Package: arear (via r-universe)

August 21, 2024

Title Geospatial Convenience Functions and a Supply Demand Catchment Area Generator

Version 0.1.1

Description Geospatial function collection from the COVID-19 pandemic. The main focus of this was integrating geospatial demographic, hospital capacity and COVID data from England, Scotland, Wales and Northern Ireland, all of which were available on different sites and methods. The UK has a wide range of administrative geographic boundaries for different purposes and moving from different scales and resolutions proved necessary. As the geospatial operations are quite time consuming but don't need to be repeated the ability to cache results of geospatial transformations is useful and embedded into these functions.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Suggests knitr, rmarkdown, testthat, ggspatial, covr, tidyverse, ggraph, readr, tidygraph, devtools, markdown, pkgdown, scales, tidyr, rprojroot

VignetteBuilder knitr

Imports rmapshaper, sf, lwgeom, dplyr, magrittr, rlang, fs, stringr, here, spdep, glue, leaflet, purrr, tibble, digest, rappdirs, ggrepel, ggplot2, gtable, grid, gridExtra, patchwork, httr, tidyselect, jsonlite

Depends R (>= 2.10)

RoxygenNote 7.2.3

URL `https://terminological.github.io/arear/index.html`, `https://github.com/terminological/arear`

BugReports `https://github.com/terminological/arear/issues`

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

RemoteUrl https://github.com/terminological/arear

RemoteRef 0.1.1

RemoteSha 93077a49f8c36ec9e21fa44d359981b31bd203c1

Contents

`apiTrusts` *NHS trust admissions data from the legacy COVID api from 2021*

Description

NHS trust admissions data from the legacy COVID api from 2021

Usage

```
apiTrusts
```

Format

A 72928 line data frame:

date the ONS code for the ares

area_name The name for the area

area_type the type of area (LSOA, DZ or LGD)

area_code the NHS ODS code

metric hospital cases

metric_name hospital cases

value the count

`CA19` *Map data: CA19*

Description

Map data: CA19

Usage

```
CA19
```

Format

An object of class `character` of length 1.

References

https://www.spatialdata.gov.scot/geonetwork/srv/eng/catalog.search#/metadata/1cd57ea6-8d6e-412b-a9d

cache_clear *Clear data from the passthrough cache for complex or long running operations*

Description

Clear data from the passthrough cache for complex or long running operations

Usage

```
cache_clear(
  cache = getOption("arear.cache.dir", default = rappdirs::user_cache_dir("arear")),
  interactive = TRUE
)
```

Arguments

`cache` the location of the cache as a directory. May get its value from options("arear.cache.dir") or the default value of rappdirs::user_cache_dir("arear")

`interactive` if FALSE will delete the files without warning. Defaults to TRUE

Value

nothing

catchment *Create a catchment area map and cache the result*

Description

This implements the label propagation algorithm described in our upcoming paper.

Usage

```
catchment(
  supplyShape,
  supplyIdVar = "code",
  supplyVar,
  supplyOutputVars = supplyShape %>% dplyr::groups(),
  demandShape,
  demandIdVar = "code",
  demandVar,
  growthConstant = 1.2,
  bridges = arear::ukconnections,
  outputMap = TRUE,
  ...
)
```

Arguments

`supplyShape` - a sf object containing a list of the locations of supply points, with a column containing supply capacity, for example NHS hospital sites, with a bed

`supplyIdVar` - the variable name of the identifier of the supplier or group of suppliers. For example this could be an NHS trust (multiple sites)

`supplyVar` - the column name of the supply parameter. This could be number of beds in a hospital.

`supplyOutputVars` - (optional - defaults to grouping) the columns from the input that are to be retained in the output

`demandShape` - the sf object with the geographical map of the demand surface. For example the geographical distribution of the population served,

`demandIdVar` - the column name of the unique identifier of the areas,

`demandVar` - the column name of the demand parameter. This could be the population in each region

`growthConstant` - a growth parameter which defines how quickly each label propagates

`bridges` - a named list containing extra linkages beyond those inferred by the demandShape topology. These are used to add in bridges

`outputMap` - should we export a shapefile or just the mapping file

`...` - cache control parameters

Value

a dataframe containing the grouping columns, the outputIdVar and the interpolated value of interpolateVar

`CCG20` *Map data: CCG20*

Description

Map data: CCG20

Usage

```
CCG20
```

Format

An object of class sf (inherits from tbl_df, tbl, data.frame) with 135 rows and 6 columns.

