

# Package ‘ozmaps’

July 22, 2025

**Version** 0.4.5

**Title** Australia Maps

**Description** Maps of Australian coastline and administrative regions. Data can be drawn or accessed directly as simple features objects. Includes simple functions for country or state maps of Australia and in-built data sets of administrative regions from the Australian Bureau of Statistics <<https://www.abs.gov.au/>>. Layers include electoral divisions and local government areas, simplified from the original sources but with sufficient detail to allow mapping of a local municipality.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**ByteCompile** true

**Depends** R (>= 3.3.0)

**Imports** oz, tibble, sf

**Suggests** paletteer (>= 0.2.1), testthat, covr, knitr, rmarkdown

**RoxygenNote** 7.1.1

**URL** <https://github.com/mdsumner/ozmaps>

**BugReports** <https://github.com/mdsumner/ozmaps/issues>

**VignetteBuilder** knitr

**LazyDataCompression** xz

**NeedsCompilation** no

**Author** Michael Sumner [aut, cre],  
Dianne Cook [ctb],  
Dario Herenu [ctb]

**Maintainer** Michael Sumner <[mdsumner@gmail.com](mailto:mdsumner@gmail.com)>

**Repository** CRAN

**Date/Publication** 2021-08-03 04:10:02 UTC

Contents

abs-data . . . . .	2
oz . . . . .	3
ozmap . . . . .	4
ozmap_data . . . . .	5
ozmap_states . . . . .	6
<b>Index</b>	<b>7</b>

---

abs-data	<i>Australian Bureau of Statistics (ABS) map data</i>
----------	---

---

Description

These data sets are simplifications of the formal statistical areas and regions published by the ABS in 2016.

Format

- Simple features data frame with
  - NAME** Area name
  - geometry** Geometry column in 'sfc' format

Details

Each layer was read from the source file with 'sf' package and attributes were removed, leaving a single 'NAME' column from the year-specific column names. The geometry has been simplified using 'rmapshaper::ms\_simplify' with default arguments (0.05 detail).

Several layers are not included from the total available.

The entire nation layer 'AUST' is not included as it is the union of the State and Territory layer.

Statistical Areas Level 1 is not included as it is very large (56Mb after simplification).

The mesh blocks are not included, nor Greater Capital City Statistical Areas, Indigenous Regions, Remoteness Structure, Statistical Areas (L2, L3, L4), State Electoral Divisions. See 'ozmaps.data' for these.

Data layers

- abs\_ced** Commonwealth Electoral Divisions
- abs\_lga** Local Government Areas
- abs\_ste** State and Territory

See Also

The script to create the data set: data-raw/abs-inbuilt.R

## Examples

```
ozmap("abs_ste")

ozmap("abs_lga", col = sample(rainbow(nrow(abs_lga), alpha = .4)))
pal <- rainbow(12, alpha = 0.6) ## boring! install paletteer for ochRe palettes

if (isTRUE(requireNamespace("paletteer", quietly = TRUE))) {
  if (utils::packageVersion("paletteer") < '1.0.0') {
    pal <- paletteer::paletteer_d(package = "ochRe", palette = "namatjira_qual")
  } else {
    pal <- paletteer::paletteer_d(palette = "ochRe::namatjira_qual")
  }
}
opal <- colorRampPalette(pal)
ozmap("abs_ced", col = opal(30))
```

---

 oz

*The oz function*


---

## Description

The classic oz package `oz::oz()` function.

## Usage

```
oz_data(data = "states", ...)
```

## Arguments

<code>data</code>	character string, "states" provides state level else country level
<code>...</code>	passed to <code>oz::ozRegion()</code>

## Details

This function calls `oz::oz()` to draw a basic outline. Use `oz_data()` to obtain the data in native form.

See `oz::ozRegion()` for more details. Here `data` is treated as an identifier, but only "states" or any other value is accepted. If not "states", then country level is returned. Further arguments to `oz::ozRegion()` can be passed in via dots.

## Value

oz class list of coordinates

---

`ozmap`*Australia map*

---

## Description

Draw a map of Australia, with or without states.

## Usage

```
ozmap(x = "states", ..., add = FALSE)
```

## Arguments

<code>x</code>	name of data set to use, default is <code>ozmap_country</code>
<code>...</code>	arguments passed to ...
<code>add</code>	add to existing plot, <code>FALSE</code> by default

## Details

outline data is purely in longitude-latitude form, see `ozmap_data()` to obtain the data itself.<sup>4</sup>

See `abs_ste` for more detailed versions from the Australian Bureau of Statistics. An example is `'abs_ste'` which means 'State and Territory', and so is a more detailed version of `'states'`.

`ozmap()` uses the `sf` package to plot, but does so by only plotting the geometry rather than every column, and leaves the plot region ready for overplotting with other data.

## Value

the data set used, in `'sf'` format

## See Also

`ozmap_data`

## Examples

```
ozmap()  
ozmap("country", lwd = 6)  
ozmap("abs_ced", add = TRUE, border = "firebrick") ## commonwealth (national) electoral divisions
```

---

`ozmap_data`*Australia map data*

---

## Description

Return simple features data frames of various Australian map layers.

## Usage

```
ozmap_data(data = "states", quiet = FALSE, ...)
```

## Arguments

<code>data</code>	name of layer to return, see details
<code>quiet</code>	set to TRUE to suppress messages
<code>...</code>	unused

## Details

Available layers are

- **states** [ozmap\\_states](#) state and territories (low resolution)
- **country** [ozmap\\_country](#) entire country (low resolution)
- **abs\_ced** [abs\\_ced](#) country level electoral divisions
- **abs\_lga** [abs\\_lga](#) local government areas
- **abs\_ste** [abs\\_ste](#) state and territories

## Value

sf data frame with 'NAME' and 'geometry' columns

## Examples

```
country_sf <- ozmap_data("country")

## can take time to print out
lga_sf <- ozmap_data("abs_lga")
lga_sf[1:6, ]
```

---

ozmap_states	<i>Australian map data</i>
--------------	----------------------------

---

**Description**

Australian coastline and boundaries data, including states and territories

**Details**

In-built data set of Australian coastline and provinces (states and territories) simplified from the Australian Bureau of Statistics layer [abs\\_ste](#).

# Index

abs-data, [2](#)  
abs\_ced, [5](#)  
abs\_ced (abs-data), [2](#)  
abs\_lga, [5](#)  
abs\_lga (abs-data), [2](#)  
abs\_ste, [5](#), [6](#)  
abs\_ste (abs-data), [2](#)  
  
oz, [3](#)  
oz::oz(), [3](#)  
oz::ozRegion(), [3](#)  
oz\_data (oz), [3](#)  
oz\_data(), [3](#)  
ozmap, [4](#)  
ozmap\_country, [5](#)  
ozmap\_country (ozmap\_states), [6](#)  
ozmap\_data, [5](#)  
ozmap\_states, [5](#), [6](#)