Package 'IPtoCountry'

August 29, 2016

Title Convert IP Addresses to Country Names or Full Location with Geoplotting

Version 0.0.1

Date 2016-07-21

Description
Tools for identifying the origins of IP addresses. Includes functions for converting IP addresses to country names, location details (region, city, zip, latitude, longitude), IP codes, binary values, as well
as a function for plotting IP locations on a world map. This product includes IP2Location LITE data available
from <http://www.ip2location.com> and is is available by Creative Commons Attribution-ShareAlike 4.0 Interational
license (CC-BY-SA 4.0).

Depends R (>= 3.2.3)

Imports data.table, ggplot2, dtables, maps, scales, devtools, install.load

License GPL-3

LazyData true

RoxygenNote 5.0.1

NeedsCompilation no

Author Ronald E. Robertson [cre, aut],
Felix W. Tran [aut],
Jonathan Mejia [aut],
Christine Mourani [aut]

Maintainer Ronald E. Robertson <robertson.ron@husky.neu.edu>

Repository CRAN

Date/Publication 2016-07-22 01:31:05

R topics documented:

`ip2location.lite.db1` *IP database - small*

Description

A dataset containing 149,449 IP integer ranges and their corresponding country names and abbreviations

Usage

```
ip2location.lite.db1
```

Format

A data.frame with four variables

IPfrom Start of IP integer range

IPto End of IP integer range

Abrv Two-character country code based on ISO 3166.

Country Country name based on ISO 3166.

Details

Last updated: 06/06/2016

Source

http://lite.ip2location.com

IPs *IP addresses*

Description

A dataset containing 1000 randomly generated IP addresses

Usage

```
IPs
```

Format

A character vector with length 1000

Source

Generated using IP_generator

IP_binary *Convert IP address to binary*

Description

Create method for splitting an IP address values into lists of binary values

Usage

```
IP_binary(IP.address, as.list = FALSE)
```

Arguments

IP.address	A character vector of IP addresses
as.list	logical, FALSE returns a named character vector instead of a list

Value

A list of IP addresses converted to character strings of binary values

IP_country *Convert IP address to country name*

Description

Convert IP address to country name

Usage

```
IP_country(IP.address, IP.database = NULL)
```

Arguments

IP.address	a character or factor vector of one or more IP addresses
IP.database	an IP database, see ?ip2location.lite.db1

Value

Returns a factor vector of country names corresponding to IP.address

Examples

```
IP_country(IPs)
```

IP_generator *Random IP address generator*

Description

Random IP address generator

Usage

```
IP_generator(n)
```

Arguments

n	number of IP addresses to return

Value

Returns a random IP address vector of length n

Examples

```
# Generate 100 random IP addresses
IP_generator(100)
```

IP_integer — *Convert IP address to IP integer*

Description

Convert IP address to IP integer

Usage

```
IP_integer(IP.address)
```

Arguments

IP.address	a character or factor vector of one or more IP addresses

Value

Returns a numeric vector of IP integers from IP.address. The coversion formula used here splits the string into four octets, multiplies the first three by 256^(4-n), and takes the sum of the four modified octets.

Examples

```
head(IP_integer(IPs))
```

IP_location — *Convert IP address to location (country name, region, city, zip code, latitude, longitude, and GMT)*

Description

Convert IP address to location (country name, region, city, zip code, latitude, longitude, and GMT)

Usage

```
IP_location(IP.address, IP.database = NULL)
```

Arguments

IP.address	a character or factor vector of one or more IP addresses
IP.database	an IP database, see ?ip2location.lite.db11

Value

Returns a data.frame containin the country name, region, city, and zip code corresponding to IP.address

Examples

```
# Only run this example in interactive R sessions
if(interactive()) {
  IP.location = IP_location(IPs)
  head(IP.location)
}
```

`IP_lookup` *Match IP integer to country name*

Description

Match IP integer to country name

Usage

```
IP_lookup(IP.integer, IP.database = NULL)
```

Arguments

`IP.integer`	a character or factor vector of one or more IP addresses
`IP.database`	an IP database, see `?ip2location.lite.db1` for details on default database from http://lite.ip2location.com

Value

Returns country from IP integers

Examples

```
IP.integer <- IP_integer(IPs)
head(IP_lookup(IP.integer))
```

`IP_lookup_full` *Match IP integer to all location data*

Description

Match IP integer to all location data

Usage

```
IP_lookup_full(IP.integer, IP.database = NULL)
```

Arguments

IP.integer	a character or factor vector of one or more IP addresses
IP.database	an IP database, see ?ip2location for details on default database from http://lite.ip2location.com

Value

Returns country from IP integers

Examples

```
if(interactive()) {
  IP.integer <- IP_integer(IPs)
  head(IP_lookup_full(IP.integer))
}
```

IP_plot *Convert IPs to countries and plot on world map*

Description

Convert IPs to countries and plot on world map

Usage

```
IP_plot(IP.address)
```

Arguments

IP.address	a vector or column of IP addresses

Value

Returns a world map plot with gradient coloring reflecting the percentage of IP addresses originating in each country

Examples

```
IP_plot(IPs)
```

IP_split *Split IP addresses*

Description

Split IP addresses

Usage

```
IP_split(IP.address, integer = TRUE, data.frame = TRUE)
```

Arguments

IP.address	vector of IPv4 addresses
integer	logical, convert output to class integer
data.frame	logical, convert output to data.frame

Value

Returns a matrix or data.frame with four columns and as many rows as IP.addresses were inputted

Examples

```
head(IP_split(IPs))
```

Index