in% c("PRIMARIA", "SECUNDARIA", "INICIAL - JARDÍN")) |>
  ggplot() +
  geom_sf(aes(color = d_niv_mod), size = 1, alpha = 0.6) +
  scale_color_brewer(palette = "Set1", name = "Level") +
  labs(
    title = "Educational Services by Level - Cusco",
    subtitle = "Regular Basic Education",
    caption = "Source: MINEDU - Educational Institutions Registry"
  ) +
```

```
  theme_minimal()

# Summary by level and modality
serv_cusco |>
  sf::st_drop_geometry() |>
  count(d_niv_mod, sort = TRUE)

# Summary by management type
serv_cusco |>
  sf::st_drop_geometry() |>
  count(d_gestion, sort = TRUE)

# Public vs private services
serv_cusco |>
  sf::st_drop_geometry() |>
  count(d_ges_dep, sort = TRUE)

# Filter only initial education
initial_serv <- serv_cusco |>
  filter(grepl("INICIAL", d_niv_mod))

# Filter public secondary services
public_secondary_serv <- serv_cusco |>
  filter(
    d_niv_mod == "SECUNDARIA",
    d_gestion == "PÚBLICA DE GESTIÓN DIRECTA"
  )

# Analysis of unidocente schools (rural)
unidocente_serv <- serv_cusco |>
  filter(d_cod_car == "UNIDOCENTE")

# Map of unidocente schools
ggplot(unidocente_serv) +
  geom_sf(color = "darkred", size = 1.5, alpha = 0.7) +
  labs(
    title = "Unidocente Educational Services - Cusco",
    subtitle = "One teacher for all grades",
    caption = "Source: MINEDU"
  ) +
  theme_minimal()

# Analysis by UGEL
serv_cusco |>
  sf::st_drop_geometry() |>
  count(d_dreugel, sort = TRUE)

# Distribution by attendance shift
serv_cusco |>
  sf::st_drop_geometry() |>
  count(d_cod_tur, sort = TRUE)

# Non-classroom services (PRONOEI, etc.)
```

```
non_classroom_serv <- serv_cusco |>
  filter(d_forma == "NO ESCOLARIZADO")

# Coverage analysis by province and level
coverage <- serv_cusco |>
  sf::st_drop_geometry() |>
  filter(d_niv_mod %in% c("PRIMARIA", "SECUNDARIA")) |>
  group_by(provincia, d_niv_mod) |>
  summarise(
    total_services = n(),
    pct_public = round(
      sum(d_gestion == "PÚBLICA DE GESTIÓN DIRECTA") / n() * 100, 1
    ),
    .groups = "drop"
  ) |>
  tidyr::pivot_wider(
    names_from = d_niv_mod,
    values_from = c(total_services, pct_public)
  )

# Comparative map public vs private
serv_cusco |>
  filter(d_niv_mod == "SECUNDARIA") |>
  mutate(
    management_type = ifelse(
      grepl("PÚBLICA", d_gestion), "Public", "Private"
    )
  ) |>
  ggplot() +
  geom_sf(aes(color = management_type), size = 2, alpha = 0.6) +
  scale_color_manual(
    values = c("Public" = "darkblue", "Private" = "darkred"),
    name = "Management"
  ) +
  labs(
    title = "Secondary Services by Management Type - Cusco",
    caption = "Source: MINEDU"
  ) +
  theme_minimal()

# CETPRO - Technical productive education
cetpro <- serv_cusco |>
  filter(grepl("TÉCNICO PRODUCTIVA", d_niv_mod))

# Higher institutes
higher_institutes <- serv_cusco |>
  filter(grepl("SUPERIOR", d_niv_mod))

## End(Not run)
```

Index