Package 'spData'

May 31, 2024

Title Datasets for Spatial Analysis

Version 2.3.1

Description

Diverse spatial datasets for demonstrating, benchmarking and teaching spatial data analysis. It includes R data of class sf (defined by the package 'sf'), Spatial ('sp'), and nb ('spdep'). Unlike other spatial data packages such as 'rnaturalearth' and 'maps', it also contains data stored in a range of file formats including GeoJSON, ESRI Shapefile and GeoPackage. Some of the datasets are designed to illustrate specific analysis techniques. cycle_hire() and cycle_hire_osm(), for example, is designed to illustrate point pattern analysis techniques.

Depends R (>= 3.3.0)

Imports sp

Suggests foreign, sf (>= 0.9-1), spDataLarge (>= 0.4.0), spdep, spatialreg

License CC0

RoxygenNote 7.3.1

LazyData true

URL https://jakubnowosad.com/spData/

BugReports https://github.com/Nowosad/spData/issues

Additional_repositories https://jakubnowosad.com/drat

Encoding UTF-8

NeedsCompilation no

Author Roger Bivand [aut] (<https://orcid.org/0000-0003-2392-6140>), Jakub Nowosad [aut, cre] (<https://orcid.org/0000-0002-1057-3721>), Robin Lovelace [aut] (<https://orcid.org/0000-0001-5679-6536>), Angelos Mimis [ctb], Mark Monmonier [ctb] (author of the state.vbm dataset), Greg Snow [ctb] (author of the state.vbm dataset)

Maintainer Jakub Nowosad <nowosad.jakub@gmail.com>

Repository CRAN

Date/Publication 2024-05-31 15:10:02 UTC

R topics documented:

afcon *Spatial patterns of conflict in Africa 1966-78*

Description

The afcon data frame has 42 rows and 5 columns, for 42 African countries, exclusing then South West Africa and Spanish Equatorial Africa and Spanish Sahara. The dataset is used in Anselin (1995), and downloaded from before adaptation. The neighbour list object africa.rook.nb is the SpaceStat 'rook.GAL', but is not the list used in Anselin (1995) - paper.nb reconstructs the list used in the paper, with inserted links between Mauritania and Morocco, South Africa and Angola and Zambia, Tanzania and Zaire, and Botswana and Zambia. afxy is the coordinate matrix for the centroids of the countries.

Usage

```
afcon
```

Format

This data frame contains the following columns:

- x: an easting in decimal degrees (taken as centroid of shapefile polygon)
- y: an northing in decimal degrees (taken as centroid of shapefile polygon)
- totcon: index of total conflict 1966-78
- name: country name
- id: country id number as in paper

Note

All source data files prepared by Luc Anselin, Spatial Analysis Laboratory, Department of Agricultural and Consumer Economics, University of Illinois, Urbana-Champaign.

Source

Anselin, L. and John O'Loughlin. 1992. Geography of international conflict and cooperation: spatial dependence and regional context in Africa. In The New Geopolitics, ed. M. Ward, pp. 39-75. Philadelphia, PA: Gordon and Breach. also: Anselin, L. 1995. Local indicators of spatial association, Geographical Analysis, 27, Table 1, p. 103.

Examples

```
data(afcon)
if (requireNamespace("spdep", quietly = TRUE)) {
  library(spdep)
  plot(africa.rook.nb, afxy)
  plot(diffnb(paper.nb, africa.rook.nb), afxy, col="red", add=TRUE)
  text(afxy, labels=attr(africa.rook.nb, "region.id"), pos=4, offset=0.4)
```

```
  moran.test(afcon$totcon, nb2listw(africa.rook.nb))
  moran.test(afcon$totcon, nb2listw(paper.nb))
  geary.test(afcon$totcon, nb2listw(paper.nb))
}
```

`alaska` *Alaska multipolygon*

Description

The object loaded is a `sf` object containing the state of Alaska from the US Census Bureau with a few variables from American Community Survey (ACS)

Usage

```
alaska
```

Format

Formal class 'sf' [package "sf"]; the data contains a data.frame with 1 obs. of 7 variables:

- GEOID: character vector of geographic identifiers
- NAME: character vector of state names
- REGION: character vector of region names
- AREA: area in square kilometers of units class
- total_pop_10: numerical vector of total population in 2010
- total_pop_15: numerical vector of total population in 2015
- geometry: sfc_MULTIPOLYGON

The object is in projected coordinates using Alaska Albers (EPSG:3467).

Source

https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.html

See Also

See the tigris package: https://cran.r-project.org/package=tigris

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  library(sf)
  data(alaska)

  plot(alaska["total_pop_15"])
}
```

auckland *Marshall's infant mortality in Auckland dataset*

Description

(Use example(auckland) to load the data from shapefile and generate neighbour list on the fly). The auckland data frame has 167 rows (census area units — CAU) and 4 columns. The dataset also includes the "nb" object auckland.nb of neighbour relations based on contiguity, and the "polylist" object auckpolys of polygon boundaries for the CAU. The auckland data frame includes the following columns:

Usage

```
auckland
```

Format

This data frame contains the following columns:

- Easting: a numeric vector of x coordinates in an unknown spatial reference system
- Northing: a numeric vector of y coordinates in an unknown spatial reference system
- M77_85: a numeric vector of counts of infant (under 5 years of age) deaths in Auckland, 1977-1985
- Und5_81: a numeric vector of population under 5 years of age at the 1981 Census

Details

The contiguous neighbours object does not completely replicate results in the sources, and was reconstructed from auckpolys; examination of figures in the sources suggests that there are differences in detail, although probably not in substance.

Source

Marshall R M (1991) Mapping disease and mortality rates using Empirical Bayes Estimators, Applied Statistics, 40, 283–294; Bailey T, Gatrell A (1995) Interactive Spatial Data Analysis, Harlow: Longman — INFOMAP data set used with permission.

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  auckland <- sf::st_read(system.file("shapes/auckland.gpkg", package="spData")[1])
  plot(sf::st_geometry(auckland))
  if (requireNamespace("spdep", quietly = TRUE)) {
    auckland.nb <- spdep::poly2nb(auckland)
  }
}
```

`baltimore` *House sales prices, Baltimore, MD 1978*

Description

House sales price and characteristics for a spatial hedonic regression, Baltimore, MD 1978. X,Y on Maryland grid, projection type unknown.

Usage

```
baltimore
```

Format

A data frame with 211 observations on the following 17 variables.

- STATION: a numeric vector
- PRICE: a numeric vector
- NROOM: a numeric vector
- DWELL: a numeric vector
- NBATH: a numeric vector
- PATIO: a numeric vector
- FIREPL: a numeric vector
- AC: a numeric vector
- BMENT: a numeric vector
- NSTOR: a numeric vector
- GAR: a numeric vector
- AGE: a numeric vector
- CITCOU: a numeric vector
- LOTSZ: a numeric vector
- SQFT: a numeric vector
- X: a numeric vector
- Y: a numeric vector

Source

Prepared by Luc Anselin. Original data made available by Robin Dubin, Weatherhead School of Management, Case Western Research University, Cleveland, OH. http://sal.agecon.uiuc.edu/datasets/baltimore.zip

References

Dubin, Robin A. (1992). Spatial autocorrelation and neighborhood quality. Regional Science and Urban Economics 22(3), 433-452.