References

https://geoportal.statistics.gov.uk/datasets/clinical-commissioning-groups-april-2020-generalised-c

---

createCatchment *Create a catchment area map*

Description

This implements the label propagation algorithm described in our upcoming paper.

Usage

```
createCatchment(
  supplyShape,
  supplyIdVar = "code",
  supplyVar,
  supplyOutputVars = supplyShape %>% dplyr::groups(),
  demandShape,
  demandIdVar = "code",
  demandVar,
  growthConstant = 1.2,
  bridges = arear::ukconnections,
  outputMap = TRUE
)
```

Arguments

supplyShape - a sf object containing a list of the locations of supply points, with a column containing supply capacity, for example NHS hospital sites, with a bed

supplyIdVar - the variable name of the identifier of the supplier or group of suppliers. For example this could be an NHS trust (multiple sites)

supplyVar - the column name of the supply parameter. This could be number of beds in a hospital.

supplyOutputVars
- (optional - defaults to grouping) the columns from the input that are to be retained in the output

demandShape - the sf object with the geographical map of the demand surface. For example the geographical distribution of the population served,

demandIdVar - the column name of the unique identifier of the areas,

demandVar - the column name of the demand parameter. This could be the population in each region `growthConstant` - a growth parameter which defines how quickly each label propagates

`bridges` - a named list containing extra linkages beyond those inferred by the demandShape topology. These are used to add in bridges

`outputMap` - should we export a shapefile or just the mapping file

Value

a dataframe containing the grouping columns, the outputIdVar and the interpolated value of interpolateVar

`createNeighbourNetwork` *create a neighbourhood network from touching regions in a shapefile, with additional capability to connect non touching areas where there may be bridges etc.*

Description

create a neighbourhood network from touching regions in a shapefile, with additional capability to connect non touching areas where there may be bridges etc.

Usage

```
createNeighbourNetwork(
  shape,
  idVar = "code",
  bridges = arear::ukconnections,
  queen = FALSE,
  ...
)
```

Arguments

`shape` - a sf object, if not present will be loaded from cache

`idVar` - the column containing the coded identifier of the map

`bridges` - a df with the following columns: name start.lat start.long end.lat end.long defining connections between non touching shapes (e.g. bridges / ferries / etc.)

`queen` - include neighbouring areas that only touch at corners, defaults to false.

`...` - passed on to .cached() (cache control) - relevant is nocache = TRUE which prevents this from being precalculated

Value

an edge list of ids with from and to columns

CTRY19 *Map data: CTRY19*

Description

Map data: CTRY19

Usage

```
CTRY19
```

Format

An object of class `sf` (inherits from `tbl_df`, `tbl`, `data.frame`) with 4 rows and 6 columns.

References

https://geoportal.statistics.gov.uk/datasets/countries-december-2019-boundaries-uk-bgc/data

CTYUA19 *Map data: CTYUA19*

Description

Map data: CTYUA19

Usage

```
CTYUA19
```

Format

An object of class `sf` (inherits from `tbl_df`, `tbl`, `data.frame`) with 216 rows and 6 columns.

References

https://geoportal.statistics.gov.uk/datasets/counties-and-unitary-authorities-april-2019-boundaries

`downloadGeojson` *Download a geojson url, standardise it and cache the result*

Description

Download a geojson url, standardise it and cache the result

Usage

```
downloadGeojson(
  geojsonUrl,
  codeCol = "code",
  nameCol = "name",
  altCodeCol = NULL,
  codeType = NA_character_,
  simplify = FALSE,
  id,
  license = "unknown",
  ...
)
```

Arguments

`geojsonUrl` the URL of the geojson resource

`codeCol` - the name of the column containing the id or code

`nameCol` - the name of the column containing the label (optional - defaults to the same as codeCol)

`altCodeCol` - an optional column name containing another code type

`codeType` - the "type" of the code - optional. defaults to NA

`simplify` - do you want to simplify the map

`id` - an id for the map that can be used to retrieve it in the future (through getMap()).

