

# Package: mapindia (via r-universe)

November 15, 2024

**Type** Package

**Title** Plot Map of the Indian Subcontinent

**Version** 1.0.1.9000

**Maintainer** Shubham Dutta <shubhamdutta26@gmail.com>

**Description** Get map data frames for the Indian subcontinent with different region levels (e.g., district, state). The package also offers convenience functions for plotting choropleths, visualizing spatial data, and handling state/district codes.

**License** MIT + file LICENSE

**URL** <https://github.com/shubhamdutta26/mapindia>,  
<https://www.shubhamdutta.com/mapindia/>

**BugReports** <https://github.com/shubhamdutta26/mapindia/issues>

**Depends** R (>= 2.10)

**Imports** ggplot2, mapindiatools, rlang, sf, vdiff

**Suggests** knitr, patchwork, rmarkdown, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.3.2

**VignetteBuilder** knitr

**Config/pak/sysreqs** libgdal-dev gdal-bin libgeos-dev make libicu-dev  
libpng-dev libxml2-dev libssl-dev libproj-dev libsqlite3-dev  
libudunits2-dev libx11-dev

**Repository** <https://shubhamdutta26.r-universe.dev>

**RemoteUrl** <https://github.com/shubhamdutta26/mapindia>

**RemoteRef** HEAD

**RemoteSha** e06f51fc60d7f1d0e2ca84440d39470e8318f37e

## Contents

.central	2
.east	3
.north	3
.northeast	4
.south	4
.southsp	5
.west	5
codes	6
code_info	7
mapindia_crs	8
mapindia_transform	8
map_india	10
map_using_data	11
plot_india	12
statepop	14
wb_2011	14
<b>Index</b>	<b>16</b>

---

.central	<i>Central Zonal Council</i>
----------	------------------------------

---

### Description

Central Zonal Council comprises of Chhattisgarh, Madhya Pradesh, Uttarakhand and Uttar Pradesh.

### Usage

```
.central
```

### Format

An object of class character of length 4.

### Details

See [https://en.wikipedia.org/wiki/Central\\_Zonal\\_Council](https://en.wikipedia.org/wiki/Central_Zonal_Council)

### Examples

```
plot_india(include = .central, labels = TRUE)
```

---

.east	<i>Eastern Zonal Council</i>
-------	------------------------------

---

**Description**

Eastern Zonal Council comprises of Bihar, Jharkhand, Odisha, and West Bengal.

**Usage**

```
.east
```

**Format**

An object of class character of length 4.

**Details**

See [https://en.wikipedia.org/wiki/Eastern\\_Zonal\\_Council](https://en.wikipedia.org/wiki/Eastern_Zonal_Council)

**Examples**

```
plot_india(include = .east, labels = TRUE)
```

---

.north	<i>Northern Zonal Council</i>
--------	-------------------------------

---

**Description**

Northern Zonal Council comprises of Chandigarh, Delhi, Haryana, Himachal Pradesh, Jammu and Kashmir, Ladakh, Punjab, and Rajasthan.

**Usage**

```
.north
```

**Format**

An object of class character of length 8.

**Details**

See [https://en.wikipedia.org/wiki/Northern\\_Zonal\\_Council](https://en.wikipedia.org/wiki/Northern_Zonal_Council)

**Examples**

```
plot_india(include = .north, labels = TRUE)
```

---

.northeast	<i>Northeastern Zonal Council</i>
------------	-----------------------------------

---

**Description**

North Eastern Council comprises of Assam, Arunachal Pradesh, Manipur, Meghalaya, Mizoram, Nagaland and Tripura.

**Usage**

```
.northeast
```

**Format**

An object of class character of length 7.

**Details**

See [https://en.wikipedia.org/wiki/North\\_Eastern\\_Council](https://en.wikipedia.org/wiki/North_Eastern_Council)

**Examples**

```
plot_india(include = .northeast, labels = TRUE)
```

---

.south	<i>Southern Zonal Council (w/o Special invitees)</i>
--------	--

---

**Description**

Southern Zonal Council comprises of Andhra Pradesh, Karnataka, Kerala, Puducherry, Tamil Nadu, and Telangana.

**Usage**

```
.south
```

**Format**

An object of class character of length 6.

**Details**

See [https://en.wikipedia.org/wiki/Southern\\_Zonal\\_Council](https://en.wikipedia.org/wiki/Southern_Zonal_Council)

**Examples**

```
plot_india(include = .south, labels = TRUE)
```

---

.southsp                      *Southern Zonal Council (w/ Special invitees)*

---

**Description**

Southern Zonal Council comprises of Andhra Pradesh, Karnataka, Kerala, Puducherry, Tamil Nadu, and Telangana. Andaman and Nicobar and Lakshadweep are special invitees of the Southern Zonal Council.