Examples

```
data(baltimore)
str(baltimore)

if (requireNamespace("sf", quietly = TRUE)) {
  library(sf)
  baltimore_sf <- baltimore %>% st_as_sf(., coords = c("X","Y"))
  plot(baltimore_sf["PRICE"])
}
```

---

boston *Corrected Boston Housing Data*

Description

The `boston.c` data frame has 506 rows and 20 columns. It contains the Harrison and Rubinfeld (1978) data corrected for a few minor errors and augmented with the latitude and longitude of the observations. Gilley and Pace also point out that MEDV is censored, in that median values at or over USD 50,000 are set to USD 50,000. The original data set without the corrections is also included in package `mlbench` as `BostonHousing`. In addition, a matrix of tract point coordinates projected to UTM zone 19 is included as `boston.utm`, and a sphere of influence neighbours list as `boston.soi`.

Format

This data frame contains the following columns:

- TOWN: a factor with levels given by town names
- TOWNNO: a numeric vector corresponding to TOWN
- TRACT: a numeric vector of tract ID numbers
- LON: a numeric vector of tract point longitudes in decimal degrees
- LAT: a numeric vector of tract point latitudes in decimal degrees
- MEDV: a numeric vector of median values of owner-occupied housing in USD 1000
- CMEDV: a numeric vector of corrected median values of owner-occupied housing in USD 1000
- CRIM: a numeric vector of per capita crime
- ZN: a numeric vector of proportions of residential land zoned for lots over 25000 sq. ft per town (constant for all Boston tracts)
- INDUS: a numeric vector of proportions of non-retail business acres per town (constant for all Boston tracts)
- CHAS: a factor with levels 1 if tract borders Charles River; 0 otherwise
- NOX: a numeric vector of nitric oxides concentration (parts per 10 million) per town
- RM: a numeric vector of average numbers of rooms per dwelling
- AGE: a numeric vector of proportions of owner-occupied units built prior to 1940

- DIS: a numeric vector of weighted distances to five Boston employment centres
- RAD: a numeric vector of an index of accessibility to radial highways per town (constant for all Boston tracts)
- TAX: a numeric vector full-value property-tax rate per USD 10,000 per town (constant for all Boston tracts)
- PTRATIO: a numeric vector of pupil-teacher ratios per town (constant for all Boston tracts)
- B: a numeric vector of `1000*(Bk - 0.63)^2` where Bk is the proportion of blacks
- LSTAT: a numeric vector of percentage values of lower status population

Note

Details of the creation of the tract GPKG file: tract boundaries for 1990 (formerly at: http://www.census.gov/geo/cob/bdy/tr/tr
counties in the BOSTON SMSA http://www.census.gov/population/metro/files/lists/historical/63mfips.txt);
tract conversion table 1980/1970 (formerly at : https://www.icpsr.umich.edu/icpsrweb/ICPSR/studies/7913?q=07913&permit
http://www.icpsr.umich.edu/cgi-bin/bob/zipcart2?path=ICPSR&study=7913&bundle=all&ds=1&dups=yes).
The shapefile contains corrections and extra variables (tract 3592 is corrected to 3593; the extra columns are:

- units: number of single family houses
- cu5k: count of units under USD 5,000
- c5_7_5: counts USD 5,000 to 7,500
- C*_*: interval counts
- co50k: count of units over USD 50,000
- median: recomputed median values
- BB: recomputed black population proportion
- censored: whether censored or not
- NOXID: NOX model zone ID
- POP: tract population

Source

Previously available from http://lib.stat.cmu.edu/datasets/boston_corrected.txt

References

Harrison, David, and Daniel L. Rubinfeld, Hedonic Housing Prices and the Demand for Clean Air, *Journal of Environmental Economics and Management*, Volume 5, (1978), 81-102. Original data.

Gilley, O.W., and R. Kelley Pace, On the Harrison and Rubinfeld Data, *Journal of Environmental Economics and Management*, 31 (1996),403-405. Provided corrections and examined censoring.

Pace, R. Kelley, and O.W. Gilley, Using the Spatial Configuration of the Data to Improve Estimation, *Journal of the Real Estate Finance and Economics*, 14 (1997), 333-340.

Bivand, Roger. Revisiting the Boston data set - Changing the units of observation affects estimated willingness to pay for clean air. REGION, v. 4, n. 1, p. 109-127, 2017. https://openjournals.wu.ac.at/ojs/index.php/region/article/view/107.

Examples

```
if (requireNamespace("spdep", quietly = TRUE)) {
  data(boston)
  hr0 <- lm(log(MEDV) ~ CRIM + ZN + INDUS + CHAS + I(NOX^2) + I(RM^2) +
              AGE + log(DIS) + log(RAD) + TAX + PTRATIO + B + log(LSTAT), data = boston.c)
  summary(hr0)
  logLik(hr0)
  gp0 <- lm(log(CMEDV) ~ CRIM + ZN + INDUS + CHAS + I(NOX^2) + I(RM^2) +
              AGE + log(DIS) + log(RAD) + TAX + PTRATIO + B + log(LSTAT), data = boston.c)
  summary(gp0)
  logLik(gp0)
  spdep::lm.morantest(hr0, spdep::nb2listw(boston.soi))
}
if (requireNamespace("sf", quietly = TRUE)) {
boston.tr <- sf::st_read(system.file("shapes/boston_tracts.gpkg",
                           package="spData")[1])
  if (requireNamespace("spdep", quietly = TRUE)) {
    boston_nb <- spdep::poly2nb(boston.tr)
  }
}
```

---

coffee_data *World coffee production data*

Description

A tiny dataset containing estimates of global coffee in thousands of 60 kg bags produced by country. Purpose: teaching **not** research.

Usage

```
coffee_data
```

Format

A data frame (tibble) with 58 for the following 12 variables:

- name_long: name of country or coffee variety
- coffee_production_2016: production in 2016
- coffee_production_2017: production in 2017

Details

The examples section shows how this can be joined with spatial data to create a simple map.

Source

The International Coffee Organization (ICO). See http://www.ico.org/ and http://www.ico.org/prices/m1-exports.pdf

Examples

```
head(coffee_data)
## Not run:
if (requireNamespace("dplyr")) {
library(dplyr)
library(sf)
# found by searching for "global coffee data"
u = "http://www.ico.org/prices/m1-exports.pdf"
download.file(u, "data.pdf", mode = "wb")
if (requireNamespace("pdftables")) { # requires api key
pdftables::convert_pdf(input_file = "data.pdf", output_file = "coffee-data-messy.csv")
d = read_csv("coffee-data-messy.csv")
file.remove("coffee-data-messy.csv")
file.remove("data.pdf")
coffee_data = slice(d, -c(1:9)) %>%
        select(name_long = 1, coffee_production_2016 = 2, coffee_production_2017 = 3) %>%
         filter(!is.na(coffee_production_2016)) %>%
         mutate_at(2:3, str_replace, " ", "") %>%
         mutate_at(2:3, as.integer)
world_coffee = left_join(world, coffee_data)
plot(world_coffee[c("coffee_production_2016", "coffee_production_2017")])
b = c(0, 500, 1000, 2000, 3000)
library(tmap)
tm_shape(world_coffee) +
  tm_fill("coffee_production_2017", title = "Thousand 60kg bags", breaks = b,
          textNA = "No data", colorNA = NULL)
tmap_mode("view") # for an interactive version
}}

## End(Not run)
```

`columbus` *Columbus OH spatial analysis data set*

Description

The `columbus` data frame has 49 rows and 22 columns. Unit of analysis: 49 neighbourhoods in Columbus, OH, 1980 data. In addition the data set includes a `polylist` object `polys` with the boundaries of the neighbourhoods, a matrix of polygon centroids `coords`, and `col.gal.nb`, the neighbours list from an original GAL-format file. The matrix `bbs` is DEPRECATED, but retained for other packages using this data set.