`license` - an optional license string

`...` - passed to .cache, param nocache=TRUE to disable caching

Value

the sf object for this geoJson

downloadMap *Download a map, unpack it, and rename columns.*

Description

to standard code, name, altCode and codeType columns

Usage

```
downloadMap(
  zipUrl,
  mapName = NULL,
  codeCol = "code",
  nameCol = "name",
  altCodeCol = NULL,
  codeType = NA_character_,
  simplify = FALSE,
  wd = getOption("arear.download.dir", tempdir()),
  id = NULL,
  license = "unknown",
  ...
)
```

Arguments

`zipUrl`	- the URL of the zipped shapefile
`mapName`	- the layer name or map name - this is the "xyz" of a zip file containing "xyz.shp". If you are getting multiple layers it is OK to repeatedly call this within the same session as the download is stored, see wd option.
`codeCol`	- the name of the column containing the id or code
`nameCol`	- the name of the column containing the label (optional - defaults to the same as codeCol)
`altCodeCol`	- an optional column name containing another code type
`codeType`	- the "type" of the code - optional. defaults to NA
`simplify`	- do you want to simplify the map
`wd`	- an optional working directory (defaults to ‘getOption("arear.download.dir", tempdir())‘)
`id`	- an optional id for the map that can be used to retrieve it later (through getMap()) - defaults to either the mapName or if not present the name of the zip file.
`license`	- an optional license string
`...`	- passed to .cache, param nocache=TRUE to disable caching

Value

a sf object containing the map

Examples

```
## Not run:
downloadMap(
  zipUrl="https://bit.ly/3A9TnR1",
  mapName="NHS_England_Regions__April_2020__Boundaries_EN_BGC",
  codeCol="nhser20cd",
  nameCol="nhser20nm"
)

## End(Not run)
```

DZ11 *Map data: DZ11*

Description

Map data: DZ11

Usage

```
DZ11
```

Format

An object of class `sf` (inherits from `data.frame`) with 6976 rows and 6 columns.

References

https://data.gov.uk/dataset/ab9f1f20-3b7f-4efa-9bd2-239acf63b540/data-zone-boundaries-2011

GBR_ISO3166_2 *Map data: GBR_ISO3166_2*

Description

Map data: GBR_ISO3166_2

Usage

```
GBR_ISO3166_2
```

Format

An object of class `sf` (inherits from `data.frame`) with 183 rows and 6 columns.

References

https://gadm.org/download_country_v3.html

GBR_ISO3166_3 *Map data: GBR_ISO3166_3*

Description

Map data: GBR_ISO3166_3

Usage

```
GBR_ISO3166_3
```

Format

An object of class `sf` (inherits from `data.frame`) with 406 rows and 6 columns.

References

https://gadm.org/download_country_v3.html

getContainedIn *Generate a mapping representing how the input points fit into the output shape*

Description

This assumes an id column in input and output shapes and

Usage

```
getContainedIn(
  inputShape,
  outputShape,
  inputVars = inputShape %>% dplyr::groups(),
  outputVars = outputShape %>% dplyr::groups(),
  suffix = c(".x", ".y")
)
```

Arguments

`inputShape` - a sf containing points of interest (or shapes)

`outputShape` - a sf containing polygons to locate the input in

`inputVars` - defines the columns of the input that you want to retain (as a dplyr::vars(...) list). This grouping should uniquely identify the row. If not present will use the current grouping of inputShape.

outputVars	- defines the columns of the output that you want to retain (as a dplyr::vars(...) list). This grouping should uniquely identify the row. If not present will use the current grouping of outputShape.
suffix	- the suffix of any duplicated columns as per dplyr::inner_join()

Value

- a mapping as a dataframe relating the input id column and output id columns

Examples

```
# find the hospitals in a given area.
mapping = getContainedIn(
inputShape = arear::surgecapacity,
outputShape = arear::ukcovidmap(),
inputVars = dplyr::vars(hospitalId),
outputVars = dplyr::vars(code)
)
```

getIntersection *Intersection of 2 shapes*

Description

get the intersection between to maps with ids. Caches the result in the working directory.

