

# Package ‘choroplethrAdmin1’

October 12, 2022

**Type** Package

**Title** Contains an Administrative-Level-1 Map of the World

**Version** 1.1.1

**Author** Ari Lamstein <ari@lamsteinconsulting.com>

**Maintainer** Ari Lamstein <ari@lamsteinconsulting.com>

**Description** Contains an administrative-level-1 map of the world.  
Administrative-level-1 is the generic term for the largest sub-national subdivision of a country. This package was created for use with the choroplethr package.

**License** BSD\_3\_clause + file LICENSE

**URL** <http://www.arilamstein.com/open-source>

**Imports** ggplot2

**RoxygenNote** 6.0.0

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2017-02-22 08:42:21

## R topics documented:

admin1.map . . . . .	2
admin1.regions . . . . .	2
admin1_map . . . . .	3
get_admin1_countries . . . . .	3
get_admin1_map . . . . .	4
get_admin1_regions . . . . .	4

<b>Index</b>	<b>6</b>
--------------	----------

---

`admin1.map`*An Administrative Level 1 map of every country in the world*

---

**Description**

"Administration Level 1" is the generic term for the largest subnational administrative unit of a country. This unit has different names depending on the country: for example, "state" in the USA and "prefecture" in Japan. In this data.frame the country name is in the column "admin" and the admin1 region name is in the column "region". Rather than working with this object directly, consider using the helper functions listed below.

**Usage**

```
data(admin1.map)
```

**Note**

This map is too large to efficiently render by itself with ggplot2. You should subset it by country before attempting to render. Please see the helper functions.

**References**

Taken from <http://www.naturalearthdata.com/downloads/10m-cultural-vectors/10m-admin-1-states-provinces/>. This is version 3.0.0 of the map and is considered to be beta. The wikipedia page on "Administrative division": [http://en.wikipedia.org/wiki/Administrative\\_division](http://en.wikipedia.org/wiki/Administrative_division)

**See Also**

[admin1.regions](#), [get\\_admin1\\_regions](#), [admin1\\_map](#) and [get\\_admin1\\_map](#)

---

`admin1.regions`*Names of all (country, region) pairs on the admin1.map data.frame. Here "region" means "Administrative Level 1 Region".*

---

**Description**

Names of all (country, region) pairs on the admin1.map data.frame. Here "region" means "Administrative Level 1 Region".

**Usage**

```
data(admin1.regions)
```

**See Also**

[admin1.map](#), [get\\_admin1\\_regions](#), [admin1\\_map](#) and [get\\_admin1\\_map](#)

**Examples**

```
data(admin1.regions)
head(admin1.regions)
```

---

admin1_map	<i>Render an Administrative Level 1 map for a specified country</i>
------------	---

---

**Description**

Uses the map ?admin1.map.

**Usage**

```
admin1_map(country.name)
```

**Arguments**

country.name    The name of the country you want to render.

**See Also**

[admin1.map](#), [admin1.regions](#), [get\\_admin1\\_regions](#), and [get\\_admin1\\_map](#)

**Examples**

```
## Not run:
admin1_map("japan")

admin1_map("canada")

## End(Not run)
```

---

get_admin1_countries	<i>Get all countries on the admin1 map</i>
----------------------	--

---

**Description**

Uses ?admin1.regions

**Usage**

```
get_admin1_countries()
```

**Examples**

```
get_admin1_countries()
```

get\_admin1\_map      *Get an admin1 map for a country*

---

**Description**

Uses ?admin1.map. See ?admin1.regions for how countries are spelled in this map.

**Usage**

```
get_admin1_map(country.name)
```

**Arguments**

country.name      The name of the country you want the admin1 map for.

**See Also**

[admin1.map](#), [admin1.regions](#), [get\\_admin1\\_regions](#) and [admin1\\_map](#)

**Examples**

```
## Not run:
japan.map = get_admin1_map("japan")

ggplot(japan.map, aes(long, lat, group=group)) +
  geom_polygon() +
  ggtitle("An admin1 map of Japan")

## End(Not run)
```

---

get\_admin1\_regions      *Get all admin1 region names for a given country*

---

**Description**

Get all admin1 region names for a given country

**Usage**

```
get_admin1_regions(country.name)
```

**Arguments**

country.name      The name of the country you want the admin1 region names of.

*get\_admin1\_regions*

5

### **See Also**

[admin1.map](#), [admin1.regions](#), [admin1\\_map](#) and [get\\_admin1\\_map](#)

### **Examples**

```
get_admin1_regions("japan")  
get_admin1_regions("canada")
```

# Index

`admin1.map`, 2, 2, 3–5

`admin1.regions`, 2, 2, 3–5

`admin1_map`, 2, 3, 4, 5

`get_admin1_countries`, 3

`get_admin1_map`, 2, 3, 4, 5

`get_admin1_regions`, 2–4, 4