**Usage**

.southsp

**Format**

An object of class character of length 8.

**Details**

See [https://en.wikipedia.org/wiki/Southern\\_Zonal\\_Council](https://en.wikipedia.org/wiki/Southern_Zonal_Council)

**Examples**

```
plot_india(include = .southsp, labels = TRUE)
```

---

.west                              *Western Zonal Council*

---

**Description**

Western Zonal Council comprises of Dadra and Nagar Haveli and Daman and Diu, Goa, Gujarat, and Maharashtra.

**Usage**

.west

**Format**

An object of class character of length 5.

**Details**

See [https://en.wikipedia.org/wiki/Western\\_Zonal\\_Council](https://en.wikipedia.org/wiki/Western_Zonal_Council)

**Examples**

```
plot_india(include = .west, labels = TRUE)
```

---

`codes`*Retrieve codes for either a India state or district*

---

**Description**

Each state and district has a unique two and five digit code code respectively. Use this function to obtain the code for a state or district.

**Usage**

```
codes(state, district = c())
```

**Arguments**

<code>state</code>	The state(s) for which to obtain a code(s). Can be entered as either a state abbreviation or full name (case-insensitive). ‘state‘ can be entered as either a single state or a vector of states. If ‘state‘ is a vector, ‘district‘ must be omitted.
<code>district</code>	The county for which to obtain a code. Can be entered with or without "district" (case-insensitive).

**Details**

State and district codes are two and five digit codes, respectively. They uniquely identify all states and districts within India. The state and district codes is merged into one code with 5 digits. The first two digits of the five digit district codes correspond to the state that the district belongs to.

**Value**

The code(s) of given state or district.

If only states are entered, a vector of length equal to the number of states is returned. If any states are not found or are invalid, ‘NA‘ is returned in their place.

If a state and district are entered, a single value with the code for the given district is returned. If the district is invalid for the given state, an error is thrown.

If both ‘state‘ and ‘district‘ are omitted, the entire list of available codes are returned, sorted by the state’s abbreviation.

**Note**

A state must be included when searching for `district`, otherwise multiple results may be returned for duplicate district names.

**See Also**

[code\_info()]

## Examples

```
codes()
codes("AP")
codes("Tamil Nadu")

codes(c("AP", "WB", "TN"))

codes("WB", district = "Kolkata")
codes(state = "Uttarakhand", district = "Nainital")
codes(state = "RJ", district = "Pratapgarh")
```

---

code_info	<i>Retrieve states or districts using codes</i>
-----------	---

---

## Description

Retrieve states or districts using codes

## Usage

```
code_info(code11, sortAndRemoveDuplicates = FALSE)

## S3 method for class 'numeric'
code_info(code11, sortAndRemoveDuplicates = FALSE)

## S3 method for class 'character'
code_info(code11, sortAndRemoveDuplicates = FALSE)
```

## Arguments

**code11** A one to three digit, either numeric or character, vector of codes for which to look up states or districts. States have a two digit code and districts have a five digit code (where the first 2 numbers pertain to the state).

**sortAndRemoveDuplicates** Whether or not to sort the output and remove duplicates. By default, the output will be returned in the order of the values provided to the codes parameter. Set this parameter to TRUE to return the output sorted by codes with a single instance of each code.

## Value

A data frame with the states or counties and the associated codes.

If 'codes' is omitted, the data frame containing all available states is returned.

## See Also

[codes()]

**Examples**

```
code_info(2)
code_info("02")
code_info(c("02", "03", "04"))

code_info(19335)
code_info(c("19335", "19337"), sortAndRemoveDuplicates = TRUE)
```

---

mapindia_crs	<i>mapindia coordinate reference system</i>
--------------	---

---

**Description**

This coordinate reference system (CRS) represents the canonical projection used by the mapindia package. It can be used to transform shape files, spatial points, spatial data frames, etc. to the same coordinate representation that is used by the plot\_india function.

**Usage**

```
mapindia_crs()
```

**Value**

An 'sf::st\_crs' object representing the Coordinate Reference System (CRS) for India, specifically EPSG:4326 (WGS 84). This CRS uses latitude and longitude coordinates and is commonly used for geographic data.

**Examples**

```
mapindia_crs()
```

---

mapindia_transform	<i>Convert spatial data to mapindia projection</i>
--------------------	--

---

**Description**

Converting a spatial object of map coordinates will allow those points to line up with the regular mapindia plot by applying the same World Geodetic System 1984 projection to those points as well.

The input 'data' is assumed to contain longitude and latitude coordinates by default. If this is not the case, provide an [sf::st\_crs] object to the 'crs' parameter with the appropriate coordinate reference system.