Usage

```
getIntersection(
  inputShape,
  outputShape,
  suffix = c(".x", ".y"),
  recalcArea = TRUE,
  ...
)
```

Arguments

inputShape	- the input sf
outputShape	- the output sf
suffix	- the suffix of any duplicated columns as per dplyr::inner_join()
recalcArea	- do you need the area of the intersected shape (e.g. for areal interpolation)
...	passed on to .cached() (cache control) - relevant is nocache = TRUE which prevents this from being precalculated

Value

a sf object representing the intersection of the input and output shapes.

getMap *Gets maps for which the metadata is known.*

Description

If a map needs to be downloaded as a shapefile then it is stored temporarily. The location of this download directory can be set as option("arear.download.dir" = "~/.)

Usage

```
getMap(mapId, sources = .loadSources(...), codeType = mapId, ...)
```

Arguments

mapId	- a name of a map
sources	- a list of map sources - see ‘getOption("arear.mapsources",arear::mapsources)“
codeType	- defaults to mapId, the codeType of the map
...	- passed to .cache, param ‘nocache=TRUE‘ to disable caching

Value

a standard sf map

Examples

```
## Not run:
map = getMap("NHSER20")

## End(Not run)
```

GOOGLE_MOBILITY *Map data: GOOGLE_MOBILITY*

Description

Map data: GOOGLE_MOBILITY

Usage

```
GOOGLE_MOBILITY
```

Format

An object of class `sf` (inherits from `tbl_df`, `tbl`, `data.frame`) with 151 rows and 6 columns.

References

https://github.com/datasciencecampus/google-mobility-reports-data/tree/master/geography

HB19 *Map data: HB19*

Description

Map data: HB19

Usage

```
HB19
```

Format

An object of class `sf` (inherits from `data.frame`) with 14 rows and 6 columns.

References

https://www.spatialdata.gov.scot/geonetwork/srv/api/records/f12c3826-4b4b-40e6-bf4f-77b9ed01dc14

interpolateByArea *interpolate a variable from one set of shapes to another*

Description

interpolate a variable from one set of shapes to another

Usage

```
interpolateByArea(
  inputDf,
  inputShape,
  by,
  interpolateVar,
  outputShape,
  inputVars = inputDf %>% dplyr::groups(),
  outputVars = outputShape %>% dplyr::groups(),
  aggregateFn = sum
)
```

Arguments

`inputDf` - in input data frame containing the variable(s) of interest to interpolate. Stratification of the variable can be achieved by grouping

`inputShape` - an input sf map,

`by` - the columns to use to join the inputDf to the map provided in inputShape. This is in the format of a dplyr join specification.

`interpolateVar` - the column that we want to do areal interpolation on,

`outputShape` - an output map which may be grouped by the desired output,

`inputVars` - a list of columns from the inputDf (as a dplyr::vars(...) list) that define the stratification of inputDf and are desired in the output. Defaults to the grouping of inputDf

`outputVars` - a list of columns from the outputShape (as a dplyr::vars(...) list) that we want preserved in output, or defined as a grouping of outputShape

`aggregateFn` - a function that will be applied to area weighted components of interpolateVar - defaults to sum

Value

a dataframe containing the grouping columns, the outputIdVar and the interpolated value of interpolateVar

`LAD19` *Map data: LAD19*

Description

Map data: LAD19

Usage

```
LAD19
```

Format

An object of class `sf` (inherits from `tbl_df`, `tbl`, `data.frame`) with 382 rows and 6 columns.

References

`https://geoportal.statistics.gov.uk/datasets/local-authority-districts-december-2019-boundaries-uk-`

`LAD20` *Map data: LAD20*

Description

Map data: LAD20

Usage

```
LAD20
```

Format

An object of class `sf` (inherits from `tbl_df`, `tbl`, `data.frame`) with 379 rows and 6 columns.

References

https://geoportal.statistics.gov.uk/datasets/local-authority-districts-may-2020-boundaries-uk-buc

`LGD12` *Map data: LGD12*

Description

Map data: LGD12

Usage

```
LGD12
```

Format

An object of class `sf` (inherits from `data.frame`) with 11 rows and 6 columns.

References

https://data.gov.uk/dataset/05f72866-b72b-476a-b6f3-57bd4a768674/osni-open-data-largescale-boundar

LHB19 *Map data: LHB19*

Description

Map data: LHB19

Usage

```
LHB19
```

Format

An object of class `sf` (inherits from `tbl_df`, `tbl`, `data.frame`) with 7 rows and 6 columns.