Usage

```
columbus
```

Format

This data frame contains the following columns:

- AREA: computed by ArcView
- PERIMETER: computed by ArcView
- COLUMBUS_: internal polygon ID (ignore)
- COLUMBUS_I: another internal polygon ID (ignore)
- POLYID: yet another polygon ID
- NEIG: neighborhood id value (1-49); conforms to id value used in Spatial Econometrics book.
- HOVAL: housing value (in 1,000 USD)
- INC: household income (in 1,000 USD)
- CRIME: residential burglaries and vehicle thefts per thousand households in the neighborhood
- OPEN: open space in neighborhood
- PLUMB: percentage housing units without plumbing
- DISCBD: distance to CBD
- X: x coordinate (in arbitrary digitizing units, not polygon coordinates)
- Y: y coordinate (in arbitrary digitizing units, not polygon coordinates)
- NSA: north-south dummy (North=1)
- NSB: north-south dummy (North=1)
- EW: east-west dummy (East=1)
- CP: core-periphery dummy (Core=1)
- THOUS: constant=1,000
- NEIGNO: NEIG+1,000, alternative neighborhood id value

Details

The row names of `columbus` and the `region.id` attribute of `polys` are set to `columbus$NEIGNO`.

Note

All source data files prepared by Luc Anselin, Spatial Analysis Laboratory, Department of Agricultural and Consumer Economics, University of Illinois, Urbana-Champaign, http://sal.agecon.uiuc.edu/datasets/columbus.zip.

Source

Anselin, Luc. 1988. Spatial econometrics: methods and models. Dordrecht: Kluwer Academic, Table 12.1 p. 189.

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  columbus <- sf::st_read(system.file("shapes/columbus.gpkg", package="spData")[1])
  plot(sf::st_geometry(columbus))
}

if (requireNamespace("spdep", quietly = TRUE)) {
  library(spdep)
  col.gal.nb <- read.gal(system.file("weights/columbus.gal", package="spData")[1])
}
```

---

congruent *Datasets to illustrate the concept of spatial congruence*

Description

Sample of old (incongruent) and new (congruent) administrative zones from UK statistical agencies

Usage

```
congruent
```

Format

Simple feature geographic data in a projected CRS (OSGB) with random values assigned for teaching purposes.

Source

https://en.wikipedia.org/wiki/ONS_coding_system

Examples

```
if(requireNamespace("sf", quietly = TRUE)) {
  library(sf)
  plot(aggregating_zones$geometry, lwd = 5)
  plot(congruent$geometry, add = TRUE, border = "green", lwd = 2)
  plot(incongruent$geometry, add = TRUE, border = "blue", col = NA)
  rbind(congruent, incongruent)
}
# Code used to download the data:
## Not run:
#devtools::install_github("robinlovelace/ukboundaries")
library(sf)
library(tmap)
library(dplyr)
#library(ukboundaries)
sel = grepl("003|004", msoa2011_lds$geo_label)
aggregating_zones = st_transform(msoa2011_lds[sel, ], 27700)
```

```
# find lsoas in the aggregating_zones
lsoa_touching = st_transform(lsoa2011_lds, 27700)[aggregating_zones, ]
lsoa_cents = st_centroid(lsoa_touching)
lsoa_cents = lsoa_cents[aggregating_zones, ]
sel = lsoa_touching$geo_code %in% lsoa_cents$geo_code
# same for ed zones
ed_touching = st_transform(ed1981, 27700)[aggregating_zones, ]
ed_cents = st_centroid(ed_touching)
ed_cents = ed_cents[aggregating_zones, ]
incongruent_agg_ed = ed_touching[ed_cents, ]
set.seed(2017)
incongruent_agg_ed$value = rnorm(nrow(incongruent_agg_ed), mean = 5)
congruent = aggregate(incongruent_agg_ed["value"], lsoa_touching[sel, ], mean)
congruent$level = "Congruent"
congruent = congruent[c("level", "value")]
incongruent_cents = st_centroid(incongruent_agg_ed)
aggregating_value = st_join(incongruent_cents, congruent)$value.y
incongruent_agg = aggregate(incongruent_agg_ed["value"], list(aggregating_value), FUN = mean)
incongruent_agg$level = "Incongruent"
incongruent = incongruent_agg[c("level", "value")]
summary(st_geometry_type(congruent))
summary(st_geometry_type(incongruent))
incongruent = st_cast(incongruent, "MULTIPOLYGON")
summary(st_geometry_type(incongruent))
summary(st_geometry_type(aggregating_zones))
devtools::use_data(congruent, overwrite = TRUE)
devtools::use_data(incongruent, overwrite = TRUE)
devtools::use_data(aggregating_zones, overwrite = TRUE)

## End(Not run)
```

`cycle_hire` *Cycle hire points in London*

Description

Points representing cycle hire points accross London.

Usage

```
cycle_hire
```

Format

FORMAT:

- id: Id of the hire point
- name: Name of the point
- area: Area they are in

- nbikes: The number of bikes currently parked there
- nempty: The number of empty places
- geometry: sfc_POINT

Source

https://www.data.gov.uk/

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  library(sf)
  data(cycle_hire)
  # or
  cycle_hire <- st_read(system.file("shapes/cycle_hire.geojson", package="spData"))

  plot(cycle_hire)
}

## Not run:
# Download the data
cycle_hire = readr::read_csv("http://cyclehireapp.com/cyclehirelive/cyclehire.csv",
  col_names = FALSE, skip = TRUE)
cycle_hire = cycle_hire[c_names]
c_names = c("id", "name", "area", "lat", "lon", "nbikes", "nempty")
cycle_hire = st_sf(cycle_hire, st_multipoint(c_names[c("lon", "lat")]))

## End(Not run)
```

---

cycle_hire_osm *Cycle hire points in London from OSM*

Description

Dataset downloaded using the osmdata package representing cycle hire points accross London.

Usage

```
cycle_hire_osm
```

Format

- osm_id: The OSM ID
- name: The name of the cycle point
- capacity: How many bikes it can take
- cyclestreets_id: The ID linked to cyclestreets' photomap
- description: Additional description of points
- geometry: sfc_POINT

Source

https://www.openstreetmap.org

See Also

See the osmdata package: https://cran.r-project.org/package=osmdata

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  library(sf)
  data(cycle_hire_osm)
  # or
 cycle_hire_osm <- st_read(system.file("shapes/cycle_hire_osm.geojson", package="spData"))

  plot(cycle_hire_osm)
}

# Code used to download the data:
## Not run:
library(osmdata)
library(dplyr)
library(sf)
q = add_osm_feature(opq = opq("London"), key = "network", value = "tfl_cycle_hire")
lnd_cycle_hire = osmdata_sf(q)
cycle_hire_osm = lnd_cycle_hire$osm_points
nrow(cycle_hire_osm)
plot(cycle_hire_osm)
cycle_hire_osm = dplyr::select(cycle_hire_osm, osm_id, name, capacity,
                               cyclestreets_id, description) %>%
  mutate(capacity = as.numeric(capacity))
names(cycle_hire_osm)
nrow(cycle_hire_osm)

## End(Not run)
```

---

depmunic *Municipality departments of Athens (Sf)*

Description

The geographic boundaries of departments (sf) of the municipality of Athens. This is accompanied by various characteristics in these areas.

Usage