**Usage**

```
mapindia_transform(data, ...)  
  
## S3 method for class 'sf'  
mapindia_transform(data, ...)  
  
## S3 method for class 'data.frame'  
mapindia_transform(data, ..., input_names = c("lon", "lat"))
```

**Arguments**

data	A data frame containing coordinates in a two column format where the first column represents longitude and the second data frame represents latitude. The names of the data frame column do not matter, just that the order of the columns is kept intact.
...	Additional parameters passed onto [sf::st_as_sf]. By default, 'crs = sf::st_crs(4326)' is used, implying longitude and latitude coordinates.
input_names	A character vector of length two which specifies the longitude and latitude columns of the input data (the ones that should be transformed), respectively. Only required if the input data is a 'data.frame' object. Defaults to 'c("lon", "lat")'.

**Value**

An 'sf' object containing the transformed coordinates from the input data frame with the World Geodetic System 1984 projection applied. The transformed columns will be appended to the data frame so that all original columns should remain intact.

**Examples**

```
data <- data.frame(  
  lon = c(77.10, 88.36, 80.27, 72.87, 77.59, 78.03),  
  lat = c(28.71, 22.57, 13.08, 19.07, 12.97, 30.31),  
  pop = c(8398748, 2325502, 3990456, 2705994, 32113, 347397)  
)  
  
# Transform data  
transformed_data <- mapindia_transform(data)  
  
# Plot transformed data on map  
library(ggplot2)  
  
plot_india() + geom_sf(  
  data = transformed_data,  
  aes(size = pop),  
  color = "red", alpha = 0.5  
)
```

---

map_india	<i>Retrieve India map data</i>
-----------	--------------------------------

---

### Description

Retrieve India map data

### Usage

```
map_india(  
  regions = c("states", "state", "districts", "district"),  
  include = c(),  
  exclude = c()  
)
```

### Arguments

regions	The region breakdown for the map, can be one of ("states", "state", "districts", "district"). The default is "states".
include	The regions to include in the resulting map. If regions is "states"/"state", the value can be either a state name, abbreviation or code. For districts, the district codes must be provided as there can be multiple districts with the same name. If states are provided in the districts map, only districts in the included states will be returned.
exclude	The regions to exclude in the resulting map. If regions is "states"/"state", the value can be either a state name, abbreviation or code. For districts, the district codes must be provided as there can be multiple districts with the same name. The regions listed in the include parameter are applied first and the exclude regions are then removed from the resulting map. Any excluded regions not present in the included regions will be ignored.

### Value

A data frame of India map coordinates divided by the desired regions.

### See Also

[mapindiatools::map\_india()] of which this function is a wrapper for.

### Examples

```
str(map_india())  
  
df <- map_india(regions = "districts")  
states <- map_india(include = c("WB", "NCT", "AP"))
```

---

map_using_data	<i>Join district or state level data to India mapping data</i>
----------------	--

---

**Description**

Join district or state level data to India mapping data

**Usage**

```
map_using_data(data, values = "values", include = c(), exclude = c(), na = NA)
```

**Arguments**

data	The data that should be joined to a India map. This parameter should be a data frame consisting of two columns, a code (2 characters for state, 5 characters for district where first 2 characters correspond to the respective state) and the value that should be associated with that region. The columns of data <i>must</i> be code or state and the value of the 'values' parameter. If both code and state are provided, this function uses the code.
values	The name of the column that contains the values to be associated with a given region. The default is "values".
include	The regions to include in the resulting map. If regions is "states"/"state", the value can be either a state name, abbreviation or code. For districts, the district codes must be provided as there can be multiple districts with the same name. If states are provided in the districts map, only districts in the included states will be returned.
exclude	The regions to exclude in the resulting map. If regions is "states"/"state", the value can be either a state name, abbreviation or code. For districts, the district codes must be provided as there can be multiple districts with the same name. The regions listed in the include parameter are applied first and the exclude regions are then removed from the resulting map. Any excluded regions not present in the included regions will be ignored.
na	The value to be inserted for states or districts that don't have a value in data. This value must be of the same type as the value column of data.

**Value**

A data frame composed of the map data frame (from [map\_india()]) except an extra column containing the values in data is included.

The result can be plotted using [ggplot2::ggplot()] or [plot\_india()].