References

https://geoportal.statistics.gov.uk/datasets/local-health-boards-april-2019-boundaries-wa-buc

listStandardMaps *List the standard maps available to download*

Description

List the standard maps available to download

Usage

```
listStandardMaps()
```

Value

a vector of map names

`londonShape` *An outline of the London NHSER region for use as a mask*

Description

Data available under open government licence: http://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/

Usage

```
londonShape
```

Format

A sf geometry with 1 row:

codeType NHSER

code E4000003 - the NHSER code for London

name London

altCode NA

area the area

geometry the outline

`LSOA11` *Map data: LSOA11*

Description

Map data: LSOA11

Usage

```
LSOA11
```

Format

An object of class `sf` (inherits from `tbl_df`, `tbl`, `data.frame`) with 34753 rows and 6 columns.

References

`https://geoportal.statistics.gov.uk/datasets/lower-layer-super-output-areas-december-2011-boundarie`

`mapsources` *Locations of UK and international shapefiles relevant to COVID*

Description

A list of URLs to get maps, and metadata about the maps in the shapefiles and the column labelling.

Usage

```
mapsources
```

Format

A list with:

source the human readable location of the map

url the web location of the downloadable map shapefile

mapName the name of the map contained in the shapefile (which can contain multiple maps)

codeCol the name of the shapefile column containing the code of the area

nameCol the name of the shapefile column containing the name of the area

altCodeCol the name of the shapefile column containing the an alternative code for the area

simplify should the map be simplified when loaded?

license license terms

`mapTheme` *A map theme to remove extraneous clutter*

Description

A map theme to remove extraneous clutter

Usage

```
mapTheme()
```

Examples

```
ggplot2::ggplot()+mapTheme()
```

MSOA11 *Map data: MSOA11*

Description

Map data: MSOA11

Usage

```
MSOA11
```

Format

An object of class `sf` (inherits from `tbl_df`, `tbl`, `data.frame`) with 7201 rows and 6 columns.

References

https://geoportal.statistics.gov.uk/datasets/middle-layer-super-output-areas-december-2011-boundar

NHSER20 *Map data: NHSER20*

Description

Map data: NHSER20

Usage

```
NHSER20
```

Format

An object of class `sf` (inherits from `tbl_df`, `tbl`, `data.frame`) with 7 rows and 6 columns.

References

https://geoportal.statistics.gov.uk/datasets/nhs-england-regions-april-2020-boundaries-en-bgc

OUTCODE *Map data: OUTCODE*

Description

Map data: OUTCODE

Usage

```
OUTCODE
```

Format

An object of class `sf` (inherits from `data.frame`) with 2880 rows and 6 columns.

References

https://www.opendoorlogistics.com/downloads/

PHEC16 *Map data: PHEC16*

Description

Map data: PHEC16

Usage

```
PHEC16
```

Format

An object of class `sf` (inherits from `tbl_df`, `tbl`, `data.frame`) with 9 rows and 6 columns.

References

https://ons-inspire.esriuk.com/arcgis/rest/services/Health_Boundaries/Public_Health_England_Centres_December_2016_Boundaries

`plotLabelledMap` *Create a map, usually as a chloropleth, with selected areas labelled.*

Description

This can be used to pick out specific highlighted regions based on a filter, label it on a map using a short code, and provide a tabular lookup of label to full name.

Usage

```
plotLabelledMap(
  data,
  mapping,
  ...,
  labelMapping,
  labelStyle = list(),
  labelFilter = rank(-!!labelSort) <= labels,
  labelSort = str2lang(rlang::as_label(mapping$fill)),
  labels = 6,
  labelSize = 6,
  tableSize = 6,
  labelInset = c("both", "inset", "main")
)
```

Arguments

`data` A sf object with some data in it. If using facets this should be grouped. (and if it is grouped facetting will be automatically added)

`mapping` the aesthetics as would be passed to geom_sf

`...` additional formatting parameters as would be passed to geom_sf (defaults to a thin grey line for the edge of the maps.)

`labelMapping` the aesthetics of the label layer. This could include any aesthetics that apply to ggrepel::geom_label_repel other than x,y.. It must include a label aesthetic (which will go on the map) and a name aesthetic (which will go in the lookup table)

`labelStyle` any additional formatting parameters that would be passed to ggrepel::geom_label_repel. Defaults to a blue label on a light transparent background which works for dark maps. A ‘list(segment.colour = "cyan", colour="cyan", fill=="#000000A0")‘ should give a cyan label on a dark transparent background which might work for lighter maps.