```
depmunic
```

Format

An sf object of 7 polygons with the following 7 variables.

- num_dep: An unique identifier for each municipality department.
- airbnb: The number of airbnb properties in 2017
- museums: The number of museums
- population: The population recorded in census at 2011.
- pop_rest: The number of citizens that the origin is a non european country.
- greensp: The area of green spaces (unit: square meters).
- area: The area of the polygon (unit: square kilometers).

See Also

properties

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  library(sf)
  data(depmunic)

  depmunic$foreigners <- 100*depmunic$pop_rest/depmunic$population
  plot(depmunic["foreigners"], key.pos=1)
}
```

---

eire *Eire data sets*

Description

The Eire data set has been converted to shapefile format and placed in the etc/shapes directory. The initial data objects are now stored as a SpatialPolygonsDataFrame object, from which the contiguity neighbour list is recreated. For purposes of record, the original data set is retained. The `eire.df` data frame has 26 rows and 9 columns. In addition, polygons of the 26 counties are provided as a multipart polylist in eire.polys.utm (coordinates in km, projection UTM zone 30). Their centroids are in eire.coords.utm. The original Cliff and Ord binary contiguities are in eire.nb.

Format

This data frame contains the following columns:

- A: Percentage of sample with blood group A
- towns: Towns/unit area
- pale: Beyond the Pale 0, within the Pale 1
- size: number of blood type samples

- ROADACC: arterial road network accessibility in 1961
- OWNCONS: percentage in value terms of gross agricultural output of each county consumed by itself
- POPCHG: 1961 population as percentage of 1926
- RETSALE: value of retail sales British Pound000
- INCOME: total personal income British Pound000
- names: County names

Source

Upton and Fingleton 1985, - Bailey and Gatrell 1995, ch. 1 for blood group data, Cliff and Ord (1973), p. 107 for remaining variables (also after O'Sullivan, 1968). Polygon borders and Irish data sourced from Michael Tiefelsdorf's SPSS Saddlepoint bundle, originally hosted at: http://geog-www.sbs.ohio-state.edu/faculty/tiefelsdorf/GeoStat.htm.

Examples

```
library(spdep)
eire <- sf::st_read(system.file("shapes/eire.gpkg", package="spData")[1])
eire.nb <- poly2nb(eire)

# Eire physical anthropology blood group data
summary(eire$A)
brks <- round(fivenum(eire$A), digits=2)
cols <- rev(heat.colors(4))
plot(eire, col=cols[findInterval(eire$A, brks, all.inside=TRUE)])
title(main="Percentage with blood group A in Eire")
legend(x=c(-50, 70), y=c(6120, 6050),
  c("under 27.91", "27.91 - 29.26", "29.26 - 31.02", "over 31.02"),
  fill=cols, bty="n")

plot(st_geometry(eire))
plot(eire.nb, st_geometry(eire), add=TRUE)

lA <- lag.listw(nb2listw(eire.nb), eire$A)
summary(lA)
moran.test(eire$A, nb2listw(eire.nb))
geary.test(eire$A, nb2listw(eire.nb))
cor(lA, eire$A)
moran.plot(eire$A, nb2listw(eire.nb), labels=eire$names)
A.lm <- lm(A ~ towns + pale, data=eire)
summary(A.lm)
res <- residuals(A.lm)
brks <- c(min(res),-2,-1,0,1,2,max(res))
cols <- rev(cm.colors(6))

plot(eire, col=cols[findInterval(res, brks, all.inside=TRUE)])
title(main="Regression residuals")
legend(x=c(-50, 70), y=c(6120, 6050),
```

```
  legend=c("under -2", "-2 - -1", "-1 - 0", "0 - 1", "1 - 2", "over 2"),
  fill=cols, bty="n")

lm.morantest(A.lm, nb2listw(eire.nb))
lm.morantest.sad(A.lm, nb2listw(eire.nb))
lm.LMtests(A.lm, nb2listw(eire.nb), test="LMerr")

# Eire agricultural data
brks <- round(fivenum(eire$OWNCONS), digits=2)
cols <- grey(4:1/5)
plot(eire, col=cols[findInterval(eire$OWNCONS, brks, all.inside=TRUE)])
title(main="Percentage own consumption of agricultural produce")
legend(x=c(-50, 70), y=c(6120, 6050),
  legend=c("under 9", "9 - 12.2", "12.2 - 19", "over 19"), fill=cols, bty="n")

moran.plot(eire$OWNCONS, nb2listw(eire.nb))
moran.test(eire$OWNCONS, nb2listw(eire.nb))
e.lm <- lm(OWNCONS ~ ROADACC, data=eire)
res <- residuals(e.lm)
brks <- c(min(res),-2,-1,0,1,2,max(res))
cols <- rev(cm.colors(6))
plot(eire, col=cols[findInterval(res, brks, all.inside=TRUE)])
title(main="Regression residuals")
legend(x=c(-50, 70), y=c(6120, 6050),
  legend=c("under -2", "-2 - -1", "-1 - 0", "0 - 1", "1 - 2", "over 2"),
  fill=cm.colors(6), bty="n")

lm.morantest(e.lm, nb2listw(eire.nb))
lm.morantest.sad(e.lm, nb2listw(eire.nb))
lm.LMtests(e.lm, nb2listw(eire.nb), test="LMerr")
print(localmoran.sad(e.lm, eire.nb, select=seq(along=eire.nb)))
```

`elect80` *1980 Presidential election results*

Description

A data set for 1980 Presidential election results covering 3,107 US counties using geographical coordinates. In addition, three spatial neighbour objects, `k4` not using Great Circle distances, `dll` using Great Circle distances, and `e80_queen` of Queen contiguities for equivalent County polygons taken from file `co1980p020.tar.gz` on the USGS National Atlas site, and a spatial weights object imported from `elect.ford` - a 4-nearest neighbour non-GC row-standardised object, but with coercion to symmetry.

Usage

```
elect80
```

Format

A SpatialPointsDataFrame with 3107 observations on the following 7 variables.

- FIPS: a factor of county FIPS codes
- long: a numeric vector of longitude values
- lat: a numeric vector of latitude values
- pc_turnout: Votes cast as proportion of population over age 19 eligible to vote
- pc_college: Population with college degrees as proportion of population over age 19 eligible to vote
- pc_homeownership: Homeownership as proportion of population over age 19 eligible to vote
- pc_income: Income per capita of population over age 19 eligible to vote

Source

Pace, R. Kelley and Ronald Barry. 1997. "Quick Computation of Spatial Autoregressive Estimators", in Geographical Analysis; sourced from the data folder in the Spatial Econometrics Toolbox for Matlab, http://www.spatial-econometrics.com/html/jplv7.zip, files `elect.dat` and `elect.ford` (with the final line dropped).

Examples

```
if (requireNamespace("sp", quietly = TRUE)) {
  library(sp)
  data(elect80)
  summary(elect80)
  plot(elect80)
}
```

---

elev.tif *Artificial elevation raster data set*

Description

The raster data represents elevation in meters and uses WGS84 as a coordinate reference system.

Examples

```
system.file("raster/elev.tif", package = "spData")
```

getisord *Getis-Ord remote sensing example data*

Description

The go_xyz data frame has 256 rows and 3 columns. Vectors go_x and go_y are of length 16 and give the centres of the grid rows and columns, 30m apart. The data start from the bottom left, Getis and Ord start from the top left - so their 136th grid cell is our 120th.