**See Also**

[plot\_india()]

**Examples**

```
data_01 <- data.frame(code = c("01", "02", "04"), values = c(1, 5, 8))
df <- map_using_data(data_01, na = 0)
```

```
data_02 <- data.frame(state = c("AP", "WB", "Tamil Nadu"), values = c(6, 9, 3))
df <- map_using_data(data_02, na = 0)
```

---

plot\_india

*Conveniently plot basic India map*


---

**Description**

Conveniently plot basic India map

**Usage**

```
plot_india(
  regions = c("states", "state", "districts", "district"),
  include = c(),
  exclude = c(),
  data = data.frame(),
  values = "values",
  theme = theme_map(),
  labels = FALSE,
  label_color = "black",
  label_size = 4,
  ...
)
```

**Arguments**

- |         |   |
|---------|---|
| regions | The region breakdown for the map, can be one of ("states", "state", "districts", "district"). The default is "states".  |
| include | The regions to include in the resulting map. If regions is "states"/"state", the value can be either a state name, abbreviation or code. For districts, the district codes must be provided as there can be multiple districts with the same name. If states are provided in the districts map, only districts in the included states will be returned.   |
| exclude | The regions to exclude in the resulting map. If regions is "states"/"state", the value can be either a state name, abbreviation or code. For districts, the district codes must be provided as there can be multiple districts with the same name. The regions listed in the include parameter are applied first and the exclude regions are then removed from the resulting map. Any excluded regions not present in the included regions will be ignored. |

data	A data frame containing values to plot on the map. This parameter should be a data frame consisting of two columns, a code (2 characters for state, 3 characters for district) and the value that should be associated with that region. The columns of data <i>must</i> be code or state and the value of the 'values' parameter.
values	The name of the column that contains the values to be associated with a given region. The default is "value".
theme	The theme that should be used for plotting the map. The default is theme_map from <a href="#">ggthemes</a> .
labels	Whether or not to display labels on the map. Labels are not displayed by default.
label_color	The color of the labels to display. Corresponds to the color option in the [ggplot2::aes()] mapping. The default is "black".
label_size	The size of the labels to display. Corresponds to the size option in the [ggplot2::aes()] mapping. The default is 4. for more color options.
...	Other arguments to pass to [ggplot2::aes()]. These are often aesthetics, used to set an aesthetic to a fixed value, like color = "red" or linewidth = 3. They affect the appearance of the polygons used to render the map (for example fill color, line color, line thickness, etc.). If any of color/colour, fill, or linewidth are not specified they are set to their default values of color="black", fill="white", and linewidth=0.4.

### Value

A [ggplot2::ggplot] object that contains a basic US map with the described parameters. Since the result is a ggplot object, it can be extended with more [ggplot2::Geom] layers, scales, labels, themes, etc.

### See Also

[indiamap], [ggplot2::theme()]

### Examples

```
plot_india()
plot_india(regions = "states")
plot_india(regions = "districts")

# Output is ggplot object so it can be extended
# with any number of ggplot layers
library(ggplot2)
plot_india(include = c("GJ", "MH", "MP", "UP")) +
  labs(title = "Indian States")
```

---

`statepop`*Indian Population (census and projections) by states*

---

**Description**

Indian Population (census and projections) by states.

The data is formatted for easy merging with output from `[mapindia::map_india()]`.

**Usage**

```
data(statepop)
```

**Format**

A data frame with 36 rows and 8 variables.

**Details**

- `code11` The 2-digit state code.
- `abbr` The 2-letter state abbreviation.
- `state` The full state name.
- `pop_1901` The 1901 population estimate (in number of people).
- `pop_1951` The 1951 population estimate (in number of people).
- `pop_2011` The 2011 population estimate (in number of people).
- `pop_2023` The 2023 population estimate (in number of people).
- `pop_2024` The 2024 population estimate (in number of people).

---

`wb_2011`*West Bengal population, sex-ratio, and literacy data (2011)*

---

**Description**

West Bengal population, sex-ratio, and literacy data for 2011.

The data is formatted for easy merging with output from `[mapindia::map_india()]`.

**Usage**

```
data(wb_2011)
```

**Format**

A data frame with 23 rows and 8 variables.

**Details**

- `code11` The 5-digit code corresponding to the district.
- `abbr` The 2-letter state abbreviation.
- `district` The full district name.
- `pop_2011` The 2011 population estimate (in number of people).
- `pop_increase_2011` The 2011 population increase (percent of people).
- `sex_ratio_2011` The 2011 sex ratio (females per 100 males).
- `literacy_per_2011` The 2011 literacy percentage.
- `pop_2011` The 2011 population density (unknown units).

# Index

## \* datasets

- .central, 2
- .east, 3
- .north, 3
- .northeast, 4
- .south, 4
- .southsp, 5
- .west, 5

## \* data

- statepop, 14
- wb\_2011, 14
- .central, 2
- .east, 3
- .north, 3
- .northeast, 4
- .south, 4
- .southsp, 5
- .west, 5

code\_info, 7

codes, 6

map\_india, 10

map\_using\_data, 11

mapindia\_crs, 8

mapindia\_transform, 8

plot\_india, 12

statepop, 14

wb\_2011, 14