`labelFilter` (optional) on what criteria should we select labels to display. by default it gives the top N labels as determined by the fill aesthetic. Bottom N can be achieved with ‘rank(!!labelSort)<=labels‘. In general though any expression filter can be used on the data but bear in mind it will be interpreted in the context of the grouped data which has first sorted by the labelSort expression.

`labelSort` (optional) how should we sort the labels before applying the labelFilter. This defaults to the descending order of the same variable that determines the fill of the main map.

`labels` how many labels do you want, per facet. The default 6 is good for a small number of facets. This will be overridden if labelFilter is specified

`labelSize` in points.

`tableSize` the labels and their associated names from all facets will be assembled into a table as a ggplot/patchwork object. This defines the font size (in points) of this table. No other config is allowed.

`labelInset` if a map has an zoomed in inset as produced by ‘popoutArea()‘, for areas which are in both the main map and the inset you may wish to label only the zoomed area in the "inset", only the unzoomed area in the "main" map or "both".

Value

a list containing 4 items. Plot and legend may be added together to form a ggplot patchwork. e.g. ‘p = plotLabelledMap(...)‘ then ‘p$plot+ggplot2::scale_fill_viridis_c()+ggplot2::facet_wrap(dplyr::vars(...))+p$legend+patchw to actually show the map.

- **plot** a ggplot object showing a chloropleth (usually) which is defined by the main mapping aesthetics, with an overlaid labelling layer defined by the labelMapping aesthetics. This does not include fill or colour scales so you will probably want ‘plot+ggplot2::scale_fill_viridis_c()‘ or something similar to define the fill
- **legend** a ggplot patchwork containing the lookup table from labels to names (as determined by the names aesthetic)
- **labelDf** the filtered dataframe of the labels appearing in the labelling layer. The .x and .y columns are added which show where the label is placed on the main map. the .label and .name show the labels and names respectively
- **labeller** A function that returns a layer of the labels, formatted in same way as the main map. the labeller function takes optional xVar and yVar parameter which are columns in the sf object. These define the x and y aesthetics of the labeller and default to the same position as the main map. The labeller function can be used to add a labels layer to a different map, or to a different graph. This might be useful if you want to combine cartograms with points of interest and have them consistently labelled.

`popoutArea` *Create a new map with a popout panel to show areas where there is a high density of people for example.*

Description

Defaults work well for London on an England only map.

Usage

```
popoutArea(
  shape,
  popoutShape = arear::londonShape,
  popoutPosition = c("NE", "NW", "SE", "SW"),
  popoutScale = 3,
  nudgeX = 0.25,
  nudgeY = 0.25
)
```

Arguments

`shape`	The original shape
`popoutShape`	The mask shape. This will be summarised before applied.
`popoutPosition`	Which corner to place the popout NE,NW,SE or SW
`popoutScale`	A factor to grow the popout area by. This is linear scale so the popout will appear the square of this factor bigger.
`nudgeX`	shift the popout panel by a small amount (in coordinate units)
`nudgeY`	shift the popout panel by a small amount (in coordinate units)

Value

A new map with the content intersecting the popout area duplicated, expanded and placed in the specified corner.

`preview` *Preview a map with POI using leaflet*

Description

Preview a map with POI using leaflet

Usage

```
preview(
  shape,
  shapeLabelGlue = "{name}",
  shapePopupGlue = "{code}",
  poi = NULL,
  poiLabelGlue = "{name}",
  poiPopupGlue = "{code}"
)
```

Arguments

`shape` - the map

`shapeLabelGlue` - a glue specification for the label for each shape

`shapePopupGlue` - a glue specification for the popup for each shape

`poi` - a list of points of interest as a sf object

`poiLabelGlue` - a glue specification for the label for each poi

`poiPopupGlue` - a glue specification for the popup for each poi

Value

htmlwidget

`saveShapefile` *save a shapefile to disk in the current working directory*

Description

save a shapefile to disk in the current working directory

Usage

```
saveShapefile(shape, mapId, dir = getwd(), overwrite = FALSE)
```

Arguments

`shape` - the sf shape

`mapId` - a mapId - will become the zip filename, and the filename of the zipped .shp file