Format

This data frame contains the following columns:

- x: grid eastings
- y: grid northings
- val: remote sensing values

Source

Getis, A. and Ord, J. K. 1996 Local spatial statistics: an overview. In P. Longley and M. Batty (eds) *Spatial analysis: modelling in a GIS environment* (Cambridge: Geoinformation International), 266.

Examples

```
data(getisord)
image(go_x, go_y, t(matrix(go_xyz$val, nrow = 16, ncol=16, byrow = TRUE)), asp = 1)
text(go_xyz$x, go_xyz$y, go_xyz$val, cex = 0.7)
polygon(c(195, 225, 225, 195), c(195, 195, 225, 225), lwd = 2)
title(main = "Getis-Ord 1996 remote sensing data")
```

grain.tif *Artificial raster dataset representing grain sizes*

Description

The ratified raster dataset represents grain sizes with the three classes clay, silt and sand, and WGS84 as a coordinate reference system.

Examples

```
system.file("raster/grain.tif", package = "spData")
```

hawaii *Hawaii multipolygon*

Description

The object loaded is a `sf` object containing the state of Hawaii from the US Census Bureau with a few variables from American Community Survey (ACS)

Usage

```
hawaii
```

Format

Formal class 'sf' [package "sf"]; the data contains a data.frame with 1 obs. of 7 variables:

- GEOID: character vector of geographic identifiers
- NAME: character vector of state names
- REGION: character vector of region names
- AREA: area in square kilometers of units class
- total_pop_10: numerical vector of total population in 2010
- total_pop_15: numerical vector of total population in 2015
- geometry: sfc_MULTIPOLYGON

The object is in projected coordinates using Hawaii Albers Equal Area Conic (ESRI:102007).

Source

https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.html

See Also

See the tigris package: https://cran.r-project.org/package=tigris

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  library(sf)
  data(hawaii)

  plot(hawaii["total_pop_15"])
}
```

hopkins *Hopkins burnt savanna herb remains*

Description

A 20m square is divided into 40 by 40 0.5m quadrats. Observations are in tens of grams of herb remains, 0 being from 0g to less than 10g, and so on. Analysis was mostly conducted using the interior 32 by 32 grid.

Usage

```
hopkins
```

Format

num [1:40, 1:40] 0 0 0 0 0 0 0 0 0 0 1 ...

Source

Upton, G., Fingleton, B. 1985 Spatial data analysis by example: point pattern and quatitative data, Wiley, pp. 38–39.

References

Hopkins, B., 1965 Observations on savanna burning in the Olokemeji Forest Reserve, Nigeria. Journal of Applied Ecology, 2, 367–381.

Examples

```
data(hopkins)
image(1:32, 1:32, hopkins[5:36,36:5], breaks=c(-0.5, 3.5, 20),
      col=c("white", "black"))
```

house *Lucas county OH housing*

Description

Data on 25,357 single family homes sold in Lucas County, Ohio, 1993-1998 from the county auditor, together with an nb neighbour object constructed as a sphere of influence graph from projected coordinates.

Usage

```
house
```

Format

Formal class 'SpatialPointsDataFrame' [package "sp"] with 5 slots. The data slot is a data frame with 25357 observations on the following 24 variables.

- price: a numeric vector
- yrbuilt: a numeric vector
- stories: a factor with levels `one` `bilevel` `multilvl` `one+half` `two` `two+half` `three`
- TLA: a numeric vector
- wall: a factor with levels `stucdrvt` `ccbtile` `metlvnyl` `brick` `stone` `wood` `partbrk`
- beds: a numeric vector
- baths: a numeric vector
- halfbaths: a numeric vector
- frontage: a numeric vector
- depth: a numeric vector
- garage: a factor with levels `no garage` `basement` `attached` `detached` `carport`
- garagesqft: a numeric vector
- rooms: a numeric vector
- lotsize: a numeric vector
- sdate: a numeric vector
- avalue: a numeric vector
- s1993: a numeric vector
- s1994: a numeric vector
- s1995: a numeric vector
- s1996: a numeric vector
- s1997: a numeric vector
- s1998: a numeric vector
- syear: a factor with levels `1993` `1994` `1995` `1996` `1997` `1998`
- age: a numeric vector

Its projection is `CRS(+init=epsg:2834)`, the Ohio North State Plane.

Source

Dataset included in the Spatial Econometrics toolbox for Matlab, http://www.spatial-econometrics.com/html/jplv7.zip.

Examples

```
if (requireNamespace("sp", quietly = TRUE)) {
  library(sp)
  data(house)
  str(house)
  plot(house)
}
```

huddersfield *Prevalence of respiratory symptoms*

Description

Prevalence of respiratory symptoms in 71 school catchment areas in Huddersfield, Northern England

Usage

```
huddersfield
```

Format

A data frame with 71 observations on the following 2 variables.

- `cases`: Prevalence of at least mild conditions
- `total`: Number of questionnaires returned

Source

Martuzzi M, Elliott P (1996) Empirical Bayes estimation of small area prevalence of non-rare conditions, Statistics in Medicine 15, 1867–1873, pp. 1870–1871.

Examples

```
data(huddersfield)
str(huddersfield)
```

jenks71 *Illinois 1959 county gross farm product value per acre*

Description

Classic data set for the choice of class intervals for choropleth maps.

Usage

```
jenks71
```

Format

A data frame with 102 observations on the following 2 variables.

- `jenks71`: a numeric vector: Per acre value of gross farm products in dollars by county for Illinois in #' 1959
- `area`: a numeric vector: county area in square miles

Source

Jenks, G. F., Caspall, F. C., 1971. "Error on choroplethic maps: definition, measurement, reduction". Annals, Association of American Geographers, 61 (2), 217–244

Examples

```
data(jenks71)
jenks71
```

`lnd` *The boroughs of London*

Description

Polygons representing large administrative zones in London

Usage

```
lnd
```

Format

- NAME: Borough name
- GSS_CODE: Official code
- HECTARES: How many hectares
- NONLD_AREA: Area outside London
- ONS_INNER: Office for national statistics code
- SUB_2009: Empty column
- SUB_2006: Empty column
- geometry: sfc_MULTIPOLYGON

Source

https://github.com/Robinlovelace/Creating-maps-in-R

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  library(sf)
  data(lnd)
  summary(lnd)
  plot(st_geometry(lnd))
}
```

`nc.sids` *North Carolina SIDS data*

Description

(Use `example(nc.sids)` to read the data set from shapefile, together with import of two different list of neighbours). The `nc.sids` data frame has 100 rows and 21 columns. It contains data given in Cressie (1991, pp. 386-9), Cressie and Read (1985) and Cressie and Chan (1989) on sudden infant deaths in North Carolina for 1974-78 and 1979-84. The data set also contains the neighbour list given by Cressie and Chan (1989) omitting self-neighbours (ncCC89.nb), and the neighbour list given by Cressie and Read (1985) for contiguities (ncCR85.nb). The data are ordered by county ID number, not alphabetically as in the source tables `sidspolys` is a "polylist" object of polygon boundaries, and `sidscents` is a matrix of their centroids.

Usage

```
nc.sids
```

Format

This data frame contains the following columns:

- SP_ID: SpatialPolygons ID
- CNTY_ID: county ID
- east: eastings, county seat, miles, local projection
- north: northings, county seat, miles, local projection
- L_id: Cressie and Read (1985) L index
- M_id: Cressie and Read (1985) M index
- names: County names
- AREA: County polygon areas in degree units
- PERIMETER: County polygon perimeters in degree units
- CNTY_: Internal county ID
- NAME: County names
- FIPS: County ID
- FIPSNO: County ID
- CRESS_ID: Cressie papers ID
- BIR74: births, 1974-78
- SID74: SID deaths, 1974-78
- NWBIR74: non-white births, 1974-78
- BIR79: births, 1979-84
- SID79: SID deaths, 1979-84
- NWBIR79: non-white births, 1979-84

Source

Cressie, N (1991), *Statistics for spatial data*. New York: Wiley, pp. 386–389; Cressie, N, Chan NH (1989) Spatial modelling of regional variables. *Journal of the American Statistical Association*, 84, 393–401; Cressie, N, Read, TRC (1985) Do sudden infant deaths come in clusters? *Statistics and Decisions* Supplement Issue 2, 333–349; http://sal.agecon.uiuc.edu/datasets/sids.zip.

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  if (requireNamespace("spdep", quietly = TRUE)) {
    library(spdep)
    nc.sids <- sf::st_read(system.file("shapes/sids.gpkg", package="spData")[1])
    row.names(nc.sids) <- as.character(nc.sids$FIPS)
    rn <- row.names(nc.sids)
    ncCC89_nb <- read.gal(system.file("weights/ncCC89.gal", package="spData")[1],
                          region.id=rn)
    ncCR85_nb <- read.gal(system.file("weights/ncCR85.gal", package="spData")[1],
                          region.id=rn)

    plot(sf::st_geometry(nc.sids), border="grey")
    plot(ncCR85_nb, sf::st_geometry(nc.sids), add=TRUE, col="blue")
    plot(sf::st_geometry(nc.sids), border="grey")
    plot(ncCC89_nb, sf::st_geometry(nc.sids), add=TRUE, col="blue")
  }
}
```

---

nydata *New York leukemia data*

Description

New York leukemia data taken from the data sets supporting Waller and Gotway 2004 (the data should be loaded by running `example(NY_data)` to demonstrate spatial data import techniques)

Usage

```
nydata
```

Format

A data frame with 281 observations on the following 12 variables, and the binary coded spatial weights used in the source.

- AREANAME: name of census tract
- AREAKEY: unique FIPS code for each tract
- X: x-coordinate of tract centroid (in km)
- Y: y-coordinate of tract centroid (in km)

- POP8: population size (1980 U.S. Census)
- TRACTCAS: number of cases 1978-1982
- PROPCAS: proportion of cases per tract
- PCTOWNHOME: percentage of people in each tract owning their own home
- PCTAGE65P: percentage of people in each tract aged 65 or more
- Z: ransformed propoprtions
- AVGIDIST: average distance between centroid and TCE sites
- PEXPOSURE: "exposure potential": inverse distance between each census tract centroid and the nearest TCE site, IDIST, transformed via log(100*IDIST)

Details

The examples section shows how the DBF files from the book website for Chapter 9 were converted into the `nydata` data frame and the `listw_NY` spatial weights list. The `shapes` directory includes the original version of the UTM18 census tract boundaries imported from BNA format (http://sedac.ciesin.columbia.edu/ftpsite/pub/census/usa/tiger/ny/bna_st/t8_36.zip) before the OGR/GDAL BNA driver was available. The `NY8_utm18` shapefile was constructed using a bna2mif converter and converted to shapefile format after adding data using `writeOGR`. The new file `NY8_bna_utm18.gpkg` has been constructed from the original BNA file, but read using the OGR BNA driver with GEOS support. The NY8 shapefile includes invalid polygons, but because the OGR BNA driver may have GEOS support (used here), the tract polygon objects are valid.

Source

http://www.sph.emory.edu/~lwaller/ch9index.htm

References

Waller, L. and C. Gotway (2004) *Applied Spatial Statistics for Public Health Data.* New York: John Wiley and Sons.

Examples

```
## NY leukemia

if (requireNamespace("sf", quietly = TRUE)) {
library(foreign)
nydata <- read.dbf(system.file("misc/nydata.dbf", package="spData")[1])
nydata <- sf::st_as_sf(nydata, coords=c("X", "Y"), remove=FALSE)
plot(sf::st_geometry(nydata))

nyadjmat <- as.matrix(read.dbf(system.file("misc/nyadjwts.dbf",
                                           package="spData")[1])[-1])
ID <- as.character(names(read.dbf(system.file("misc/nyadjwts.dbf",
                                              package="spData")[1]))[-1])
identical(substring(ID, 2, 10), substring(as.character(nydata$AREAKEY), 2, 10))

if (requireNamespace("sf", quietly = TRUE)) {
library(spdep)
```

```
listw_NY <- mat2listw(nyadjmat, as.character(nydata$AREAKEY), style="B")
}
}
```

---

nz *Regions in New Zealand*

Description

Polygons representing the 16 regions of New Zealand (2018). See https://en.wikipedia.org/wiki/Regions_of_New_Zealand for a description of these regions and https://www.stats.govt.nz for information on the data source

Usage

```
nz
```

Format

FORMAT:

- Name: Name
- Island: Island
- Land_area: Land area
- Population: Population
- Median_income: Median income (NZD)
- Sex_ratio: Sex ratio (male/female)
- geom: sfc_MULTIPOLYGON

Source

https://www.stats.govt.nz

https://en.wikipedia.org/wiki/Regions_of_New_Zealand

See Also

See the nzcensus package: https://github.com/ellisp/nzelect

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  library(sf)
  summary(nz)
  plot(nz)
}
## Not run:
# Find "Regional Council 2018 Clipped (generalised)"
# select the GeoPackage option in the "Vectors/tables" dropdown
# at https://datafinder.stats.govt.nz/data/ (requires registration)
# Save the result as:
unzip("statsnzregional-council-2018-clipped-generalised-GPKG.zip")
library(sf)
library(tidyverse)
nz_full = st_read("regional-council-2018-clipped-generalised.gpkg")
print(object.size(nz_full), units = "Kb") # 14407.2 Kb
nz = rmapshaper::ms_simplify(nz_full, keep = 0.001, sys = TRUE)
print(object.size(nz), units = "Kb") # 39.9 Kb
names(nz)
nz$REGC2018_V1_00_NAME
nz = filter(nz, REGC2018_V1_00_NAME != "Area Outside Region") %>%
        select(Name = REGC2018_V1_00_NAME, `Land_area` = LAND_AREA_SQ_KM)
# regions basic info
# devtools::install_github("hadley/rvest")
library(rvest)
doc = read_html("https://en.wikipedia.org/wiki/Regions_of_New_Zealand") %>%
        html_nodes("div table")
tab = doc[[3]] %>% html_table()
tab = tab %>% select(Name = Region, Population = `Population[20]`, Island)
tab = tab %>% mutate(Population = str_replace_all(Population, ",", "")) %>%
        mutate(Population = as.numeric(Population)) %>%
        mutate(Name = str_remove_all(Name, " \\([1-9]\\)?.+"))
nz$Name = as.character(nz$Name)
nz$Name = str_remove(nz$Name, " Region")
nz$Name %in% tab$Name
# regions additional info
library(nzcensus)
nz_add_data = REGC2013 %>%
        select(Name = REGC2013_N, Median_income = MedianIncome2013,
               PropFemale2013, PropMale2013) %>%
        mutate(Sex_ratio = PropMale2013 / PropFemale2013) %>%
        mutate(Name = gsub(" Region", "", Name)) %>%
        select(Name, Median_income, Sex_ratio)
# data join
nz = left_join(nz, tab, by = "Name") %>%
        left_join(nz_add_data, by = "Name") %>%
        select(Name, Island, Land_area, Population, Median_income, Sex_ratio)

## End(Not run)
```

nz_height *High points in New Zealand*

Description

Top 101 heighest points in New Zealand (2017). See https://data.linz.govt.nz/layer/50284-nz-height-points-to for details.