`dir` - the directory (defaults to current working directory)

`overwrite` - the save function will not write over existing files unless this is set to true

Value

a the filename of the zipped shapefile

standardiseMap *Standardise maps to a minimal set of attributes with consistent naming with code, name and codeType columns and an optional altCode column*

Description

Standardise maps to a minimal set of attributes with consistent naming with code, name and codeType columns and an optional altCode column

Usage

```
standardiseMap(sf, codeCol, nameCol, altCodeCol, codeType)
```

Arguments

`sf`	- a non standard map
`codeCol`	- a column name containing the unique code or id for the map
`nameCol`	- the name column
`altCodeCol`	- an alternative code column
`codeType`	- a fixed value for the codeType

Value

a standardised map

surgecapacity *Locations of UK general medical hospitals in mid march 2020 with estimates of beds available and maximal surge capacity HDU beds*

Description

This was manually assembled and curated from various sources in mid march 2020 as the NHS geared up to provide additional capacity to cope with the surge in COVID cases. It is not an up to date picture of NHS capacity. It does not include mental health or community hospitals. The surge capacity seems to have been calculated quite differently in Scotland.

Usage

```
surgecapacity
```

Format

A sf geometry with:

nation England, Wales, etc...

hospitalId An id for the hospital

sector NHS or independent

hospitalName the hospital name

pcds the UK postcode of the hospital

trustId the NHS trust or local health board of the hospital

trustName the NHS trust or local health board name

tier1 indicator of the role of the hospital as an acure provider

hduBeds the number of hdu beds the hospital could have provided at maximum surge in March 2020

acuteBeds the number of acute beds the hospital could have provided at maximum surge in March 2020

`testdata` *A list of regular lattice SF polygons used for testing purposes*

Description

A list of regular lattice SF polygons used for testing purposes

Usage

```
testdata
```

Format

A list with:

grid5x5 5x5 grid centred on 0,0

grid11x11 11x11 grid centred on 0,0

diamond5x5 5x5 diagonal grid centred on 0,0

diamond11x11 11x11 diagonal grid centred on 0,0

gridDemand an 11x11 grid with a demand parameter

gridSupply a 3 point supply

gridSupply2 another 3 point supply with only 2 ids

gridSupplyViolateConstraint a 4 point supply with 2 points in the same grid square

gridSupplyDegenerate a 5 point supply with 2 points in the same grid square amd a different 2 points with the same id

`uk2019adultpopulation` *Mid year 2019 small area estimates for adult population of England, Wales, Scotland, and Northern Ireland*

Description

Small area single digit estimates are aggregated to include only over 18s, and combining gender Data available under open government licence: http://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/

Usage

```
uk2019adultpopulation
```

Format

A 41740 line data frame:

code the ONS code for the ares

name The name for the area

codeType the type of area (LSOA, DZ or LGD)

population the size of the population

Source

England and Wales https://www.ons.gov.uk/peoplepopulationandcommunity/populationandmigration/populationestimates

Scotland https://www.nrscotland.gov.uk/statistics-and-data/statistics/statistics-by-theme/population/population-estimates/2011-based-special-area-population-estimates/small-area-population-estimates/time-series#2018

Northern Ireland https://www.opendatani.gov.uk/dataset/3333626e-b96e-4b90-82fb-474c6c03b868/resource/64bd8dc4-935f-4bdd-9232-90ff33f24732/

`uk2019demographics` *UK small area single year of age population estimates from 2019*

Description

These estimates are appropriate for the majority of the pandemic, and are the highest geographical resolution estimates by single year of age that I could find.