Usage

```
nz_height
```

Format

FORMAT:

- t50_fid: ID
- elevation: Height above sea level in m
- geometry: sfc_POINT

Source

https://data.linz.govt.nz

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  library(sf)
  summary(nz_height)
  plot(nz$geom)
  plot(nz_height$geom, add = TRUE)
}
## Not run:
library(dplyr)
# After downloading data
unzip("lds-nz-height-points-topo-150k-SHP.zip")
nz_height = st_read("nz-height-points-topo-150k.shp") %>%
  top_n(n = 100, wt = elevation)
library(tmap)
tmap_mode("view")
qtm(nz) +
  qtm(nz_height)
f = list.files(pattern = "*nz-height*")
file.remove(f)

## End(Not run)
```

properties *Dataset of properties in the municipality of Athens (sf)*

Description

A dataset of apartments in the municipality of Athens for 2017. Point location of the properties is given together with their main characteristics and the distance to the closest metro/train station.

Usage

```
properties
```

Format

An sf object of 1000 points with the following 6 variables.

- id: An unique identifier for each property.
- size : The size of the property (unit: square meters)
- price : The asking price (unit: euros)
- prpsqm : The asking price per squre meter (unit: euroes/square meter).
- age : Age of property in 2017 (unit: years).
- dist_metro: The distance to closest train/metro station (unit: meters).

See Also

depmunic

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  if (requireNamespace("spdep", quietly = TRUE)) {
    library(sf)
    library(spdep)

    data(properties)

    summary(properties$prpsqm)

    pr.nb.800 <- dnearneigh(properties, 0, 800)
    pr.listw <- nb2listw(pr.nb.800)

    moran.test(properties$prpsqm, pr.listw)
    moran.plot(properties$prpsqm, pr.listw, xlab = "Price/m^2", ylab = "Lagged")
  }
}
```

`seine` *Small river network in France*

Description

Lines representing the Seine, Marne and Yonne rivers.

Usage

```
seine
```

Format

FORMAT:

- name: name
- geometry: sfc_MULTILINESTRING

The object is in the RGF93 / Lambert-93 CRS.

Source

https://www.naturalearthdata.com/

See Also

See the rnaturalearth package: https://cran.r-project.org/package=rnaturalearth

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  library(sf)
  seine
  plot(seine)
}
## Not run:
library(sf)
library(rnaturalearth)
library(tidyverse)

seine = ne_download(scale = 10, type = "rivers_lake_centerlines",
                    category = "physical", returnclass = "sf") %>%
        filter(name %in% c("Yonne", "Seine", "Marne")) %>%
        select(name = name_en) %>%
        st_transform(2154)

## End(Not run)
```

SplashDams *Data for Splash Dams in western Oregon*

Description

Data for Splash Dams in western Oregon

Usage

```
SplashDams
```

Format

Formal class 'SpatialPointsDataFrame with 232 obs. of 6 variables:

- streamName
- locationCode
- height
- lastDate
- owner
- datesUsed

Source

R. R. Miller (2010) Is the Past Present? Historical Splash-dam Mapping and Stream Disturbance Detection in the Oregon Coastal Province. MSc. thesis, Oregon State University; packaged by Jonathan Callahan

Examples

```
if (requireNamespace("sp", quietly = TRUE)) {
  library(sp)
  data(SplashDams)
  plot(SplashDams, axes=TRUE)
}
```

state.vbm *US State Visibility Based Map*

Description

A SpatialPolygonsDataFrame object to plot a Visibility Based Map.

Usage

```
state.vbm
```

Format

An object of class `SpatialPolygonsDataFrame` with 50 rows and 2 columns.

Details

A SpatialPolygonsDataFrame object to plot a map of the US states where the sizes of the states have been adjusted to be more equal. This map can be useful for plotting state data using colors patterns without the larger states dominating and the smallest states being lost. The original map is copyrighted by Mark Monmonier. Official publications based on this map should acknowledge him. Comercial publications of maps based on this probably need permission from him to use.

Author(s)

Greg Snow <greg.snow@imail.org> (of this compilation)

Source

The data was converted from the maps library for S-PLUS. S-PLUS uses the map with permission from the author. This version of the data has not received permission from the author (no attempt made, not that it was refused), most of my uses I feel fall under fair use and do not violate copyright, but you will need to decide for yourself and your applications.

References

http://www.markmonmonier.com/index.htm, http://euclid.psych.yorku.ca/SCS/Gallery/bright-ideas.html

Examples

```
if (requireNamespace("sp", quietly = TRUE)) {
  library(sp)
  data(state.vbm)
  plot(state.vbm)

  tmp <- state.x77[, 'HS Grad']
  tmp2 <- cut(tmp, seq(min(tmp), max(tmp), length.out=11),
            include.lowest=TRUE)
```

```
  plot(state.vbm, col=cm.colors(10)[tmp2])
}
```

---

urban_agglomerations *Major urban areas worldwide*

Description

Dataset in a 'long' form from the United Nations population division with projections up to 2050. Includes only the top 30 largest areas by population at 5 year intervals.

Usage

```
urban_agglomerations
```

Format

Selected variables:

- year: Year of population estimate
- country_code: Code of country
- urban_agglomeration: Name of the urban agglomeration
- population_millions: Estimated human population
- geometry: sfc_POINT

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  library(sf)
  plot(urban_agglomerations)
}
# Code used to download the data:
## Not run:
f = "WUP2018-F11b-30_Largest_Cities_in_2018_by_time.xls"
download.file(
  destfile = f,
  url = paste0("https://population.un.org/wup/Download/Files/", f)
 )
library(dplyr)
library(sf)
urban_agglomerations = readxl::read_excel(f, skip = 16) %>%
    st_as_sf(coords = c("Longitude", "Latitude"), crs = 4326)
names(urban_agglomerations)
names(urban_agglomerations) <- gsub(" |\\n", "_", tolower(names(urban_agglomerations)) ) %>%
        gsub("\\(|\\)", "", .)
names(urban_agglomerations)
urban_agglomerations
usethis::use_data(urban_agglomerations, overwrite = TRUE)
```

```
file.remove("WUP2018-F11b-30_Largest_Cities_in_2018_by_time.xls")

## End(Not run)
```

---

used.cars *US 1960 used car prices*

Description

The used.cars data frame has 48 rows and 2 columns. The data set includes a neighbours list for the 48 states excluding DC from poly2nb().

Usage

```
used.cars
```

Format

This data frame contains the following columns:

- tax.charges: taxes and delivery charges for 1955-9 new cars
- price.1960: 1960 used car prices by state

Source

Hanna, F. A. 1966 Effects of regional differences in taxes and transport charges on automobile consumption, in Ostry, S., Rhymes, J. K. (eds) Papers on regional statistical studies, Toronto: Toronto University Press, pp. 199-223.

References

Hepple, L. W. 1976 A maximum likelihood model for econometric estimation with spatial series, in Masser, I (ed) Theory and practice in regional science, London: Pion, pp. 90-104.