Usage

```
uk2019demographics()
```

Value

a dataframe with age, gender, codeType, code, name and count

uk2019demographicsmap *Shape file related to the mid year 2019 small area estimates for England, Wales, Scotland, and Northern Ireland*

Description

There are 10 regions (mostly in Scotland) where the demographics estimates don't align with this map. This is a small number of people

Usage

```
uk2019demographicsmap()
```

Details

Data available under open government licence: http://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/

Value

A 41730 line data frame:

code the ONS code for the ares

name The name for the area

codeType the type of area (LSOA, DZ or LGD)

altCode NA

Source

England and Wales https://geoportal.statistics.gov.uk/datasets/lower-layer-super-output-areas-december-2011-boundaries-ew-bgc

Scotland https://data.gov.uk/dataset/ab9f1f20-3b7f-4efa-9bd2-239acf63b540/data-zone-boundaries-2011

Northern Ireland https://data.gov.uk/dataset/05f72866-b72b-476a-b6f3-57bd4a768674/osni-open-data-largescale-boundaries-local-government-districts-2012

`uk2019retiredpopulation` *Mid year 2019 small area estimates for adult population of England, Wales, Scotland, and Northern Ireland*

Description

Small area single digit estimates are aggregated to include only over 65s, and combining gender Data available under open government licence: http://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/

Usage

```
uk2019retiredpopulation
```

Format

A 41730 line data frame:

code the ONS code for the ares

name The name for the area

codeType the type of area (LSOA, DZ or LGD)

population the size of the population

Source

England and Wales https://www.ons.gov.uk/peoplepopulationandcommunity/populationandmigration/populationestimates

Scotland https://www.nrscotland.gov.uk/statistics-and-data/statistics/statistics-by-theme/population/population-estimates/2011-based-special-area-population-estimates/small-area-population-estimates/time-series#2018

Northern Ireland https://www.opendatani.gov.uk/dataset/3333626e-b96e-4b90-82fb-474c6c03b868/resource/64bd8dc4-935f-4bdd-9232-90ff33f24732/

`ukconnections` *Locations of UK bridge or ferry route start and end points*

Description

geographical location of bridges / ferries UK connections. picked from https://developers.google.com/maps/documentation/ja click-latlng#maps_event_click_latlng-html

Usage

```
ukconnections
```

Format

A dataframe with:

name the connection

start.lat the latitude of one end

start.long the longditude of one end

end.lat the latitude of the other end

end.long the longditude of the other end

`ukcovidmap` *Shape file related to the detailed level for PHE coronavirus statistical reporting in England, Wales, Scotland, and Northern Ireland on the coronavirus.gov.uk tracker. The legacy is the level at which detailed data was reported in the first wave, prior to the API and was aggregated by https://github.com/tomwhite/covid-19-uk-data*

Description

This map matches the data reported by the PHE coronavirus api when it is downloading lower tier local authority regions e.g. via https://api.coronavirus.data.gov.uk/v2/data?areaType=ltla&metric=newCasesBySpecimenDat There are 2 regions here for which no data is reported in this API - the city of London & the isles of scilly.

Usage

```
ukcovidmap(legacy = FALSE)
```

Arguments

`legacy` use the legacy version of the map from pre coronavirus api times.

Details

Data available under open government licence: http://www.nationalarchives.gov.uk/doc/open-government-licence/version/3/

Value

A 382 line data frame (or a 335 line data frame for legacy):

code the ONS code for the ares

name The name for the area

codeType the type of area (LAD19)

altCode NA

Source

- **England, Wales, Scotland & NI - LAD19** https://geoportal.statistics.gov.uk/datasets/local-authority-districts-december-2019-boundaries-uk-buc
- **England & Wales - LAD19** https://geoportal.statistics.gov.uk/datasets/local-authority-districts-december-2019-boundaries-uk-buc
- **Scotland - CA19** https://www.spatialdata.gov.scot/geonetwork/srv/eng/catalog.search#/metadata/1cd57ea6-8d6e-412b-a9dd-d1c89a80ad62
- **Northern Ireland -LGD12** https://data.gov.uk/dataset/05f72866-b72b-476a-b6f3-57bd4a768674/osni-open-data-largescale-boundaries-local-government-districts-2012

`WD11` *Map data: WD11*

Description

Map data: WD11

Usage

```
WD11
```

Format

An object of class `sf` (inherits from `tbl_df`, `tbl`, `data.frame`) with 8588 rows and 6 columns.

References

https://geoportal.statistics.gov.uk/datasets/wards-december-2011-boundaries-ew-bgc

---

`WD19` *Map data: WD19*

Description

Map data: WD19

Usage

```
WD19
```

Format

An object of class `sf` (inherits from `tbl_df`, `tbl`, `data.frame`) with 8887 rows and 6 columns.

References

https://geoportal.statistics.gov.uk/datasets/wards-december-2019-boundaries-ew-bgc

Index