Examples

```
if (requireNamespace("spdep", quietly = TRUE)) {
  library(spdep)
  data(used.cars)
  moran.test(used.cars$price.1960, nb2listw(usa48.nb))
  moran.plot(used.cars$price.1960, nb2listw(usa48.nb),
           labels=rownames(used.cars))
  uc.lm <- lm(price.1960 ~ tax.charges, data=used.cars)
  summary(uc.lm)

  lm.morantest(uc.lm, nb2listw(usa48.nb))
  lm.morantest.sad(uc.lm, nb2listw(usa48.nb))
  lm.LMtests(uc.lm, nb2listw(usa48.nb))
```

```
  if (requireNamespace("spatialreg", quietly = TRUE)) {
    library(spatialreg)
    uc.err <- errorsarlm(price.1960 ~ tax.charges, data=used.cars,
                         nb2listw(usa48.nb), tol.solve=1.0e-13,
                         control=list(tol.opt=.Machine$double.eps^0.3))
    summary(uc.err)
    uc.lag <- lagsarlm(price.1960 ~ tax.charges, data=used.cars,
                       nb2listw(usa48.nb), tol.solve=1.0e-13,
                       control=list(tol.opt=.Machine$double.eps^0.3))
    summary(uc.lag)
    uc.lag1 <- lagsarlm(price.1960 ~ 1, data=used.cars,
                        nb2listw(usa48.nb), tol.solve=1.0e-13,
                        control=list(tol.opt=.Machine$double.eps^0.3))
    summary(uc.lag1)
    uc.err1 <- errorsarlm(price.1960 ~ 1, data=used.cars,
                          nb2listw(usa48.nb), tol.solve=1.0e-13,
                          control=list(tol.opt=.Machine$double.eps^0.3),
                          Durbin=FALSE)
    summary(uc.err1)
  }

}
```

`us_states` *US states polygons*

Description

The object loaded is a `sf` object containing the contiguous United States data from the US Census Bureau with a few variables from American Community Survey (ACS)

Usage

```
us_states
```

Format

Formal class 'sf' [package "sf"]; the data contains a data.frame with 49 obs. of 7 variables:

- GEOID: character vector of geographic identifiers
- NAME: character vector of state names
- REGION: character vector of region names
- AREA: area in square kilometers of units class
- total_pop_10: numerical vector of total population in 2010
- total_pop_15: numerical vector of total population in 2015
- geometry: sfc_MULTIPOLYGON

The object is in geographical coordinates using the NAD83 datum.

Source

https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.html

See Also

See the tigris package: https://cran.r-project.org/package=tigris

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  library(sf)
  data(us_states)

  plot(us_states["REGION"])
}
```

---

us_states_df *the American Community Survey (ACS) data*

Description

The object loaded is a `data.frame` object containing the US states data from the American Community Survey (ACS)

Usage

```
us_states_df
```

Format

Formal class 'data.frame'; the data contains a data.frame with 51 obs. of 5 variables:

- state: character vector of state names
- median_income_10: numerical vector of median income in 2010
- median_income_15: numerical vector of median income in 2010
- poverty_level_10: numerical vector of number of people with income below poverty level in 2010
- poverty_level_15: numerical vector of number of people with income below poverty level in 2015

Source

https://www.census.gov/programs-surveys/acs/

See Also

See the tidycensus package: https://cran.r-project.org/package=tidycensus

Examples

```
data(us_states_df)

summary(us_states_df)
```

`wheat` *Mercer and Hall wheat yield data*

Description

Mercer and Hall wheat yield data, based on version in Cressie (1993), p. 455.

Usage

```
wheat
```

Format

The format of the object generated by running `data(wheat)` is a three column data frame made available by Hongfei Li. The example section shows how to convert this to the object used in demonstrating the `aple` function, and is a formal class 'SpatialPolygonsDataFrame' [package "sp"] with 5 slots; the data slot is a data frame with 500 observations on the following 6 variables.

- lat: local coordinates northings ordered north to south
- yield: Mercer and Hall wheat yield data
- r: rows south to north; levels in distance units of plot centres
- c: columns west to east; levels in distance units of plot centres
- lon: local coordinates eastings
- lat1: local coordinates northings ordered south to north

Note

The value of 4.03 was changed to 4.33 (wheat[71,]) 13 January 2014; thanks to Sandy Burden; cross-checked with http://www.itc.nl/personal/rossiter/teach/R/mhw.csv, which agrees.

Source

Cressie, N. A. C. (1993) Statistics for Spatial Data. Wiley, New York, p. 455.

References

Mercer, W. B. and Hall, A. D. (1911) The experimental error of field trials. Journal of Agricultural Science 4, 107-132.

Examples

```
if (requireNamespace("sp", quietly = TRUE)) {
library(sp)
data(wheat)
wheat$lat1 <- 69 - wheat$lat
wheat$r <- factor(wheat$lat1)
wheat$c <- factor(wheat$lon)
wheat_sp <- wheat
coordinates(wheat_sp) <- c("lon", "lat1")
wheat_spg <- wheat_sp

gridded(wheat_spg) <- TRUE
wheat_spl <- as(wheat_spg, "SpatialPolygons")
df <- as(wheat_spg, "data.frame")
row.names(df) <- sapply(slot(wheat_spl, "polygons"),
                        function(x) slot(x, "ID"))
wheat <- SpatialPolygonsDataFrame(wheat_spl, data=df)
}


if (requireNamespace("sf", quietly = TRUE)) {
  library(sf)
  wheat <- st_read(system.file("shapes/wheat.shp", package="spData"))
  plot(wheat)
}
```

---

world *World country polygons*

Description

The object loaded is a `sf` object containing a world map data from Natural Earth with a few variables from World Bank

Usage

```
world
```

Format

Formal class 'sf' [package "sf"]; the data contains a data.frame with 177 obs. of 11 variables:

- iso_a2: character vector of ISO 2 character country codes
- name_long: character vector of country names
- continent: character vector of continent names
- region_un: character vector of region names
- subregion: character vector of subregion names

- type: character vector of type names
- area_km2: integer vector of area values
- pop: integer vector of population in 2014
- lifeExp: integer vector of life expectancy at birth in 2014
- gdpPercap: integer vector of per-capita GDP in 2014
- geom: sfc_MULTIPOLYGON

The object is in geographical coordinates using the WGS84 datum.

Source

https://www.naturalearthdata.com/

https://data.worldbank.org/

See Also

See the rnaturalearth package: https://cran.r-project.org/package=rnaturalearth

Examples

```
if (requireNamespace("sf", quietly = TRUE)) {
  library(sf)
  data(world)
  # or
  world <- st_read(system.file("shapes/world.gpkg", package="spData"))

  plot(world)
}
```

---

worldbank_df *World Bank data*

Description

The object loaded is a `data.frame` object containing data from World Bank

Usage

```
worldbank_df
```

Format

Formal class 'data.frame'; the data contains a data.frame with 177 obs. of 7 variables:

- name: character vector of country names
- iso_a2: character vector of ISO 2 character country codes
- HDI: human development index (HDI)
- urban_pop: urban population
- unemployment: unemployment, total (% of total labor force)
- pop_growth: population growth (annual %)
- literacy: adult literacy rate, population 15+ years, both sexes (%)

Source

https://data.worldbank.org/

See Also

See the wbstats package: https://cran.r-project.org/web/packages/wbstats

Examples

```
data(worldbank_df)
# or
worldbank_df <- read.csv(system.file("misc/worldbank_df.csv", package="spData"))

summary(worldbank_df)
```

Index